Мурат Маляров
Валерій Христич
Максим Журавський

КУРС
ЛЕКЦІЙ

Основи інформаційних технологій

Харків 2019

Загальний перелік лекцій

Лекція 1: Microsoft Excel 2016.
Загальні відомості, введення, редагування та форматування даних.

Лекція 2: Microsoft Excel 2016.
Використання MS Excel для графічного представлення табличних даних.

Лекція 3: Microsoft Excel 2016.
Використання електронних таблиць для створення та обробки баз даних

Лекція 4: Microsoft Access 2016.
Реляційні бази даних – створення та їх обробка за допомогою СУБД

Лекція 5: WAN Internet.
Локальні та глобальні комп'ютерні мережі. Мережа Internet

Лекція 6: Мова HTML.
Основні принципи створення WEB-сторінок.

Лекція 7: Мова HTML.
Створення таблиць та форм на WEB-сторінках

Лекція 8: Основи CSS.
Загальні принципи верстки та оформлення Web-сторінки

Лекція 9: VBA
Ведення в об'єктно-орієнтовне програмування. Мова програмування VBA

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Microsoft Excel 2016.
Загальні відомості, введення, редагування та форматування даних.

Зміст

Microsoft Excel 2016. Загальні відомості, введення, редагування та форматування даних. 1
1. *Загальна характеристика Microsoft Excel 2016.* 3
Завантаження Excel 2016. Інтерфейс програми. 4
Робоча книга 7
Первинне налаштування Excel 8
2. *Початок роботи з Microsoft Excel 2016* 11
Створення нової робочої книги 12
Збереження файлу нового документа 13
3. *Робота з робочою книгою Excel* 14
Введення даних 14
Введення та редагування числових та текстових даних у Excel 15
Операції форматування у Excel 17
Копіювання і переміщення вмісту клітинок 20
Автоматизація введення даних 21
Вставлення і вилучення стовпців і рядків робочого листа 25
Питання для самоконтролю 26

1. *Загальна характеристика Microsoft Excel 2016.*

Для того, щоб за допомогою персонального комп'ютера розв'язувати різноманітні задачі, існує **прикладне програмне забезпечення**, яке використовується широким колом користувачів. Прикладні програми поділяються на різні групи, наприклад текстові редактори, графічні редактори, електронні таблиці та системи управління базами даних (СУБД) тощо.

Електронні таблиці – це програми, що забезпечують роботу з великими таблицями чисел, а також автоматизацію математичних обчислень за допомогою формул. Вони забезпечують роботу з символьними даними, здійснюють побудову діаграм, графіків тощо. Найвідомішим представником цієї категорії прикладного програмного забезпечення є **Microsoft Excel**.

Microsoft Excel (повна назва **Microsoft Office Excel**) — табличний процесор, програма для роботи з електронними таблицями, створена корпорацією *Microsoft* для Microsoft Windows, Windows NT і Mac OS. Програма Excel входить до пакету Microsoft Office і призначена для роботи з різноманітними даними, які представляються у вигляді таблиць.

Excel дозволяє:

- завдяки тому, що лист Excel являє собою готову таблицю, Excel часто використовують для створення документів без усіляких розрахунків, що просто мають табличне представлення (наприклад, прайс-листи в магазинах, розклади занять);

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

- виконувати обчислення будь-якої складності та спрямування: наукові та інженерні; фінансово-економічні; статистичні; математичні; логічні тощо. Excel має понад 450 вбудованих функцій, використання яких суттєво полегшує вирішення багатьох завдань;
- представляти дані з таблиць в наглядній формі у вигляді різноманітних діаграм, що суттєво полегшує сприйняття цих даних;
- створювати та працювати з нескладними базами даних;
- численні та різноманітні інструменти (швидкий аналіз, умовне форматування, зведені таблиці іта діаграми) дозволяють аналізувати і прогнозувати данні з метою прийняття рішень;
- обробляти та аналізувати дані експериментів і статистичні дані, отримувати тренди та емпіричні формули;
- створювати макроси, тобто макрокоманди, які використовуються для автоматизації розв'язання процедур і задач, що часто повторюються, виконувати ряд інших функцій.
- використовувати вбудовану мову програмування Visual Basic for Applications (VBA) для створення повнофункціональних додатків з інтерфейсом користувача у вигляді екранних форм з елементами керування.

> У 1982 році Microsoft запустила на ринок свій перший електронний табличний процесор Multiplan, який був дуже популярний, але на MS-DOS системах він поступався Lotus 1-2-3. Перша версія Excel призначалася для Mac і була випущена в 1985 році, а перша версія для Windows була випущена в листопаді 1987 року. Lotus не поспішала випускати 1-2-3 під Windows, і Excel з 1988 року почала обходити з продажу 1-2-3, що в кінцевому підсумку допомогло Microsoft досягти позицій провідного розробника програмного забезпечення. Microsoft зміцнювала свою перевагу з випуском кожної нової версії, що мало місце приблизно кожні два роки. Поточна версія для платформи Windows - Excel 16, також відома як Microsoft Office Excel 2016. Поточна версія для платформи macOS - Microsoft Excel 2016.
>
> На початку свого шляху Excel став причиною позову про товарний знак від іншої компанії, вже продавала пакет програм під назвою «Excel». В результаті суперечки Microsoft була зобов'язана використовувати назву «Microsoft Excel» у всіх своїх офіційних прес-релізах і юридичних документах. Однак з часом ця практика була забута, і Microsoft остаточно усунула проблему, придбавши товарний знак іншої програми. Microsoft також вирішила використовувати літери XL як скорочена назва програми: іконка Windows-програми складається з стилізованого зображення цих двох букв, а розширення файлів за замовчуванням в Excel - .xls.
>
> Excel був першим табличним процесором, що дозволяв користувачеві змінювати зовнішній вигляд таблиці на екрані: шрифти, символи і зовнішній вигляд осередків. Він також першим представив метод розумного перерахунку комірок - оновлення тільки тих комірок, що залежать від змінених: раніше табличні процесори перераховували всі комірки; це робилося або після кожної зміни (що на великих таблицях довго), або по команді користувача (що могло вводити користувача в оману не перерахованими значеннями).

Завантаження Excel 2016. Інтерфейс програми.

Існує кілька способів запуску Excel, але найбільш ефективним є запуск клацанням по значку **Excel** або на Панелі завдань (рис. 1.1), або на Робочому столі.

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

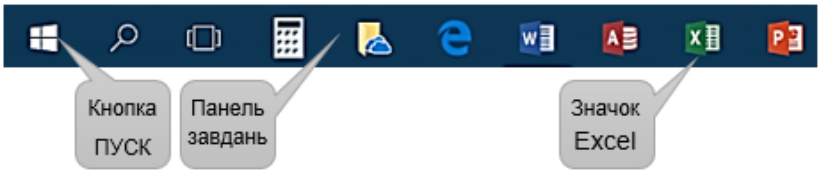


Рис. 1.1 Панель завдань зі значком Excel

Також можна знайти цю програму з усіх встановлених на комп'ютері програм, що відображаються на панелі **Усі програми**, яка відкривається після клацання по кнопці **Пуск**, розташованої в лівому нижньому кутку екрану. Програми на цій панелі розташовуються відповідно з їх назвами за абеткою.

Програма **Excel** запускається після клацання по її значку (рис. 1.1) і відкривається в стандартному для додатків MS Office вікні (рис. 1.2). Як і всі вікна в операційній системі Windows, вікно Microsoft Excel має **заголовок** (1), у правій частині якого є стандартні **кнопки управління вікном** (3) (Згорнути , Розгорнути , Закрити).

У лівій частині заголовка розташований значок програми , що являє собою **кнопку системного меню** (2), праворуч від якої розташована **Панель швидкого доступу** (5), яка може настроюватися під потреби користувача: змінювати розмір і положення, додавати або видаляти кнопки команд. За замовчуванням на ній всього три кнопки (**Зберегти**, **Відмінити дію**, **Повторити дію**), але шляхом налаштування на неї можна вивести будь-яку кількість команд Excel.

Нижче заголовка розташована стрічка (6) з восьма вкладками (4). Вкладка – це смуга з розміщеними на ній керуючими елементами (кнопками, розкривними списками, полями тощо). Кілька елементів можуть бути об'єднані в групу. У цьому випадку вони вкладені в рамку, в нижній частині якої знаходиться назва групи. У деяких групах, праворуч від назви груп, розташовуються маленькі значки - Кнопки виклику діалогових вікон . При натисканні такої кнопки відкривається відповідне діалогове вікно або область завдань, що містить додаткові параметри, пов'язані з даною групою.

Змінити розташування стрічки на екрані неможливо, але для збільшення розмірів робочої області стрічку можна згорнути:

- - командою Згорнути стрічку з контекстного меню. Контекстне меню викликається клацанням правої кнопки миші по стрічці;
- - або сполученням клавіш Ctrl+F1.

При цьому на екрані відображаються тільки назви вкладок. Клацання по назві будь-якої вкладки розгортає відповідну стрічку.

Вкладки - назви стрічок, які об'єднують командні кнопки за функціональним принципом. Одна з вкладок на стрічці завжди активна. Крім основних вкладок, відповідно до поточного стану об'єкта на стрічці автоматично можуть з'являтися Контекстні вкладки (вкладки що відповідають конкретному об’єкту та його поточному стану). Наприклад, якщо в документі виділити рисунок, то автоматично з'являться вкладка Засоби креслення

У Microsoft Excel командними вкладками є:

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Вкладка **Головна** містить команди, пов'язані з буфером обміну, вибором шрифтів, вирівнюванням даних, оформленням числових значень, стилями і правкою.

Вкладка **Вставка** містить інструменти для додавання таблиць, діаграм, ілюстрацій, посилань, колонтитулів, текстових об'єктів і символів.

Вкладка **Розмітка** сторінки містить команди для роботи з темами, фоновими зображеннями, параметрами сторінок тощо.

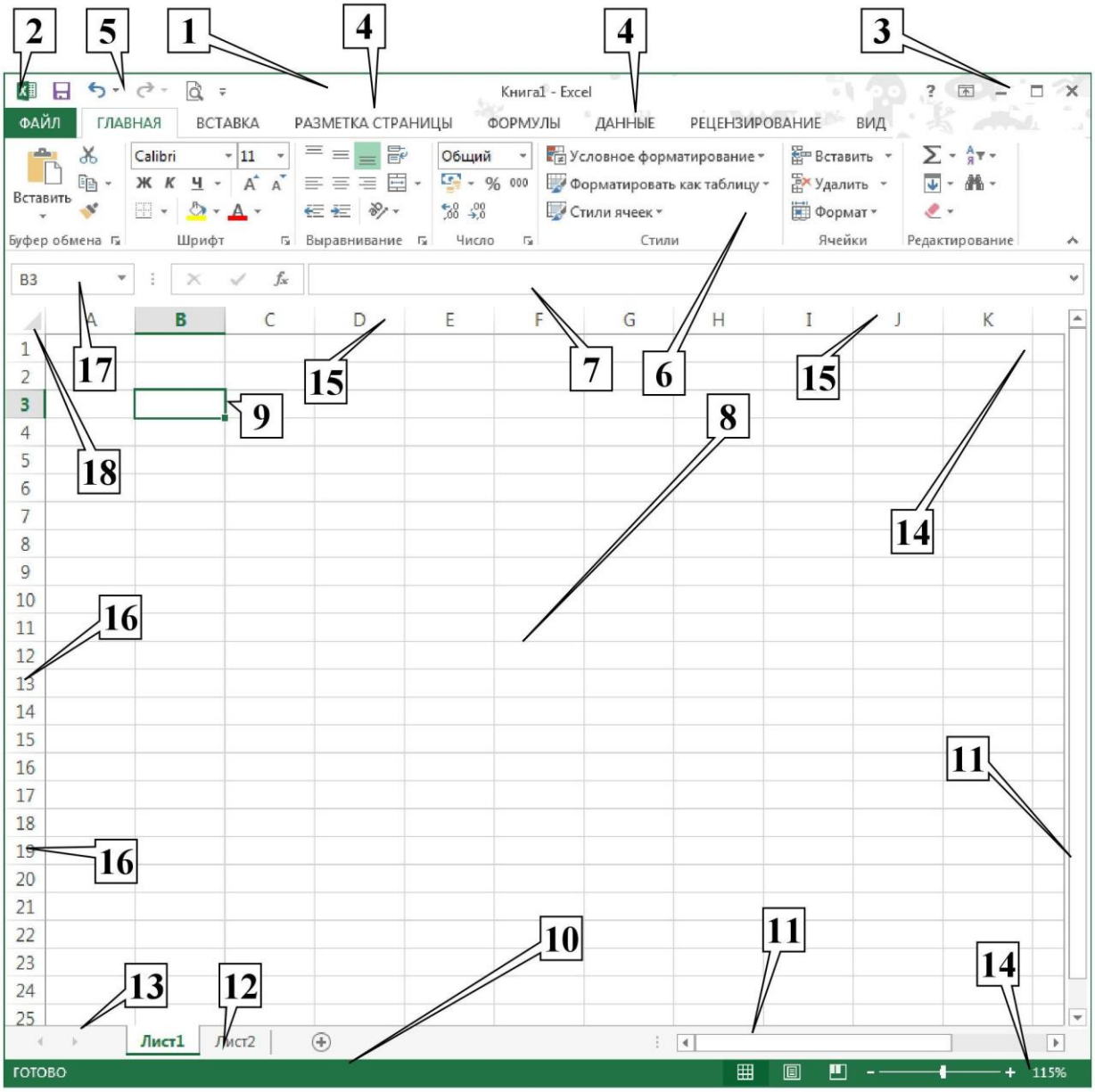


Рис. 1.2 Вікно програми MS Excel 2016

Містить наступні елементи: рядок заголовку (1); кнопка системного меню (2); кнопки керування вікном додатку (3); вкладки (4); панель швидкого доступу (5); стрічка команд (6); рядок формул (7); робоча область (8); рамка виділення (9); рядок стану (10); смуги прокручування (11); ярлички аркушів (12); кнопки переходу по аркушах (13); масштабування аркуша (14); заголовки стовпців (15); заголовки рядків (16); поле імені комірки (17); кнопка виділення всієї таблиці (18).

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Вкладка **Формули** призначена для роботи з формулами, функціями, які зручним чином розміщені по категоріям для обробки іменованих комірок, перевірки формул і параметрів обчислення.

Вкладка **Дані** призначена для роботи з базами даних. Містить команди для отримання зовнішніх даних, управління зовнішніми з'єднаннями, сортування та фільтрації даних, усунення дублікатів, перевірки і консолідації даних, а також групування і розгрупування діапазонів комірок.

Вкладка **Рецензування** містить в собі команди, необхідні для рецензування, коментування, розповсюдження та захисту листа.

Вкладка **Вид** має команди для вибору різних представлень робочої книги, приховування та відображення елементів робочого листа (сітки, лінійки, рядка формул тощо), збільшення або зменшення масштабу, а також роботи з вікном робочого листа.

Вкладка **Розробник** містить засоби створення макросів і форм, а також функції для роботи з .xml. За замовчуванням ця вкладка не відображається на стрічці.

Для того щоб побачити вкладку, треба клацнути по відповідній назві вкладки (ярлика). При завантаженні програми активною є вкладка **Головна**.

Також на стрічці є вкладка **Файл**, що відкриває представлення Microsoft Office Backstage, яке містить команди для роботи з файлами, роботи з поточним документом, а також команди для налаштування Excel.

Під стрічкою команд розташована Робоча область (8), що займає найбільшу центральну частину вікна і служить для відображення робочого документа. У нижній частині під робочою областю розташовано рядок стану (10). Він містить деяку важливу інформацію і команди, що полегшують роботу з документом. У його правій частині розміщено Повзунок масштабування (14), переміщення якого у бік + або - дозволяє збільшувати або зменшувати масштаб відображення документа у вікні.

Робоча книга

Документ Excel називається Робочою книгою та при збереженні отримує розширення **.xlsx** (для попередніх версій **.xls**). Назва Робочої книги відображається в Рядку заголовка (1). Робоча книга являє собою набір Робочих аркушів (8), кожен з яких має табличну структуру. Загальна кількість Робочих аркушів залежить від налаштувань Excel. У робочій області вікна Excel відображається тільки поточний Робочий аркуш, з яким і ведеться робота. Кожен Робочий аркуш має назву, яка відображається на Ярлику аркуша (12). Ярлики відображаються в нижній частині вікна програми. За допомогою клацань по ярликах можна переключатися між Робочими аркушами, які входять в Робочу книгу. Для переміщення можна також використовувати кнопки зліва від ярликів (13). Під час роботи аркуші можна перейменовувати, переміщати (копіювати), додавати і вилучати, використовуючи контекстні меню ярличків або відповідні команди вкладок

Електронна таблиця (робочий аркуш) – це сітка, яка складається з **стовпців** і **рядків**, перетин яких утворює **клітинки**.

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Стовпці позначаються номерами з великих латинських літер, які вказуються у верхній частині аркуша (15). Робочий лист може містити до 2^{14}=16384 стовпців, пронумерованих від A до Z, від AA до AZ, від BA до BZ і так далі до останнього, 16384-го стовпця, який позначається літерами XFD.

Рядки позначаються ліворуч аркуша (16) і нумеруються цифрами від 1 до 2^{20}=1048576. На перетині стовпців і рядків утворюються клітинки таблиці. Вони є мінімальними контейнерами збереження даних, з стандартною шириною яка дорівнює приблизно 9 символам шрифту за замовчанням.

Кожна клітинка таблиці має свою адресу. Ця адреса складається з номера стовпця і рядка, на перетині яких вона розташована, наприклад, **A1** або **D234**. Першим **завжди** вказується номер стовпця. На Робочому аркуші завжди активна одна клітинка, вона виділяється товстою рамкою, а номер стовпця і номер рядка, на перетині яких вона знаходиться, виділяються кольором. Ця рамка в Excel грає роль табличного курсора та має назву рамка виділення (9). Усі операції введення і редагування даних завжди проводяться тільки в активній клітинці. Ім'я активної клітинки відображається в лівій верхній частині аркуша у полі імені комірки (17). У Excel можна адресуватися не тільки на окремі клітинки таблиці, а й на діапазони, що включають або частину стовпця, або частину рядка або блок сусідніх комірок. Наприклад, діапазон **A3:F3** – це частина рядка 3, що включає 6 клітинок; **D8:D11** – частина стовпця D, що включає 4 клітинки; **A1:C11** – блок клітинок, заданий адресами лівої верхньої і правої нижньої клітинок.

> Зверніть увагу, що на екрані відображається тільки частина електронної таблиці. Для перегляду невидимих частин таблиці користуються смугами прокручування (11). Excel при збереженні робочої книги записує в файл тільки прямокутну область робочих листів, яка примикає до лівого верхнього кута і яка містить клітинки з даними. Інформація про порожні клітинки поза цією областю в пам'яті комп'ютера не зберігається.

Первинне налаштування Excel

Якщо почати роботу з новим документом, то Робоча книга буде мати певну кількість аркушів, текст буде друкуватися шрифтом певного накреслення і розміру, буде задіяно певну мову введення тощо. Тобто документ буде створюватися зі заздалегідь встановленими параметрами і властивостями, які називаються **параметрами за замовчанням**. Якщо при роботі з документом потрібні параметри, які відрізняються від параметрів за замовчанням, то це зажадає багаторазових перемикань між цими параметрами, що знижує ефективність роботи з документом. Більш того, деякі параметри за замовчанням можуть призводити до автоматичного виконання деяких дій, що не передбачені користувачем. Тому для ефективної роботи з документами надзвичайно важливо виконати первинне настроювання Excel. Excel допускає встановлення великої кількості параметрів, з яких розглянемо найбільш важливі й тільки ті, які можуть істотно вплинути на роботу з документами. Послідовність зміни параметрів, встановлених за замовчанням, істотного значення не має.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Найбільш важливі параметри встановлюються на вкладці **Файл** (рис. 1.3). Вона містить команди, які дозволяють керувати документами, контролювати метадані і персональні дані, налаштовувати параметри інтерфейсу тощо. Ця вкладка найбільш важлива і відрізняється від інших вкладок стрічки, тому видалити її неможливо, крім того, її назва забарвлена у зелений колір. Вкладка Файл розгортається на весь екран, перекриваючи робочу область. У лівій частині вкладки відображаються назви підвкладок, а в правій частині параметри, що відповідають обраній підвкладці. Параметри вкладки за замовчанням автоматично встановлюються при інсталяції Excel і у більшості випадків задовільнять вимогам користувачів, тому змінювати їх зазвичай потреби не виникає. Однак доцільно уважно ознайомитися зі всіма параметрами, щоб при необхідності знати, де і як їх можна змінити.

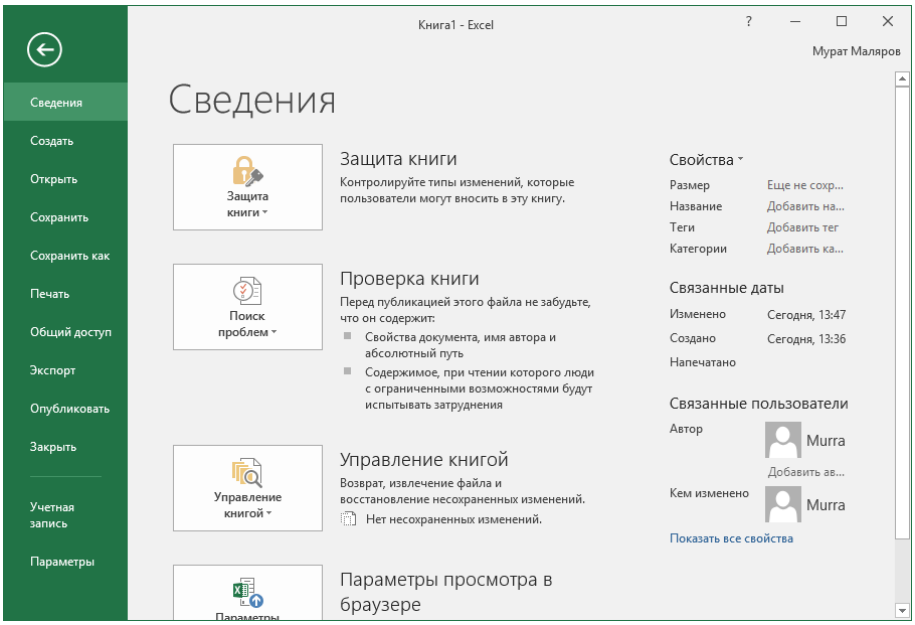


Рис. 1.3 Вкладка Файл

> Не потрібно змінювати параметри документів, встановлені за замовчанням, якщо заздалегідь не відомий результат їх застосування або ви не впевнені у своїх діях.

Вибір підвкладки **Параметри** відкриває діалог Параметри Excel на вкладці **Загальні** (рис. 1.4). В розділі Параметри інтерфейсу користувача доцільно встановити всі прапорці. Це впливає на швидкість роботи з документом. Також за допомогою вкладки можна встановити тип і розмір шрифту, який буде використовуватися при введені даних та початкову кількість аркушів для нової Робочої книги. Також на вкладці з розкривних списків можна вибрати різні варіанти оформлення вікна Excel. Вибір цих параметрів залежить від колірних переваг користувача.

На вкладці **Формули** (рис. 1.5) встановлюються параметри функціонування і обчислення формул і обробки помилок. Зазвичай параметри, встановлені на цій вкладці не потребують коригування. Однак, слід звернути увагу на те, щоб було встановлено прапорець для опції автоматичного перерахунку таблиць при введенні нових даних.

Параметри вкладки Збереження носять переважно інформаційний характер, вони автоматично встановлюються системою, змінювати їх без особливої необхідності не варто. Однак параметру **Зберігати відомості про автозбереження кожні…** в області Збереження книг необхідно приділити особливу увагу. Так як робота з документом відбувається в оперативній пам'яті комп'ютера, яка енергозалежна, то при відключенні струму вся інформація щодо змін в поточному документі втрачається. Щоб уникнути втрати даних через різні непередбачені ситуації (виключення живлення,

Перелік лекцій

перепад напруги в мережі тощо) в процесі роботи над документом бажано зберігати його автоматично через певні проміжки часу. Це дозволить відновити документ з незначними втратами після аварійної ситуації. Для цього необхідно встановити прапорець **автозбереження кожні… хвилин** і виставити тривалість інтервалу. Рекомендується значення в 10-15 хвилин, яке обмежить максимально можливий обсяг втраченої роботи. Також доцільно встановити прапорець Зберігати останню автоматичне збереження версію у разі закритті без збереження, це дозволить зберегти внесені зміни в документ, у разі якщо ви забули його зберегти перед закриттям.

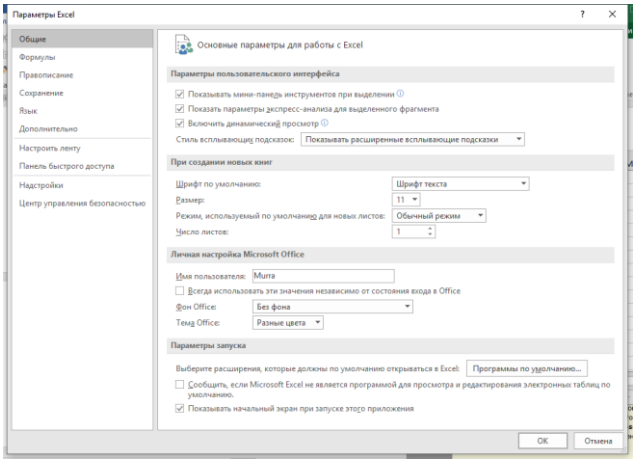


Рис. 1.4 Підвкладка Параметри вкладка Загальні

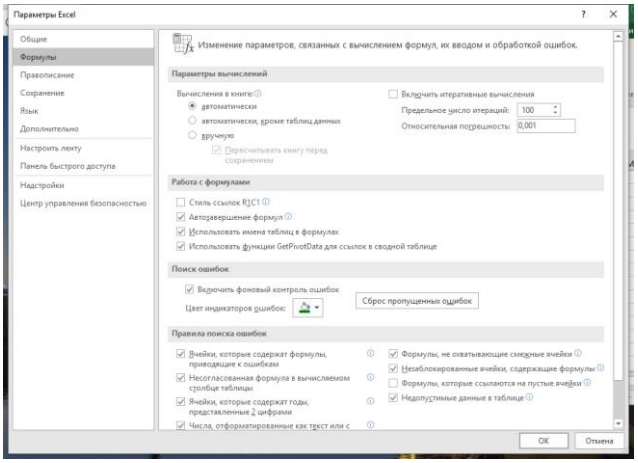


Рис. 1.5 Підвкладка Параметри вкладка Формули

Використання вкладок в більшості випадків досить зручне, однак іноді для роботи з різними об'єктами того самого документа необхідно перемикання між стрічками, що вимагає додаткових витрат часу. Підвищити ефективність роботи в такому випадку можна, створивши для користувача додаткові вкладки і розмістивши на них будь-які групи командних кнопок, що використовуються найбільш часто. Для створення користувальницької вкладки необхідно виконати команду **Файл – Параметри – Настроювання стрічки** або команду **Налаштування стрічки** з контекстного меню.

Зазвичай, щоб отримати доступ до команди, необхідно активізувати певну вкладку, а може ще й контекстну вкладку, знайти і клацнути відповідний значок команди. Це може привести до невиправданих витрат часу, особливо для деяких команд, які використовуються найбільш часто. Для того, щоб отримати швидкий і зручний доступ до команд з різних вкладок, в яких найбільш часто виникає потреба, використовується **Панель швидкого доступу** (рис. 1.2 пункт 5). Ця панель розташовується у верхній лівій частини вікна програми. Панель швидкого доступу можна налаштовувати: змінювати розмір і положення панелі, додавати або видаляти кнопки команд.

Щоб додати кнопки команд на панель, необхідно клацнути кнопку на її правому кінці. Відкриється діалог, який містить список команд - потрібні команди необхідно клацнути. Обрані команди відзначаються прапорцем. Збільшити список доступних команд можна, вибравши опцію **Інші команди**.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Ще одним способом первинних налаштувань є виведення інформації на рядку стану. Рядок стану розташовано в нижній частині вікна програми (рис. 1.2 пункт 10) і він є стандартним елементом інтерфейсу, в якому можна знайти дані про стан документа, підказки щодо його властивостей, індикатори та іншу корисну інформацію. За допомогою елементів управління, розташованих в рядку стану, можна швидко переходити між режимами представлення документа, змінювати масштаб відображення документа за допомогою повзунка в правій частині і виконувати інші корисні дії, не вдаючись до кнопок на вкладках.

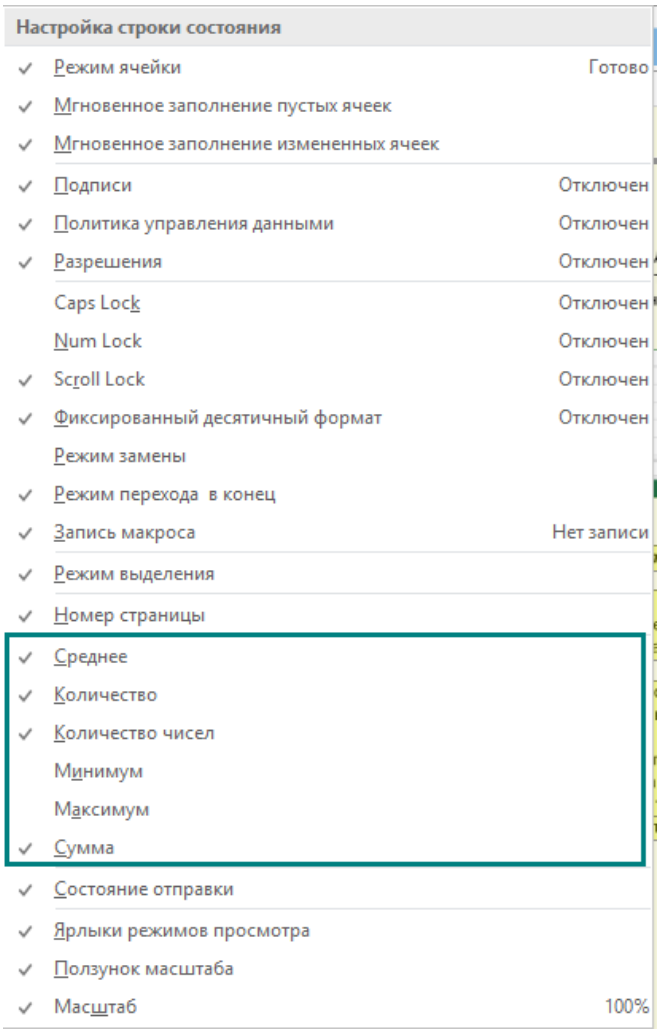


Рис. 1.6 Контекстне меню Рядку стану

Відображення інформації та командних кнопок в Рядку стану можна налаштувати з контекстного меню (рис. 1.6), клацнувши необхідні параметри. Особливо корисними можуть виявитися деякі поточні підрахунки без введення формул у клітинки аркуша. Наприклад, сума або кількість чисел в виділених клітинках. Потрібні вбудовані функції можна обрати з переліку на панелі настроювання рядка стану (рис. 1.6), встановивши необхідні прапорці. Після чого результати обчислення по цих функціях будуть відображатися у рядку стану після виділення певних клітинок. Необхідно виділити неменше двох клітинок. Це можуть бути будь які клітинки у одному чи декількох стовпцях або рядках.

2. Початок роботи з Microsoft Excel 2016

При кожному запуску програми MS Excel в оперативній пам'яті комп'ютера автоматично створюється порожній файл нового документа і в вікні відображається порожній аркуш. За замовчанням цьому файлу присвоюється ім'я **Книга 1**, яке відображається в Рядку заголовка. У загальному випадку для запуску програми Excel з метою створення порожнього файлу необхідно виконати команду **Пуск – Усі програми – Excel 2016**. Швидшим може виявитися запуск програми з **Панелі швидкого запуску** або з за допомогою ярлика на робочому столі

Існує кілька можливостей створення нової робочої книги, з яких розглянемо три основні:

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

1. **Нова книга**. Порожній файл нової робочої книги створюється в тому випадку, якщо необхідно задати всі параметри форматування самостійно, але для цього вимагається досить тривалий час і досвід роботи.

2. **Шаблони**. Шаблон - це книга-заготовка, в який потрібно додати актуальні дані. Шаблон забезпечить їх автоматичне обчислення та форматування, яке використовується в документах стандартного типу, таких, як прайс-листи, календарі, різноманітні звіти, тощо. Використання шаблону виправдано, коли користувач слабо уявляє собі структуру нового документа, або якщо в діловодстві часто використовуються документи типової структури.

3. **З існуючого документа**. Вже існуючий файл документа використовується для створення нового документа. Наприклад, для створення звіту про роботу за поточний рік можна використовувати звіт за попередній рік, змінивши в ньому лише деякі показники. При цьому основний зміст документа і його форматування залишається незмінним. Такий підхід дозволяє істотно скоротити час на підготовку документа.

Створення нової робочої книги

Створити нову порожню робочу книгу також можна у кілька способів.

1. Відкрити папку, в якій передбачається зберігати нову робочу книгу. На вільному місці у межах папки клацнути правою кнопкою миші та з контекстного меню (рис. 1.7) обрати **Створити – Аркуш Microsoft Excel**. В папці з'явиться новий документ з ім'ям **Новий аркуш Microsoft Excel**, яке буде виділено синім кольором. Це означає, що документу можна відразу задати нове змістовне ім'я. Змінювати або видаляти розширення файлу ***.xlsx**, якщо воно відображається в кінці імені неможна. Цей спосіб зручний, тому що він дозволяє не лише створити нову робочу книгу, але і зберегти її в визначеній папці під певним ім'ям. Тобто виконати відразу кілька необхідних дій.

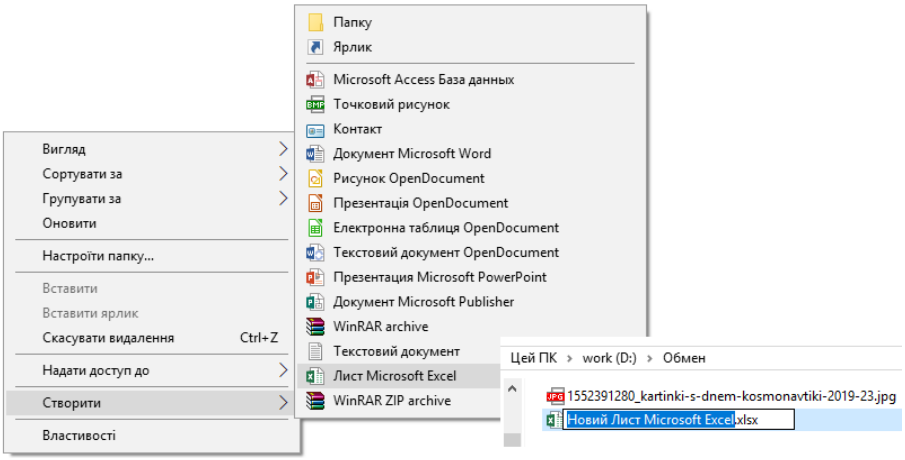


Рис. 1.7 Створення нової робочої книги через контекстне меню

2. Клацнути значок програми Excel у будь якому доступному місц. Відкриється вікно, в якому можна обрати мініатюру **Пуста книга** (рис. 1.8). Якщо клацнути по мініатюрі, то на екрані відкриється пустий аркуш нової робочої книги. Зауважимо, що ця робоча книга зберігається тільки в оперативній пам'яті комп'ютера. Для подальшої роботи її буде необхідно зберегти на будь-якому диску.

Перелік лекцій

3. Якщо програма вже запущена, відбувається робота з поточною робочою книгою, і потрібно паралельно створити нову книгу, то у цьому випадку для створення нової книги необхідно на вкладці **Файл** вибрати опцію **Створити**, в центрі вікна двічі клацнути мініатюру **Пуста книга** (рис. 1.8). Однак простіше це завдання можна вирішити комбінацією клавіш **Ctrl + N**. Після цього програма Excel, так само як і в попередньому випадку, в оперативній пам'яті створить пусту книгу, повністю готову до роботи. По суті у всіх розглянутих випадках використовується шаблон нової пустої книги, параметри якої задані за замовчанням, наприклад, кількість аркушів, накреслення і розмір шрифту тощо.

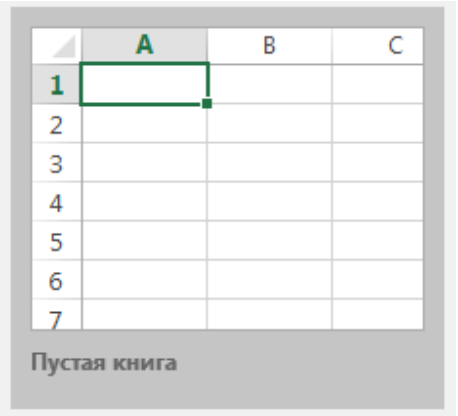


Рис. 1.8 Мініатюра Пуста книга

Збереження файлу нового документа

Поперед усе необхідно завжди зберігати на диску новий документ відразу після його створення. Це правило слід виконувати незалежно від способу створення документа.

Дії зі збереження нової книги залежать від способу її створення. Дії для збереження Нової робочої книги, або книги на основі шаблону наступні:

1. Натиснути кнопку **Зберегти** у вигляді дискети на Панелі швидкого доступу, або виконати команду **Файл – Зберегти**, або натиснути клавіші **Shift + F12**. Відкриється вікно **Зберегти як** (рис. 1.9), в якому необхідно відкрити потрібну папку.

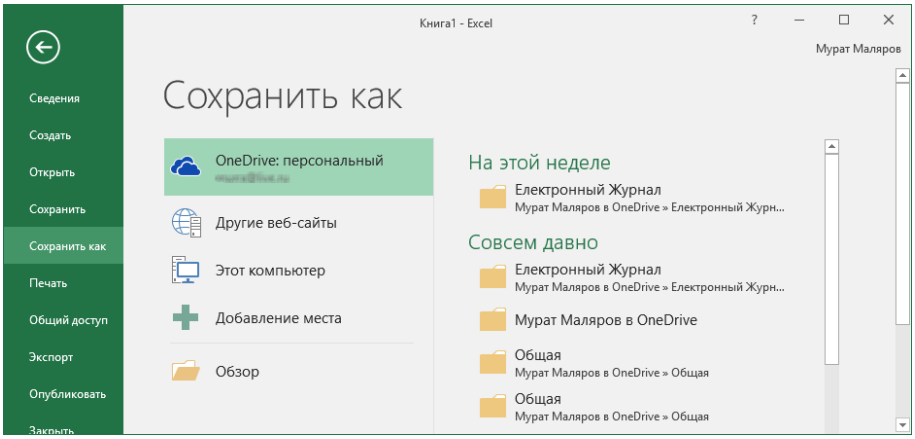


Рис. 1.9 Вікно Зберегти як

2. Папка відкриється у вікні **Збереження документа** де у поле **Ім'я файлу** необхідно ввести ім'я документа. Зазвичай, ім'я файлу повинно відповідати змісту документа. Якщо ніяких особливих вимог до документа не пред'являється, то можна погодитися з типом файлу за замовчанням ***.xlsx**. Якщо, наприклад, передбачається продовжити роботу з документом, використовуючи попередню версію програми Excel, то необхідно вибрати тип файлу **Книга Excel 97-2003 (*.xls)**, інакше файл цими програмами взагалі відкрити буде неможливо. Потрібно бути готовим і до того, що документи, відкриті в програмах попередніх версій, можуть втратити певні елементи форматування.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

3. Клацнути по кнопці **Зберегти**. Файл нового документа буде збережений на диску, а його ім'я з'явиться в рядку заголовка вікна програми. Для збереження документа на основі існуючого файлу необхідно виконати команду **Файл – Зберегти як**, після чого відкриється вікно **Зберегти як** (рис. 1.9), в якому необхідно виконати усі попередні пункти.

3. Робота з робочою книгою Excel

За замовчанням робоча книга складається з одного робочого аркушу з ім'ям **Аркуш 1**, тому дуже часто виникає потреба додавання та перейменування аркушів у книзі. Для перейменування аркушу, потрібно двічі клацнути по його ярличку (вкладці) у нижній частині вікна або використати відповідну команду з контекстного меню (рис. 1.10). Ім'я аркуша виділиться, відразу можна ввести нове ім'я. Крім того, ярличок можна зафарбувати, що дозволить швидко знаходити і перемикатися на потрібний аркуш. Для цього з контекстного меню треба виконати команду **Колір вкладки** і з палітри вибрати потрібний колір.

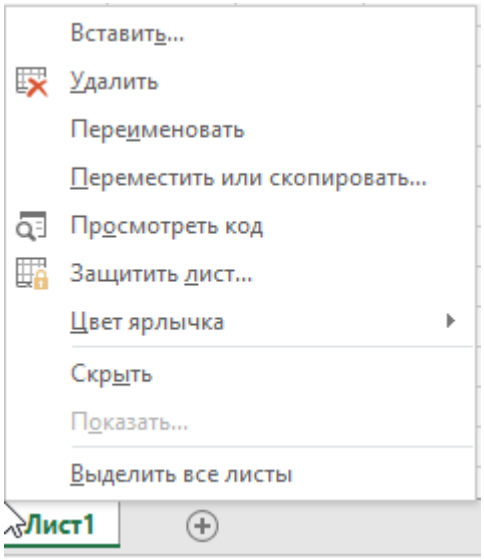


Рис. 1.10 Контекстне меню ярлика аркуша

Для додавання робочих аркушів у книгу необхідно клацнути по кнопці праворуч від їх імен (рис. 1.10). Новий аркуш додається праворуч від поточного листа. Для переміщення листа в межах книги необхідно перетягнути його ярличок лівою кнопкою миші. Місце вставки буде позначено трикутником. Щоб видалити активний аркуш, потрібно з контекстного меню виконати команду **Видалити**.

Введення даних

Вводити дані можна тільки в активну клітинку. Одна клітинка на аркуші завжди активна. Вона виділяється рамкою (рис. 1.11), яка відіграє роль табличного курсора. Перейти у іншу клітинку можна:

- за допомогою курсорних клавіш клавіатури на одну клітинку в будь-який бік;
- клацанням мишею на будь-якій потрібній клітинці;
- натисканням клавіши **Enter** для переходу на клітинку вниз;
- натисканням клавіши **Tab** для переходу на клітинку вправо.

Перелік лекцій

При наведенні курсора миші на виділену клітинку він може змінювати свій вигляд. При наведенні курсора на внутрішню частину клітинки він набуває вигляду білого хрестика. Якщо клацнути мишею, то клітинка виділяється рамкою (рис. 1.11, клітинка **B2**), а її адреса, що складається з номера стовпця і номера рядка, відображається в спеціальному полі в правій частині рядка формул.

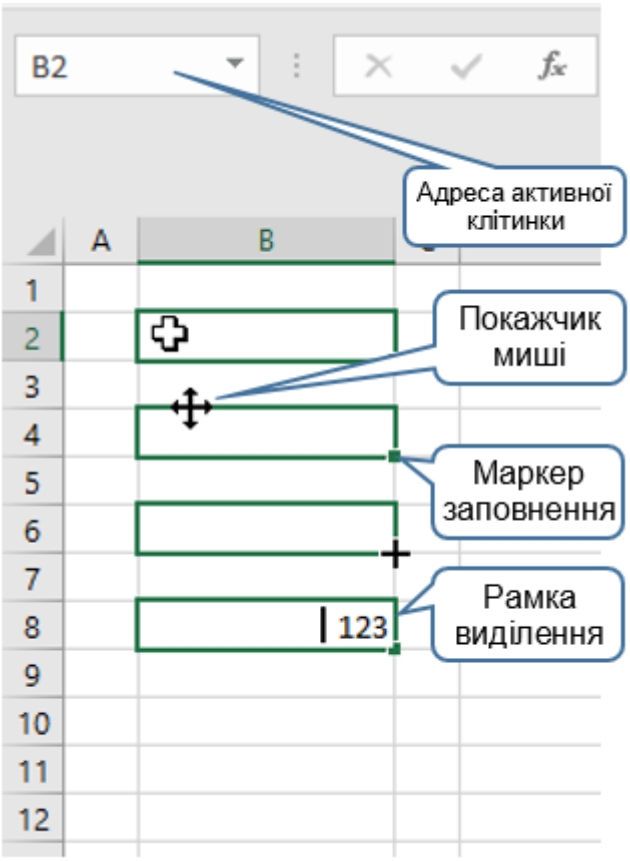


Рис. 1.11 Вигляд табличного курсору в різних режимах введення

При протягуванні курсора по декількох клітинках при натиснутій лівій кнопці миші виділяється діапазон клітинок. Діапазон клітинок виділяється загальною рамкою і кольором. У виділеному діапазоні активною є ліва верхня клітинка.

На дані, розташовані в клітинках виділеного діапазону, можна посилатися в формулах як на єдине ціле, наприклад, при підрахунку суми чисел для кількох клітинок. Для зручності діапазону замість адреси може бути присвоєно довільне ім'я. При наведенні вказівника миші на рамку виділення клітинки він набуває вигляду чотиреспрямованої стрілки (Рис. 1.11, клітинка **B4**). При перетягуванні лівою кнопкою миші вміст клітинки буде переміщено в іншу клітинку. Якщо перетягування виконати правою кнопкою миші, то в кінці з контекстного меню можна вибрати команду **копіювати** або **перемістити**.

При наведенні покажчика миші на маркер заповнення в нижньому правому куті рамки він набуває вигляду чорного хрестика (рис. 1.11, клітинка **B6**). При протягуванні таким курсором при натиснутій лівій кнопці зміст активної клітинки копіюється в суміжні клітинки. Якщо клацнути по клітинці двічі, то курсор перетвориться в текстовий курсор (вертикальна миготлива лінія, рис. 1.11, клітинка **B8**). Вміст клітинки після цього можна редагувати. Курсор в клітинці переміщається клавішами управління курсором на клавіатурі або мишею. Вміст клітинки відображається як в самій клітинці, так і у рядку формул. Щоб виділити стовпець або рядок повністю, досить клацнути по номеру стовпця (рядка). Протягуванням покажчика по номерах можна вибрати кілька стовпців або рядків, які йдуть підряд. Для виділення довільних клітинок, стовпців або рядків необхідно клацати по ним утримуючи клавішу **Ctrl**.

Введення та редагування числових та текстових даних у Excel

Перед створенням таблиці доцільно продумати і навіть намалювати на папері її макет. Необхідно визначити кількість стовпців, їх заголовки, тип даних в клітинках

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

стовпців. Кожен стовпець повинен містити прості дані одного типу. Крім того, якщо в таблиці виконуються обчислення, то необхідно записати всі формули. Клітинки можуть містити дані, які відносяться до одного з трьох типів: **текст**, **число** або **формула**, а також залишатися порожніми.

> Excel при збереженні робочої книги записує у файл тільки прямокутну область робочих аркушів, яка примикає до лівого верхнього кута і яка містить клітинки з даними. Інформація про порожні клітинки поза цією областю в пам'яті комп'ютера не зберігається.

Введення даних здійснюють безпосередньо в активну клітинку або в Рядок формул, який розташовується у верхній частині вікна програми безпосередньо під стрічкою (рис. 1.2 пункт 7). Вміст Рядка формул завжди відповідає вмісту активної клітинки, так що обидва методи введення рівноправні. Місце введення відзначається текстовим курсором. Якщо почати введення натисканням алфавітно-цифрових клавіш, то наявні дані в активній клітинці замінюються текстом, який вводиться. Якщо клацнути на рядку формул або двічі на активній клітинці, то старий вміст клітинки не видаляється і з'являється можливість його редагування. Дані, які вводяться, відображаються як в клітинці, так і в рядку формул. Щоб завершити введення і зберегти нові дані, використовують клавішу **Enter**. Щоб відхилити внесені зміни і відновити попереднє значення клітинки, використовують клавішу **Esc**. Для очищення виділеної клітинки або виділеного діапазону найпростіше використовувати клавішу **Delete**.

Тип даних, розміщених в клітинці, визначається автоматично при їх введенні. Якщо ці дані можна інтерпретувати як число, наприклад, при введенні цифр, Excel так і робить. В інших випадках дані інтерпретуються як текст. Введення формули завжди починається зі знаку дорівнює **=**.

Для обчислень застосовують три принципово різних формату запису чисел: **звичайний числовий**, **фінансовий** і **формат дати**.

Звичайний числовий формат використовують для запису різних чисел, що використовуються для обчислень, наприклад, кількості будь-якої продукції, процентної ставки, віку тощо. Числа вводяться і виводяться, як правило, у природній формі. Під час введення числа із фіксованою комою ціла частина від дрібної відокремлюється комою. Гранично припустима точність – до 30 знаків після коми. За замовчанням для відображення текстів та чисел (цілих та із фіксованою комою) використовується формат «Загальний».

Дуже великі (маленькі) числа зручно вводити в так званій нормальній формі: ±МЕ±Р, де М і Р – мантиса і порядок, а Е – основа системи, що дорівнює 10 (наприклад 2,5E+12 = $2,5\times10^{12}$ або -3,6E-8 = $-3,6\times10^{-8}$). Для відображення такого числа Excel призначає формат «Експонентний».

Простий дріб можна вводити, починаючи з цілої частини, а потім через пропуск ввести дрібну частину. Формат відображення буде «Дріб».

Для запису грошових сум використовуються **фінансові формати**, які відрізняються зазначенням грошової одиниці (наприклад, грн.) і фіксованим числом знаків після десятиного роздільника (зазвичай два числа, які показують копійки). Зрозуміло, що грошові суми можна записувати і звичайними числами, але при роботі

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

з фінансовою форматом програма інакше підходить до округлення чисел і контролює, щоб у результаті численних операцій з дуже великими сумами не загубилася жодна копійка.

Календарні дати прийнято вводити за **форматом дати** у вигляді ДД.ММ.РР (06.07.18). Тут ДД – це календарне число, ММ – місяць, РР – рік. Замість крапки можна використовувати також косу рису - «/». Для відображення дати призначається формат «Дата». Час доби, як правило, вводять за форматом ГГ:ХХ:СС (15:23:22). Тут ГГ – години, ХХ – хвилини, СС - секунди. Для відображення часу призначається формат «Час».

> Дати в Excel зберігаються у вигляді чисел і являють собою кількість днів, що минули з 0 січня 1900 року. Час обробляється програмою, як дрібне число між 0 і 1. Щоб побачити це, можна змініть формат клітинок де зберігається дата або час на Загальний.

Текстові формати використовують для запису текстових рядків і цифрових даних, які не використовуються для обчислень. Щоб ввести текст, потрібно просто набрати його в клітинці. Якщо необхідно представити у виді тексту будь-який набір символів (наприклад, цифри), то введення починається з апострофа (`).

До таких даних належать: поштові індекси, номери телефонів, реєстраційні номери та інші цифрові позначення. Наприклад, хоча номера телефонів записують цифрами, математичні операції з ними не мають ніякого сенсу.

В клітинку можна додати примітку. Її текст вводиться після використання команди **Вставити примітку** з контекстного меню. Клітинка з приміткою (коментарем) позначається невеликим червоним трикутником у її правому верхньому куті. Примітка ж залишається невидимою і з'являється тільки після виділення комірки.

Щоб представити клітинку з інформацією в більш привабливому вигляді, до неї також можна додати параметри форматування.

Операції форматування у Excel

Настройка розмірів стовпців і рядків. Для формування акуратних і зручних для аналізу і читання табличних документів доцільно привести розміри її клітинок відповідно до форматів тих даних, які в них будуть розміщуватися.

Стосовно до стовпців цю процедуру найпростіше проводити за допомогою покажчика миші. Якщо встановити покажчик миші на праву межу стовпця в рядку з іменами стовпців, то він приймає вигляді двоспрямованої горизонтальної стрілки, буксируючи яке можна плавно змінювати ширину стовпця в межах від 0 до 255 знаків. (наприклад для зміни стовбця **G**

). Для установки певної

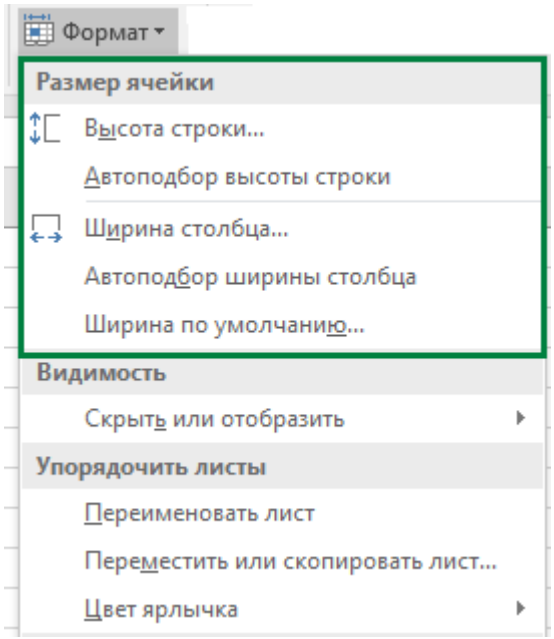


Рис. 1.12 Меню Формат

Перелік лекцій

ширини стовпця рамку виділення потрібно встановити у будь-яку його клітинку та з підменю команди **Головна - Формат** вибрати відповідну команду (рис. 1.12) та у вікні, що з'явиться ввести потрібне значення ширини. Це значення приблизно дорівнює кількості символів стандартного шрифту, які умістяться в комірці. Для використання автоматичного підбору ширини треба заздалегідь виділити заповнені клітинки таблиці. Автоматично встановлюється мінімальна ширина стовпця, яка забезпечує повне відображення вмісту самої широкої клітинки цього діапазону. Для здійснення останньої операції можна також двічі клацнути мишею на правій межі заголовка стовпця.

> Ширина стовпця не впливає на розрядність даних, які зберігаються клітинці та не впливає на точність обчислень. Однак вона впливає на те, як будуть відображатися дані на екрані. При незбігу розрядності даних із шириною стовпця їх відображення на екрані може бути зрізаним. При цьому числові дані можуть відображатися або в нормальній формі або заміщатися символами - **#######.**

Аналогічно встановлюється висота рядків, значення якої задається у пунктах від 0 до 409 (рис. 1.12). Автоматична висота рядка, що встановлюється, забезпечує повне відображення вмісту самої високої клітинки цього рядка. Аналогічно цю операцію можна виконати також двічі клацнувши мишею на нижній межі заголовка рядка.

Форматування комірок електронної таблиці. Текстові дані за замовчанням вирівнюються по лівому краю клітинки, а числа - по правому краю. Спосіб виконання обчислень не залежить від того, яким способом дані відформатовані, але для людини зовнішній вигляд даних дозволяє швидко визначити їх тип.

Україна
342
12.12.2018
2,34E+13

Вирівнювання може бути змінено за бажанням користувача командами групи вирівнювання (фрагмент на рис. 1.13) на вкладці **Головна**, або вибором відповідних опцій в діалозі, який відкривається клацанням по кнопці в правому нижньому кутку групи. У діалоговому вікні **Формат клітинок** (рис. 1.13) на вкладці Вирівнювання можна встановити вертикальне і горизонтальне вирівнювання даних в клітинці, а також встановити прапорець для перенесення слів в клітинці по словам. Якщо цей прапорець скинутий, то текст в клітинці буде розташовуватися в один рядок, і може виходити за рамку клітинки. Для примусового перенесення тексту в клітинці необхідно натиснути клавіші **Alt + Enter**.

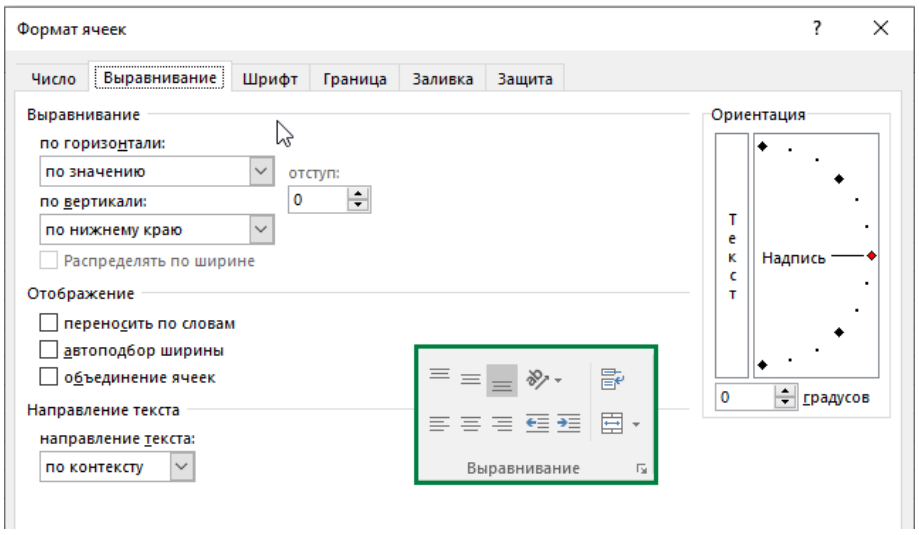


Рис. 1.13 Вікно Формат клітинки, вкладка Вирівнювання

Перелік лекцій

Взагалі параметри форматування для виділеної клітинки встановлюються вибором зі списку в групі **Число** вкладки **Головна** (фрагмент на рис. 1.14). Щоб змінити формат відображення даних в активній клітинці або в обраному діапазоні клітинок, просто встановлюють новий формат даних в списку.

Більш гнучке налаштування досягається за допомогою діалогового вікна **Формат клітинок** з шістьма вкладками: «Число», «Вирівнювання», «Шрифт», «Границя», «Вид» та «Захист», яке відкривається кнопкою в правому нижньому кутку групи **Число**.

Вкладка «**Число**» (рис. 1.14) використовується для вибору будь-яких форматів для найкращого відображення вмісту комірки.

Формат «Загальний» використовується для відображення як текстових, так і числових даних довільного типу. Кращим для подання дійсних чисел із заданою точністю є формат «Числовий».

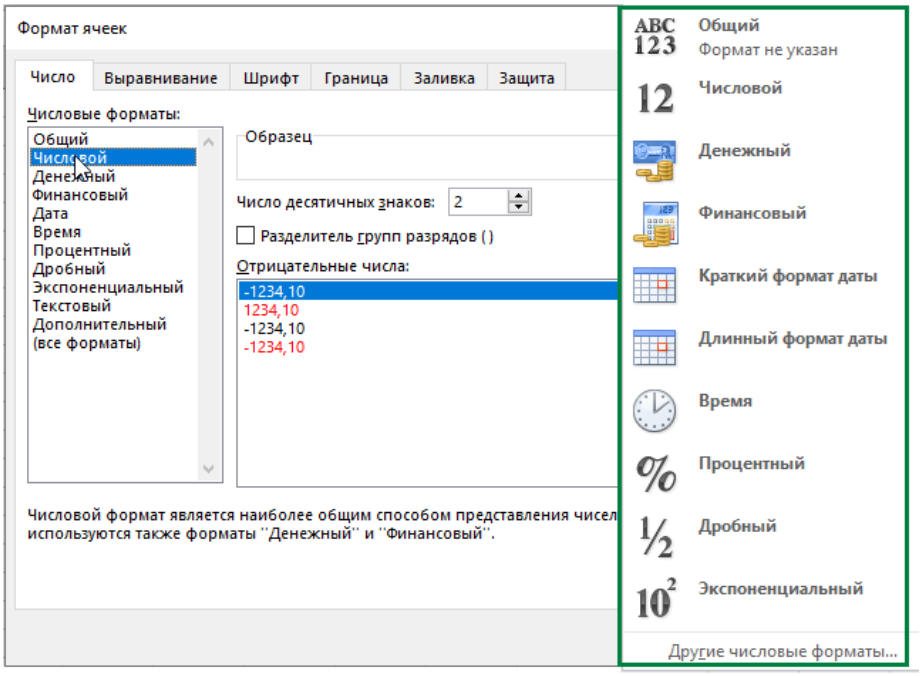


Рис. 1.14 Вікно Формат клітинки, вкладка Число

Формати «Грошовий» та «Фінансовий» застосовуються для відображення грошових величин із зазначенням виду валюти (грн., руб., $ тощо). Особливістю формату «Фінансовий» є те, що він додатково вирівнює значення за десятковим роздільником.

Формати «Дата» і «Час» зручні для подання дат і часу в різних варіаціях. Наприклад, якщо в комірку введена дата 06.07.99, то її можна відобразити як 6.7; Липень 99; 6 Липень, 1999 тощо.

У форматі «Процентний» значення комірок множаться на 100 і виводяться на екран із символом відсотка (%) і заданим числом знаків після коми.

Формат «Дріб» використовується для зображення дробі з різними частками (2, 4, 8, 16, 10 і 100) та різним числом цифр (1, 2 і 3).

Формат «Експонентний» застосовують для подання чисел в нормальній формі.

Формат «Текстовий» відображає значення будь-якого типу як текстове (вирівнює по лівому краю комірки).

Перелік лекцій

Формат «Додатковий» використовують, якщо в комірку введені поштові індекси, табельні та телефонні номери.

Нарешті, параметр «Усі формати» містить ряд шаблонів, що дають користувачеві змогу створювати нові формати.

Вкладка «Шрифт» надає користувачеві широких можливостей шрифтового оформлення робочого листка, а саме: вибір типу шрифту (Time New Roman, Arial тощо) і варіанта його накреслення (звичайний, курсив, напівжирний або напівжирний курсив); установлення або вибір із списку потрібного розміру шрифту у пунктах; завдання параметрів підкреслення (без підкреслення, з підкресленням одинарною, подвійною або штриховою лініями); керування спеціальними ефектами шрифту (із закресленням, верхній індекс або нижній індекс).

Вкладка «Граница» використовується для вибору типу та кольору ліній, що створюють контури комірок. Лінії можуть бути одинарними, подвійними, штриховими тощо; їхня товщина варіюється.

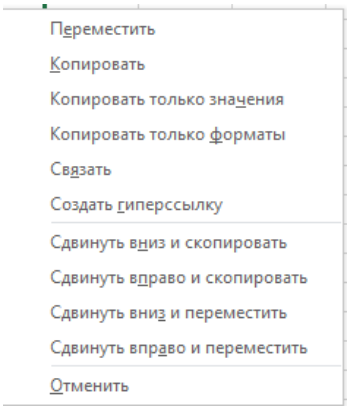
Вкладка «Вид» дозволяє сформувати електронні таблиці, у яких потрібна інформація виділяється кольором і візерунками. Для поліпшення сприйняття найважливіших елементів таблиці може бути використане і колірне штрихування.

Вкладка «Захист» забезпечує скасування захисту для обраних комірок після установки захисту для всього листка за командою **Рецензування - Захистити лист**.

Копіювання і переміщення вмісту клітинок

У процесі створення документів часто доводиться змінювати порядок розміщення рядків, стовпчиків і клітинок, що містять дані. Ця процедура виконується за допомогою перетаскування клітинок (діапазону клітинок) клавішами миші. В залежності від вибраного способу перетаскування, може здійснюватися як копіювання (переміщення) всієї клітинки, так і тільки її даних (вмісту, примітки та параметрів форматування), або тільки вмісту, або тільки параметрів форматування.

Копіювання і переміщення вмісту клітинок в Excel можна здійснювати: методом перетягування, за допомогою маркера заповнення або через буфер обміну. При роботі з невеликою кількістю клітинок зручне перетягування або використання маркера заповнення. При роботі з великими діапазонами клітинок зручніше використовувати буфер обміну.



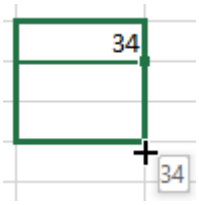
Щоб методом перетягування скопіювати або перемістити вміст поточної клітинки (виділеного діапазону), потрібно навести курсор миші на рамку клітинки, він зміниться на чотиреспрямовану стрілку (Рис. 1.11, клітинка **B4**). Тепер вміст клітинки можна перетягнути в будь-яке місце робочого аркуша, адреса вставки відображається в підказці. Якщо при перетаскуванні натиснути на клавішу **Ctrl**, то відбудеться копіювання. Щоб контролювати цю операцію рекомендується використовувати спеціальне перетягування за допомогою **правої** кнопки миші. В цьому випадку з контекстного меню

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

можна вибрати конкретну операцію: переміщення або копіювання даних клітинки; копіювання або тільки вмісту, або тільки параметрів форматування клітинки; копіювання (переміщення) клітинки зі зсувом заповнених клітинок робочого листка вниз або праворуч.

Копіювання тільки параметрів форматування комірки (комірок) можна виконати також за допомогою команди копіювання формату. Для цього потрібно виділити вже відформатовані комірки і клацнути на кнопці **Головна – Формат за зразком** . З'являється покажчик миші у вигляді кісточки. Тепер потрібно клацнути на першій комірці того діапазону, якому слід надати формат. Якщо треба надати формат однієї комірки цілому діапазону, то потрібно кісточкою провести по всіх його комірках.



Для копіювання даних в сусідні клітинки зручно користуватися маркером заповнення. Для цього необхідно виділити клітинку з даними і протягнути за маркер заповнення по сусідніх клітинках. Також копіювати дані з однієї комірки у сусідні можна за командою **Головна – Заповнити** . При цьому треба виділити вибрану комірку разом із інтервалом комірок, який треба заповнити. Далі за допомогою підменю команди **Головна – Заповнити** (рис. 1.15). задається напрямок заповнення, в залежності від того, у яку сторону відбувалося виділення: Вверх, Вниз, Ліворуч або Праворуч.

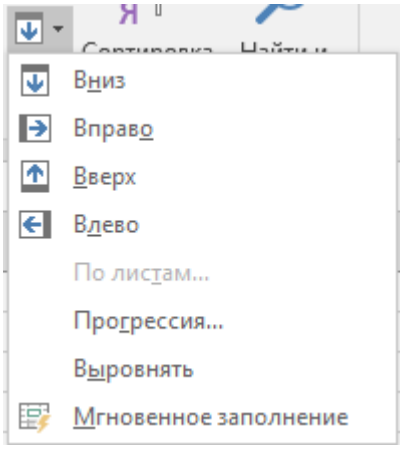


Рис. 1.15 – Підменю команди Заповнити

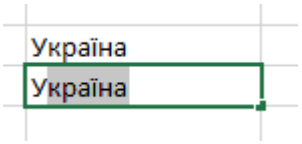
При використанні буфера обміну спочатку необхідно виділити діапазон клітинок, які копіюються (вирізаються) і натиснути кнопку **Копіювати** (або **Вирізати**), на вкладці **Головна** розділу **Буфер обміну** або вибрати команду **Копіювати** (**Вирізати**) з контекстного меню. Вставка даних на робочий аркуш можлива лише безпосередньо після їх переміщення в буфер обміну. Спроба попередньо виконати будь-яку іншу операцію призводить до скасування процесу копіювання або переміщення. Однак втрати даних не відбувається, оскільки "вирізані" дані видаляються з місця їх початкового розміщення тільки тоді, коли вставка буде успішно завершена. Місце вставки вказується шляхом виділення клітинки, яка відповідає верхньому лівому кутку діапазону або виділенням діапазону, який за розмірами **точно** відповідає тому діапазону, який копіюється (переміщається).

Вставка виконується кнопкою або командою **Вставити** з контекстного меню.

Автоматизація введення даних

Таблиці часто містять повторювані або однотипні дані. Excel надає засоби для автоматизації введення таких даних. До них відносяться: **автозавершення**, **автозаповнення** і **автозаповнення формулами**.

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua



Автозавершення застосовують при введенні в клітинки одного і того ж стовпця текстових рядків, серед яких є такі, що повторюються. При введенні першої літери тексту в поточну клітинку програма перевіряє відповідність введених символів рядкам, які містяться в цьому стовпці вище. Якщо виявлено збіг, введений текст автоматично доповнюється. Наприклад, при введені у нижню клітинку літери У, текст доповнюється літерами країна (виділені сірою заливкою) , що в цілому відповідає тексту у верхній клітинці. Клавіша **Enter** підтверджує операцію автозавершення, в протилежному випадку введення можна продовжувати, не звертаючи уваги на запропонований варіант. Якщо в таблиці є слова, які починаються з декількох однакових символів, то автозавершення спрацьовує з символу, який відрізняється в цих словах. Також автозавершення спрацьовує тільки при заповнені верхньої або нижньої суміжної клітинки.

Особливість засобу автозавершення полягає в тому, що він також веде список усього тексту, який було введено у визначений стовпчик. Цей список викликається на екран за командою **Вибрати зі списку, що розкривається** контекстного меню клітинки стовпця перед введенням тексту. Клацання мишею на елементі списку забезпечує його введення в клітинку. Метод автозаміни дає змогу автоматично усувати типові помилки в процесі введення текстових даних. Його можна також використовувати і як засіб прискорення введення даних, що мають велику довжину.

> Режим, який дозволяє автоматично повторювати тексти під час їх введення в комірки одного стовпчика, включається (виключається) перемикачем Автозавершення значень комірок (Файл – Параметри – вкладка Додатково).

Автозаповнення. Автозаповнення клітинок – це досить гнучкий і зручний інструмент автоматичного введення числових і текстових даних, календарних дат, тощо. Автоматичне заповнення діапазону клітинок робочого листка роблять за допомогою **маркера заповнення**.

> Маркер заповнення - це квадратик, розташований в правому нижньому куті рамки виділення (Рис. 1.11, клітинка B4). Якщо маркер заповнення відсутній, то треба встановити прапорець **Дозволити маркери заповнення та перетягування комірок** (**Файл – Параметри** – вкладка **Додатково**).

Щоб зробити автоматичне заповнення діапазону клітинок, рамку виділення встановлюють на клітинку із введеною інформацією. Після наведення на маркер заповнення покажчика миші він перетвориться в чорний хрестик (Рис. 1.11, клітинка B6). Тепер потрібно зробити клацання мишею і, не відпускаючи клавіші, провести в потрібному напрямку уздовж рядка або по стовпцю. Після того, як клавіша буде відпущена, дані початкової клітинки «розмножаться» по всіх інших клітинках.

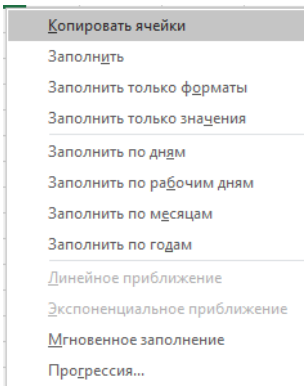
Якщо початкова клітинка містить числові або довільні текстові дані, то відбувається їх копіювання. Для того, щоб використовувати автозаповнення для заповнення клітинок датами, потрібно ввести в комірку календарну дату. Після перетаскування маркера заповнення виділені клітинки заповняться послідовністю дат, що є наступним за введеною.

Україна	123	12.12.2018	12:32
Україна	123	13.12.2018	13:32
Україна	123	14.12.2018	14:32
Україна	123	15.12.2018	15:32
Україна	123	16.12.2018	16:32

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

При заповнення клітинок часом доби, після перетягування маркера заповнення клітинки заповнюються послідовністю часу з різницею в одну годину. Такі параметри мають назву стандартних параметрів заповнення.



При буксируванні маркера заповнення за допомогою правої клавіші миші можна виконати ряд дій за командами спеціального контекстного меню, що активізується після відпускання клавіші. Перша група команд призначена для виконання копіювання даних комірки і вибіркового заповнення. Команда **Заповнити** заповнює виділені комірки членами арифметичної прогресії з кроком, рівним одиниці. Команда **Заповнити формати** – параметрами форматування, а команда **Заповнити значення** – тільки значеннями початкової комірки.

Друга група команд призначена для створення послідовностей календарних дат. Остання команда контекстного меню **Прогресія** активізує однойменне вікно (рис. 1.16), засобами якого забезпечується автозаповнення діапазону комірок різними прогресій.

Арифметична. Значення кожної наступної комірки дорівнює значенню попередньої плюс значення кроку.

Геометрична. Значення кожної наступної комірки дорівнює значенню попередньої, помноженому на значення кроку.

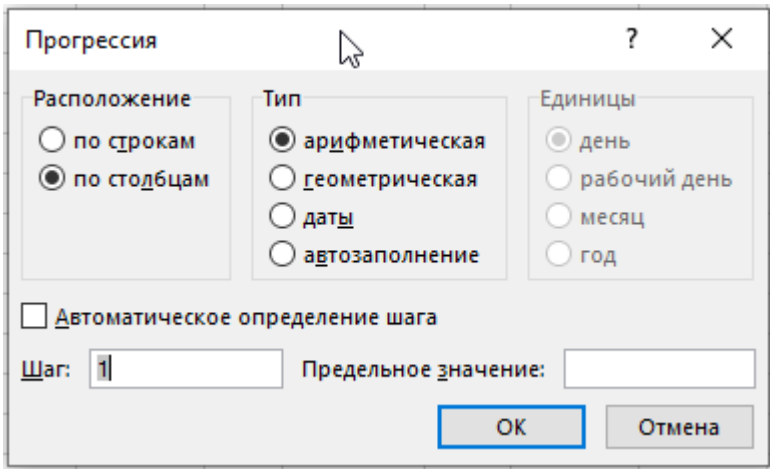
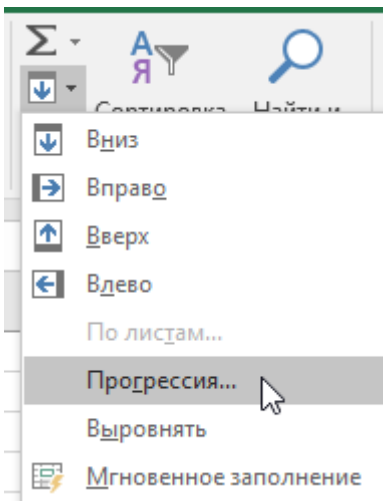


Рис. 1.16 Діалогове вікно Прогресія

Дати. Заповнює комірки послідовністю дат. Під час вибору цього параметру у групі **Одиниці** вказують вимір кроку, із яким буде змінюватися дата: день, робочий день, місяць або рік.

Автозаповнення. Заповнює комірки автоматично, стосовно списків автозаповнення, які перераховані у вікні **Списки** вікна **Параметри**.

При виборі двох перших типів прогресії значення кроку вводять у поле **Крок**. Операція завершується клацанням на кнопці **ОК**.



Автозаповнення діапазону комірок членами прогресії можна виконати і іншим способом. Для цього у першу клітинку діапазону вводять перше значення прогресії. Потім рамку виділення встановлюють на цю комірку і виконують команду **Головна – Редагування – Заповнити**.

Активізується вікно **Прогресія**» (рис. 1.16), в якому задають: тип прогресії, значення прирощення (крок) і

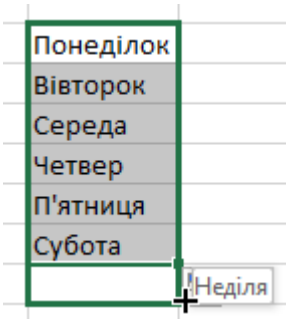
Перелік лекцій

обов'язково граничне значення (останній член прогресії). В групі **Розташування** встановлюють режим *по строкам* або режим *по столбцам*, у залежності від того, як потрібно розташовувати елементи прогресії.

Якщо перед виконанням команди разом із початковою коміркою виділити діапазон комірок, в яких потрібно продовжувати прогресію, поле **Граничне значення** можна залишити пустим. Після клацання на кнопці **ОК** значеннями прогресії будуть заповнені всі виділені комірки.

Автозаповнення клітинок елементами списку. При інсталяції Excel автоматично завантажується декілька списків (рис. 1.17), зазвичай це дані, що мають регулярний характер, наприклад, назви місяців, днів тижня тощо. Наприклад, можна автоматично вводити в комірки назви днів тижня і місяців року: *Понеділок, Вівторок, ... Неділя*; *Січень, Лютий, ... Грудень*.

Для цього в обрану комірку таблиці вводять перше значення списку, наприклад: *Понеділок*. Потім покажчик миші поєднують з маркером заповнення і «буксируванням» покажчика у вигляді чорного тонкого хрестика виділяють діапазон комірок стовпчика або рядка, який за розміром відповідає кількості даних, які треба ввести. Клавішу миші відпускають, і в комірках виділеного діапазону відображаються значення зі списку.



Якщо виділено комірок більше, ніж значень у списку, то список буде повторюватися спочатку.

Щоб ефективно використовувати списки слід спочатку ознайомитися з наявними списками, а при відсутності потрібного створити нові списки або вилучити вже існуючий. Для цього необхідно виконати команду **Файл – Параметри – Додатково –** кнопка **Змінити списки**. У полі **Списки**, можна переглянути вже існуючі списки та ввести новий список. Для створення будь-якого нового списку треба вибрати **НОВИЙ СПИСОК** та ввести елементи списку у поле **Елементи списку**, або через кому або в стовпець. До складу списків, які діють, його включають клацанням мишею на кнопці **Добавити**. Клацанням мишею на кнопці **Видалити** будь-який вибраний список вилучається. Списки можуть бути створені будь-якою мовою з будь-яких символів.

Автозаповнення формулами виконується так само, як автозаповнення числами. В ході автозаповнення до уваги береться характер посилань у формулі: **відносні** посилання автоматично змінюються відповідно до відносного розташуванням копії і оригіналу, а **абсолютні** - залишаються без змін.

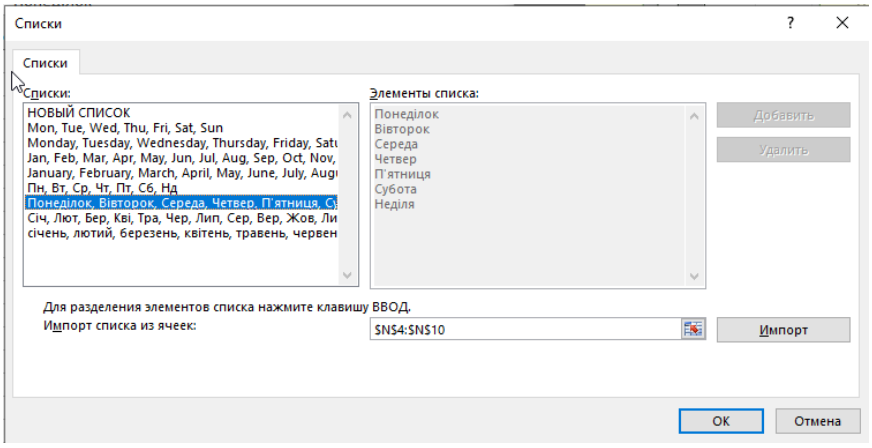
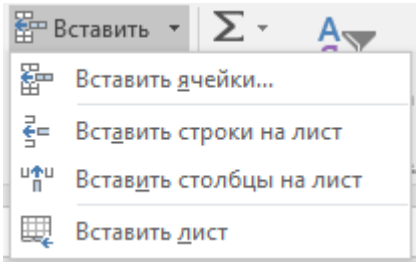


Рис. 1.17 Діалогове вікно Списки

Перелік лекцій

Вставлення і вилучення стовпців і рядків робочого листа



Для вставки порожнього стовпця (рядка) спочатку треба виділити стовпець (рядок) електронної таблиці, **перед** яким є намір його розмістити, і вибрати відповідну команду з підменю команди **Головна – Вставити**. За цією самою командою можна вставити також довільне число стовбців (рядків), але для цього заздалегідь потрібно виділити відповідне число стовпців (рядків) таблиці.

Вставлення стовпців (рядків) супроводжується зсувом заповнених стовпців (рядків) електронної таблиці праворуч (вниз).

Порожні клітинки вставляються в електронну таблицю за командою **Вставити клітинки**, яка активізує вікно «Додавання клітинок» (рис. 1.18).

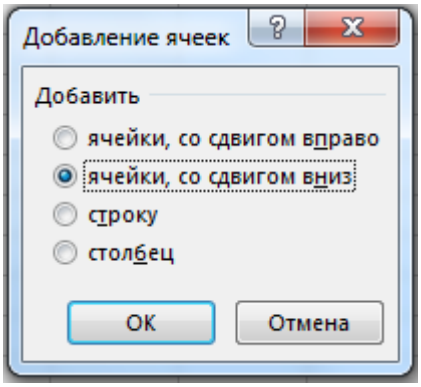
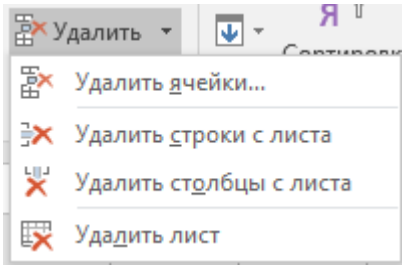


Рис. 1.18 - Вікно « Додавання клітинок »

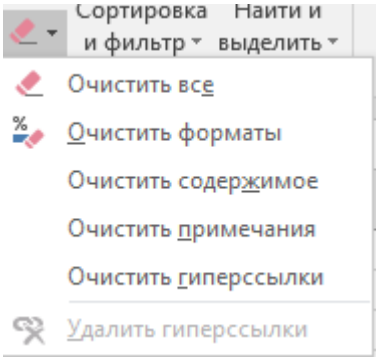
У вікні задається напрямок зсуву заповнених клітинок таблиці (праворуч або вниз). Число вставлених клітинок визначається числом клітинок у заздалегідь виділеному діапазоні. Якщо в таблицю треба додати рядки або стовпці, слід вибрати перемикач «строку» або «стовбець» (див. рис. 1.18).



Вилучення стовпця (рядка) електронної таблиці відбувається аналогічно додаванню за допомогою команд з підменю команди **Головна – Видалити**. Задане число стовпчиків (рядків) вилучається, якщо заздалегідь виділити відповідне число стовпців таблиці.

Вилучення стовпців (рядків) супроводжується зсувом заповнених стовпців (рядків) електронної таблиці ліворуч (вверх).

Вставлення (вилучення) комірок, рядків та стовпців супроводжується автоматичною настройкою комірок з формулами за місцем їх нового розміщення.



Якщо потрібно вилучити тільки вміст комірки, користуються клавішею **Delete**. При цьому сама комірка, а також параметри її форматування і примітка не вилучаються. Для вибіркового вилучення даних комірки користуються командою **Головна – Очистити** . Вона має підменю, із якого вибирають потрібний варіант очищення комірки: Усі, Формати, Зміст або Примітку. Перша команда повністю вилучає всі дані комірки: вміст, параметри форматування та примітку. Призначення інших команд очевидно.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Питання для самоконтролю

1. Призначення і основні можливості Excel?
2. Назвіть основні елементи вікна Excel.
3. Назвіть розмір стандартної таблиці Excel та адресацію комірок.
4. Як здійснити виділення фрагментів електронної таблиці (всієї таблиці)?
5. Як виконати вставку (видалення) комірок, стовпчиків, рядків?
6. Як виконати видалення інформації з комірок? Назвіть варіанти очищення комірок.
7. Якими засобами можна здійснити переміщення та копіювання вмісту комірок?
8. Як здійснити настроювання розмирів стовпчиків та рядків?
9. Як створити, зберегти нову книгу і відкрити існуючий файл?
10. Які формати введення дати часу та десятинного числа?
11. Як включити інструмент Автозаполнение ячеек та як їм користуватися?
12. Як швидко заповнити суміжні комірки повторюваними значеннями?
13. Де розташований та як виглядає маркер заповнення, його призначення та використання?
14. Як здійснюється створення складних послідовностей (прогресій)?
15. Як створити нові Списки і скористатися вже існуючими?
16. Які способі створення прогресій Вам відомі?
17. Що таке форматування робочого аркуша? Для чого воно потрібно?
18. Якими способами викликається вікно Формат клітинок…?
19. Які параметри форматування Ви знаєте?
20. Які дії можна виконувати над форматами?
21. У яких форматах можливо представити числові дані, календарну дату або час?
22. Як захистити комірки та робочий лист, як дозволити внесення змін в визначений діапазон комірок робочого аркуша, що захищається?

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

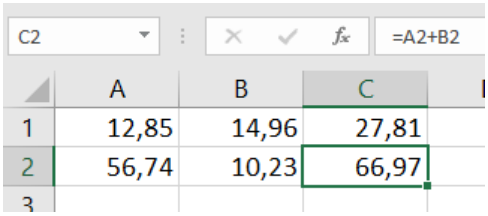
Microsoft Excel 2016.
Використання MS Excel для графічного представлення табличних даних.

Зміст

Microsoft Excel 2016. Використання MS Excel для графічного представлення табличних даних. 27
1. *Обчислення в Microsoft Excel. Робота з формулами та функціями* 27
Посилання на клітинки. Відносна та абсолютна адресація. 28
Стандартні функції. Використання майстра функцій. 29
2. *Наочне представлення табличних даних за допомогою діаграм* 31
Діаграма і її основні елементи 32
Побудова діаграм 33
Створення спарклайнів в Excel 2016 38
3. *Автоматизація завдань за допомогою макросів* 40
Запис макросу за допомогою макрорекодеру. 41
Робота з макросами, записаними в Excel 42
Питання для самоконтролю 43

1. *Обчислення в Microsoft Excel. Робота з формулами та функціями*

Обчислення в Excel виконують за допомогою **формул**. Формула може містити числові константи, посилання на клітинки і функції Excel, що сполучені знаками математичних операцій. Дужки дозволяють змінювати звичайний порядок виконання дій. Якщо клітинка містить формулу, то зазвичай в клітинці на робочому аркуші відображається тільки поточний результат обчислення цієї формули. Щоб побачити саму формулу, а не результат її роботи, потрібно виділити клітинку і подивитися на запис, який відображається в рядку формул, або двічі клацнути по клітинці.



Основне правило виконання обчислень в Excel полягає в тому, що якщо значення клітинки залежить від значень в інших клітинках таблиці, завжди потрібно використовувати формулу, навіть якщо розрахунок легко виконати "в голові"! Це гарантує те, що подальше редагування таблиці не порушить її цілісності і правильності, виконаних в ній обчислень

> Потрібно взяти за правило ніколи не виконуйте в "голові" навіть найпростіші обчислення, а завжди використовуйте для цього формули.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Посилання на клітинки. Відносна та абсолютна адресація.

У формулі на рис. 2.1 записана операція суми над вмістом двох клітинок. Замість чисел у формулі використані адреси клітинок - **посилання на клітинки**, вміст яких використовується в обчисленнях.

Це означає, що результат обчислення залежить від того, які числа знаходяться в клітинках. Клітинка, яка містить формулу, називається **залежною**. Формула в залежній клітинці перераховується кожного разу, коли змінюються значення в клітинках, адреси яких вказані у формулі. Посилання на клітинки можна задати:

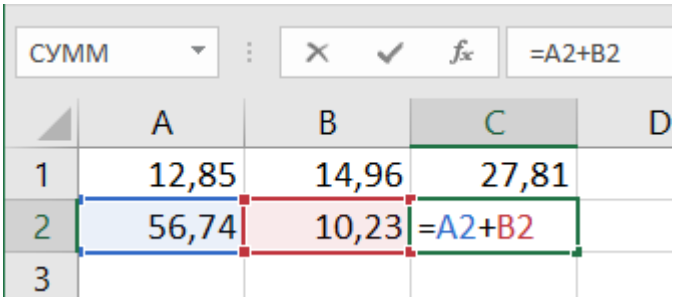


Рис. 2.1 Формульні обчислення

- адреси клітинок ввести вручну;
- клацнути на потрібній клітинці або обраному діапазоні, адресу якого потрібно ввести. Клітинка або діапазон при цьому виділяються пунктирною рамкою.

Для редагування формули потрібно двічі клацнути на відповідній клітинці. При цьому клітинки (діапазони), від яких залежить значення формули, виділяються на робочому аркуші кольоровими рамками, а самі посилання відображаються у формулі тим-самим кольором (рис. 2.1). Це полегшує редагування і перевірку правильності формул.

Відносна адресація. Від методу адресації посилань залежить, що буде з ними відбуватися при копіюванні формули з однієї клітинки в іншу. За замовчанням посилання на клітинки в формулах розглядаються як **відносні**. Це означає, що адреси посилань при копіюванні формули з однієї клітинки в іншу автоматично змінюються. Наприклад, якщо формулу з клітинки **C2** (**=A2+B2**) скопіювати в клітинку **C3** вона прийме вид **=A3+B3**, а при копіюванні формули в клітинку **D2** вона прийме вид **=B2+C2**. Таким чином відносні посилання приводяться у відповідність з відносним розташуванням початкової клітинки і створюваної копії. При копіюванні формула буде скопійована в іншу клітинку, та відносне розташування збережеться.

Абсолютна адресація. При абсолютній адресації адреси посилань при копіюванні формули не змінюються, так що адреса клітинки, на яку вказує посилання, розглядається як постійна. Для зміни способу адресації при редагуванні формули треба виділити посилання на клітинку і натиснути клавішу **F4**. Елементи адреси клітинки з абсолютною адресацію позначаються символом **$**. Наприклад, при послідовних натиснення клавіші **F4** адресу клітинки **A1** буде записуватися як **A1**, **A$1** і **$A1**. У двох останніх випадках один з компонентів адреси клітинки розглядається як абсолютний, а інший - як відносний.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Стандартні функції. Використання майстра функцій.

Введення формули у рядок формул завжди починається зі знаку **дорівнює**. Для простих обчислень використовуються числа і знаки математичних операцій. Порядок виконання дій регулюється дужками. Однак для складних обчислень виникає необхідність в використанні Стандартних функцій Excel.

Стандартна функція визначає значення деякої величини за стандартним алгоритмом, наприклад, стандартний алгоритм визначення середнього значення передбачає спочатку підсумовування декількох величин і подальший поділ отриманої суми на кількість даних. Стандартні функції використовуються в Excel тільки в формулах. Всього в Excel більше 400 стандартних функцій, велика частина з яких призначена для інженерних і наукових обчислень. За призначенням функції розбиті на 13 категорій, наприклад, окрему категорію складають фінансові функції, призначені для виконання розрахунків, пов'язаних з грошима, або статистичні, призначені для аналізу стохастичних даних.

Виклик функції полягає у вказівці у формулі імені функції, за яким в дужках слідує список параметрів. Параметри розділяються в списку крапкою з комою. Як параметр може використовуватися число, адреса клітинки або довільний вираз, для обчислення якого також можуть використовуватися функції.

Використання майстра функцій. Найпростіший спосіб вставки функції - почати введення клацанням на кнопці f_x в рядку формул або виконати команду **Формули – Вставити функцію**. Відкриється діалог Вставлення функції (рис. 2.2), який полегшує вибір і вставку в формулу потрібної функції. У списку Категорія вибирається категорія, до якої належить функція (Математичні (в даному випадку), Фінансові, Статистичні тощо). У списку Виберіть функцію вказується конкретна функція (у даному випадку функція **ABS**).

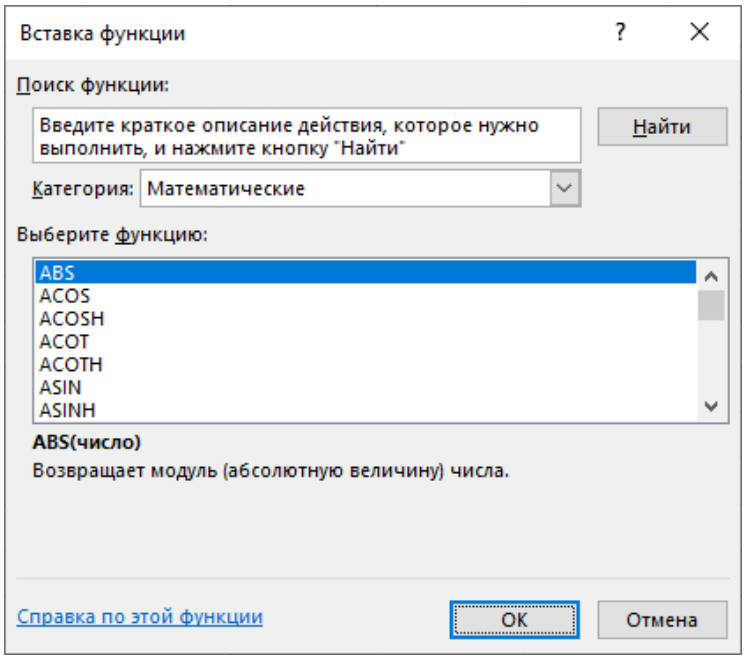


Рис. 2.2 Відкриється діалог Вставлення функції

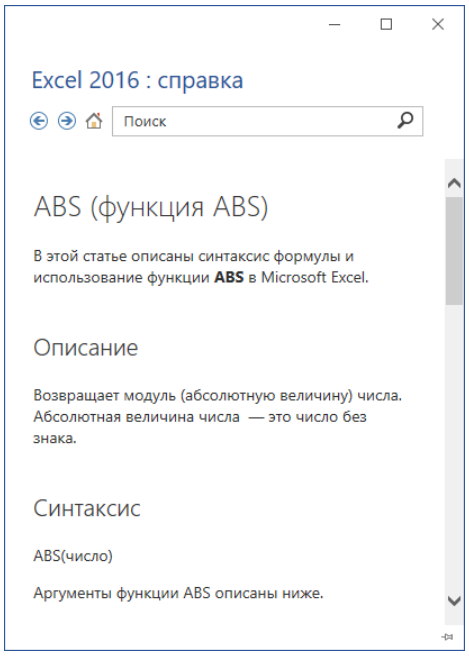


Рис. 2.3 Вікно довідки

Перелік лекцій

Назви функцій позначені послідовністю великих латинських або російських літер, наприклад, СРЗНАЧ або SIN, за якими можна не завжди зрозуміти їх призначення. Тому для пошуку потрібної функції необхідно ввести її опис у поле **Пошук функції**.

Тут же можна отримати довідку щодо обраної функції, яка відкривається у вікні Довідка (рис. 2.3).

Потрібну функцію можна також знайти і вставити в клітинку використовуючи Бібліотеку функцій вкладки Формули де функції представлені за категоріями (рис. 2.4).

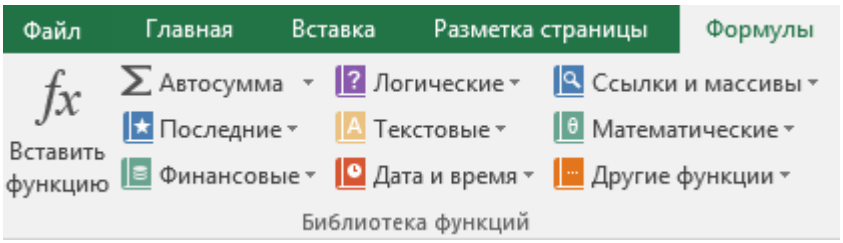


Рис. 2.4 Бібліотека функцій

Клацання на кнопці **ОК** відкриває вікно Аргументи функції (рис. 2.5), одночасно ім'я функції заноситься в рядок формул разом з дужками, які обмежують список аргументів.

Якщо назва параметра відзначено жирним шрифтом, то цей параметр є обов'язковим, і відповідне поле не може залишатися порожнім. Параметри, назви яких виводяться звичайним шрифтом, можна опускати. У середній частині вікна наводиться короткий опис функції, а також призначення і опис параметра.

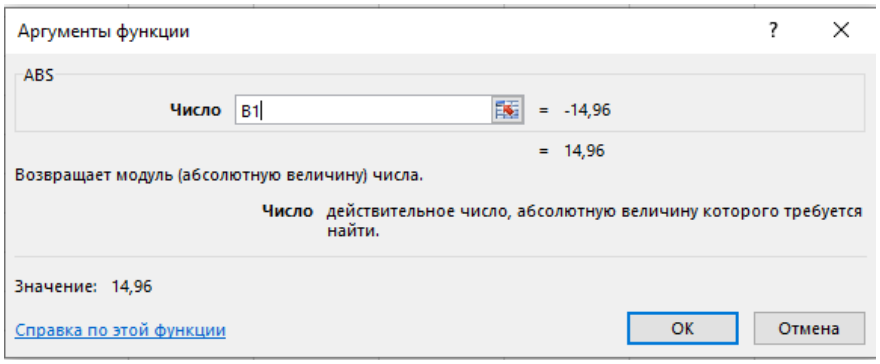


Рис. 2.5 Вікно Аргументи функції

Значення параметрів можна вводити безпосередньо в рядок формул або в поля параметрів у вікні діалогу. Якщо параметри функції знаходяться в певних клітинках, то посилання задають клацанням по відповідним клітинкам на робочому аркуші. Якщо параметр заданий, то поруч з полем виводиться його числове значення. У вікні діалогу можна також побачити значення функції, обчислене при заданих параметрах.

Правила обчислення формул, які містять функції, не відрізняються від правил обчислення більш простих формул. Посилання на клітинки, які використовуються в якості параметрів функції, можуть бути відносними або абсолютними, що необхідно враховувати при копіюванні формул методом автозаповнення.

Підсумкові обчислення. Підсумкові обчислення забезпечують одержання числових характеристик, які описують визначений набір даних у цілому. Наприклад, є можливість обчислення суми значень, які входять у набір, їх середнього значення й інших статистичних характеристик. Загальною властивістю цих функцій є те, що вони

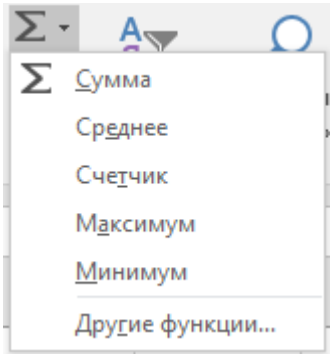
©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

можуть містити змінну кількість параметрів (або діапазон невизначеного розміру). Такі операції виконують за допомогою стандартних функцій.

Особливість використання підсумкових функцій полягає в тому, що при їх заданні програма намагається визначити, у яких клітинках знаходиться оброблюваний набір, а також автоматично задати параметри функції. Як правило, параметром підсумкової функції є діапазон клітинок.

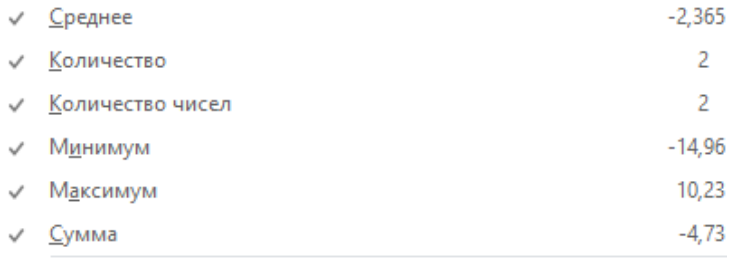
Найбільш типовою для підсумкових обчислень є функція підсумовування (**СУММ**). Вона обчислює сумарне значення чисел одного стовпця, розташованого над поточною клітинкою, або рядка, розташованого зліва від поточної клітинки. Щоб обчислити суму, рамку виділення поміщають під стовпчиком чисел або справа від рядка. Потім роблять клацання на кнопці **Сума** і натискають на клавішу **Enter**.

Автоматичне визначення діапазону не виключає можливості редагування формул з підсумковими функціями. Можна уточнити діапазон, що був обраний автоматично, а також задати додаткові параметри. Наприклад, визначити значення суми прямокутного блока клітинок або для несуміжних клітинок.



За допомогою списку, що розкривається, кнопки **Сума** можна вибрати і інші підсумкові функції. До них відносять функцію СРЗНАЧ - середнє значення, СЧЕТ - кількість клітинок із числами в діапазоні, функції МАКС - максимальне та МИН – мінімальне число будь-якого діапазону. Інші функції для підсумкових обчислень вставляються за допомогою **Майстра функцій**, вони відносяться до категорії **Статистичні**.

Є можливість формування підсумкових значень також поза робочим листком, що використовуються для аналізу й оцінки виділеного діапазону даних робочого листка. В цьому випадку вибране користувачем підсумкове значення виділених клітинок робочого листа відображається в рядку стану. Значення підсумкових функцій (середнє, кількість значень і чисел, максимум тощо), які будуть виводитися в рядку стану, можна налаштовувати, для цього потрібно натиснути правою кнопкою по цьому рядку і в контекстному меню встановити прапорці для необхідних функцій



> Необхідно мати на увазі, що отримані у такий спосіб значення підсумкових функцій відображаються тільки в рядку стану, а не в клітинках робочого аркуша. Цей спосіб зручний для проміжних орієнтовних обчислень.

2. *Наочне представлення табличних даних за допомогою діаграм*

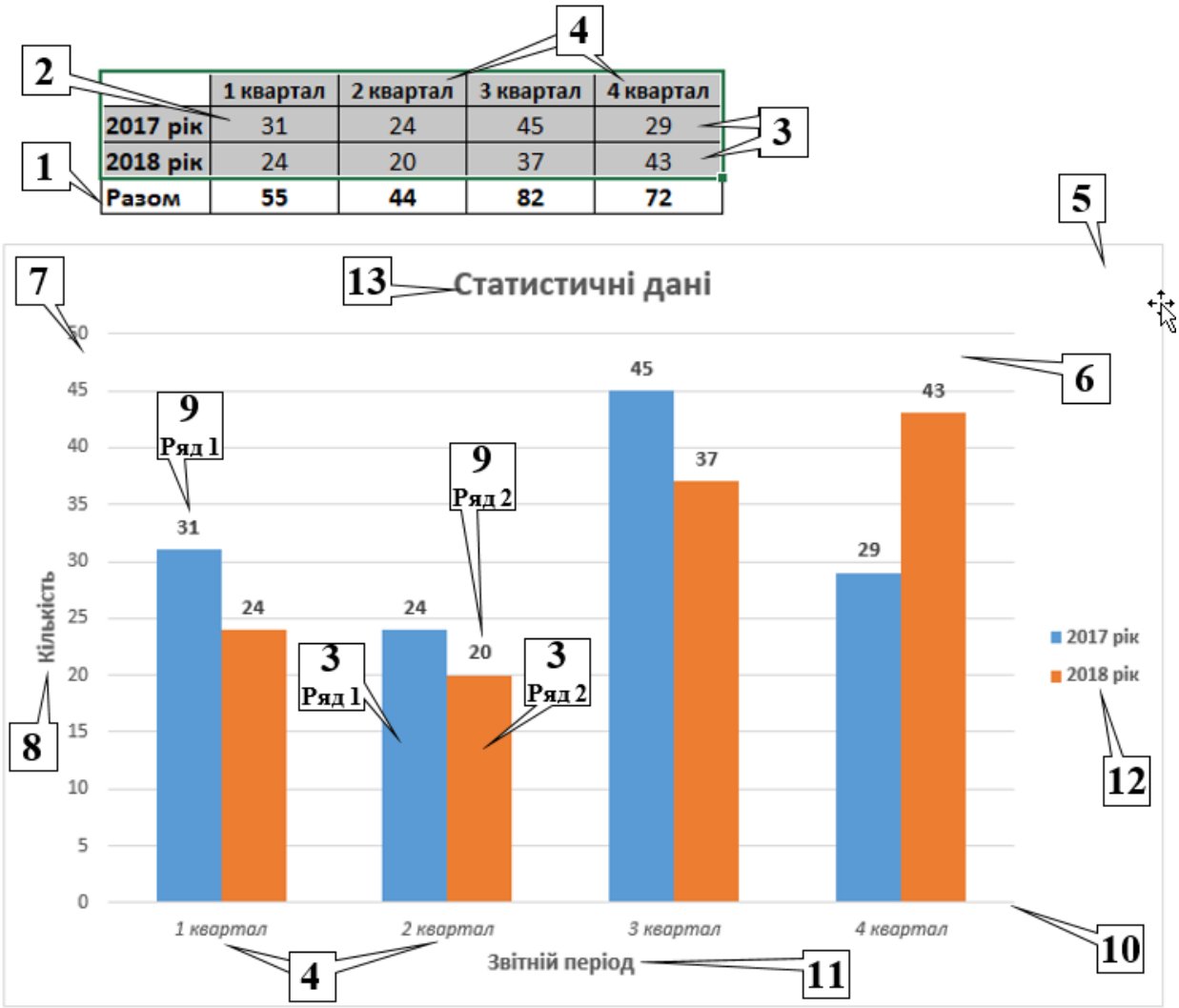
В Excel термін "діаграма" використовується для позначення всіх видів графічного відображення числових даних. Діаграми дозволяють представляти і аналізувати дані таблиць в наочній формі. Побудова діаграми здійснюється на основі

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

рядів даних (група клітинок з даними в межах рядка або стовпця). На діаграмі можна відобразити кілька рядів даних.

Діаграма і її основні елементи

Діаграма — графічне представлення числових даних лінійними відрізками, кривими або геометричними фігурами, що дозволяє швидко оцінити співвідношення кількох величин. Діаграма створюється на основі даних, які містяться в таблиці та зберігає зв'язок з нею. При зміні даних в таблиці діаграма автоматично змінюється. Вона може розташовуватися на тому- самому аркуші, що і таблиця, або на окремому аркуші.



	1 квартал	2 квартал	3 квартал	4 квартал
2017 рік	31	24	45	29
2018 рік	24	20	37	43
Разом	55	44	82	72

Рис. 2.6 Діаграма в MS Excel 2016

Містить наступні елементи: вихідна таблиця (1); виділений діапазон для побудови (2); ряди даних (3); категорії (4); область побудови (5); область діаграми (6); ось значень (7); назва осі значень (8); підписи даних (9); ось категорій (10); назва осі категорій (11); легенда (12); назва діаграми (13);

Основні елементи діаграми наведені на рис. 2.6. На рисунку зверху наведено вихідну таблицю (1), а знизу – діаграму, що побудована на підставі виділеного діапазону клітинок (2) з цієї таблиці. Окрім цього на діаграмі присутні наступні елементи:

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

- **Область побудови (5)** – область, в якій розміщуються всі елементи діаграми.
- **Область діаграми (6)** – область діаграми, обмежена осями, яка містить ряди даних.
- **Ряди даних (3)** – набір пов'язаних між собою числових даних, що відображаються по вертикальній осі діаграми у вигляді стовпців, секторів тощо. У наведеній вихідній таблиці дані розташовані по рядках.
- **Категорії (4)** – зазвичай назви даних, що відображаються під горизонтальною віссю діаграми. У наведеній вихідній таблиці ці назви розташовані по стовпцях.
- **Осі** – лінії, що обмежують область діаграми і мають поділки зі значеннями обраних одиниць виміру. Горизонтальна ось (10) (ось X) відповідає категоріям. Вертикальна ось (7) (ось У) відповідає значенням даних в категоріях. У тривимірних діаграм є третя ось (ось Z), яка зазвичай, відповідає часу.
- **Легенда (12)** – графічні зображення та підписи, які відповідають категоріям і полегшують читання діаграми.
- **Назви (8, 11, 13)** – текст, який відображає назву діаграми або осей.
- **Підписи даних (9)** – значення рядів даних в певних категоріях, полегшують читання діаграми.

Також на діаграмі може бути присутня **лінія тренду** – графік функції певного виду (лінійна, логарифмічна, статечна тощо) отриманий в результаті обробки даних ряду методом найменших квадратів, що дозволяє наочно уявити тенденцію зміни даних.

Побудова діаграм

Побудова діаграми здійснюється на основі рядів даних - груп клітинок з даними в межах одного рядка або стовпця таблиці. Ряд даних складається з категорій - окремих його значень. Діаграма зберігає зв'язок з даними, на основі яких вона була побудована. Відновлення даних на робочому листку приводить до зміни вигляду діаграми.

Основою для побудови будь-якої діаграми є дані з таблиці, яка повинна бути створена заздалегідь. Ця таблиця має відповідати певним вимогам, які зроблять подальше створення діаграми більш ефективним:

1. Перш за все, необхідно прийняти рішення, як будуть розташовуватися дані. Зазвичай ряди даних розміщують по рядках, а категорії - по стовпцях, наприклад, так, як в таблиці на рис. 2.6. Однак дані можуть бути розміщені і навпаки.
2. Слід максимально обмежити кількість рядів даних і категорій їх зміни. Велика кількість даних, що виводяться на діаграму, ускладнять її сприйняття. Досвід показує, що добре сприймаються діаграми, які складаються не більш ніж з п'яти рядів даних, які змінюються не більше ніж за п'ятьма категоріями.
3. Якщо вихідна таблиця складається з великої кількості стовпців і рядків, тільки частина з яких буде використана для створення діаграми, то саме з цих рядків і стовпців доцільно створити нову таблицю або приховати рядки і стовпці вихідної таблиці, які не будуть відображатися на діаграмі.
4. Заголовки рядів даних (категорії) повинні бути максимально короткими, для великих заголовків на діаграмах просто немає місця.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

5. Створення діаграм, як правило, розпочинається з виділення діапазону клітинок таблиці, дані з яких будуть використовуватися для побудови діаграми. Діапазон може включати тільки числові дані або числові дані разом із заголовками рядів і категорій. Якщо назви рядів та категорій входять до виділеного діапазону, то вони використовуються при створенні діаграми.

Вибір типу діаграми Щоб почати побудову діаграми необхідно на вкладці **Вставлення** в групі **Діаграми** натиснути кнопку трикутник праворуч від мініатюри будь-якої діаграми (рис. 2.7). Відкриється колекція різновидів обраного типу діаграми, з якої необхідно вибрати потрібну. Наприклад, на рис. 2.7 праворуч наведена колекція для вибору гістограми. Але, перш за все, необхідно вибрати тип діаграми. Вибір типу діаграми є відповідальним кроком, від якого залежить наочність представлення даних.

Рис. 2.7 Колекція діаграм

Іноді користувач смутно уявляє собі, який з типів діаграм найліпше підійде для представлення певних даних. У такому випадку доцільно скористатися допомогою експертної системи Excel. При виділені діапазону даних в таблиці система аналізує їх та пропонує використати певні типи діаграм для їх представлення. Пропозиції можна переглянути натиснувши кнопку **Рекомендовані діаграми** (рис. 2.7). Відкриється діалог де буде представлено декілька типів рекомендованих діаграм, серед яких необхідно здійснити вибір. Для того щоб здійснити усвідомлений вибір, доцільно ознайомитися з призначенням різних типів діаграм і умовами їх застосування.

Міркування щодо вибору типу діаграми. Всі доступні типи діаграм наводяться в діалозі Вставка діаграми (рис. 2.8), який відкривається після клацання по трикутнику у нижній правій частині розділу Діаграми (рис. 2.7), У вікні на панелі ліворуч відображаються доступні типи діаграм, а ліворуч відображається колекція мініатюр діаграм виділеного типу. Можливий вибір наступних основних типів діаграм:

Гістограма - цифрові дані відображаються у вигляді вертикальних зображень: прямокутників або стовпчиків. Використовується для порівняння значень різних категорій, коли порядок категорій неважливий;

Стовпчаста - аналогічна Гістограмі, але зображення розташовуються горизонтально, використовується для порівняння значень різних категорій коли діаграма відображає тривалість, або коли текст категорій довгий;

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Графік - дані відображаються у вигляді окремих точок, які об'єднуються лініями різних типів. Використовується для відображення тенденцій протягом певного часу (день, тиждень тощо), або коли порядок категорій чи точок даних є важливим;

Секторна - дані подаються у вигляді секторів кола. Використовується для відображення пропорцій цілого;

З областями - аналогічна **Графіку**, але області розташовуються під лініями і виділяються різними кольорами;

Точкова - на діаграмі відображаються окремі точки з позначенням координат X, У. Використовується для відображення зв'язків між наборами значень;

Поверхня - подібна **Графіку**, але дані відображаються у вигляді тривимірної поверхні;

Біржова - відображає мінімальні і максимальні ціни, а також ціни на момент закриття торгів;

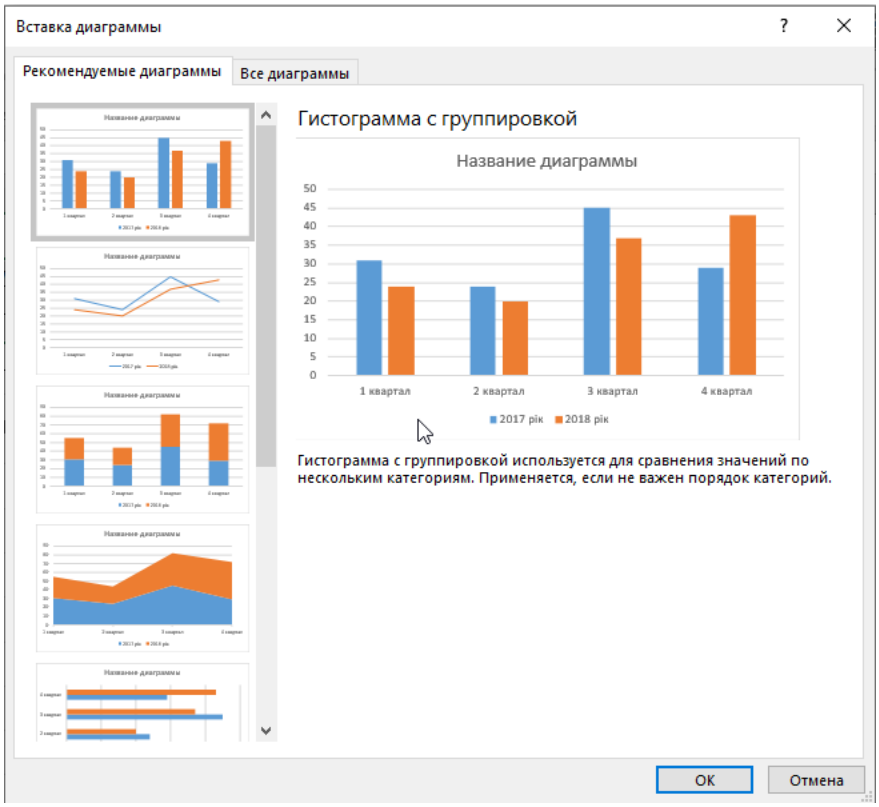


Рис. 2.8 Діалог вставка діаграми

Сонячне проміння - нагадує **Секторну діаграму**, але в центрі вона має отвір;

Пелюсткова – дані відображаються відносно центральної точки, а не щодо осей X, У.

Комбінована – ряди даних представляються комбінацією різних типів діаграм.

При виборі типу діаграми наочність представлення даних має переважувати всі інші її характеристики. Наприклад, якщо необхідно проілюструвати співвідношення окремих даних, або тенденцію розвитку, то краще вибрати Стовпчасту. Якщо необхідно проілюструвати співвідношення окремих показників, то ліпше обрати кругову діаграму.

Після того, як тип діаграми буде обрано, необхідно клацнути на відповідній мініатюрі і діаграма, що побудована відповідно до виділеного діапазоном даних в таблиці, буде автоматично вставлена на робочий аркуш. Не варто засмучуватися, якщо отриманий результат не повністю виправдає очікування. Excel спочатку створює макет-заготовку діаграми, який передбачає внесення змін і її форматування у відповідності до потреб користувача.

Перелік лекцій

Форматування діаграми проводиться командами, розташованими на контекстних вкладках **Конструктор** і **Формат** (рис. 2.9), які автоматично стають доступними відразу з виділенням діаграми.

Іноді трапляється, так, що початково обраний тип діаграми недостатньо наочно відображає дані, тому виникає задача зміни типу діаграми. Це можна зробити кількома способами. Перший спосіб - натиснути правою кнопкою в межах діаграми і з контекстного меню вибрати команду **Змінити тип діаграми**. Другий спосіб - виділити діаграму і на вкладці Конструктор у розділі Тип клацнути по кнопці **Змінити тип діаграми** (рис. 2.9).

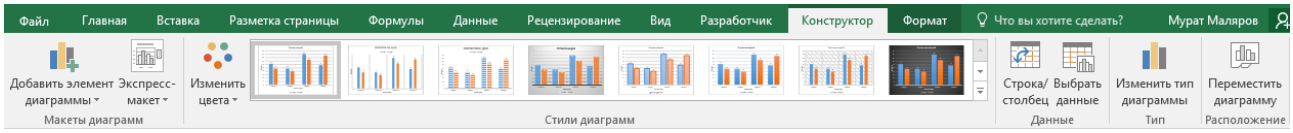


Рис. 2.9 Контекстна вкладка Конструктор (робота з діаграмами)

У будь-якому випадку відкриється діалог **Змінення типу діаграми**, в якому потрібно вибрати іншу діаграму. Діаграма на аркуші відразу зміниться відповідно до обраного нового типу.

Якщо при побудові діаграми були виділені тільки чистові дані без включення назв рядків і (або) стовпців, то в цьому випадку за замовчуванням кожному ряду присвоюється системне ім'я: **Ряд1**, **Ряд2** тощо.

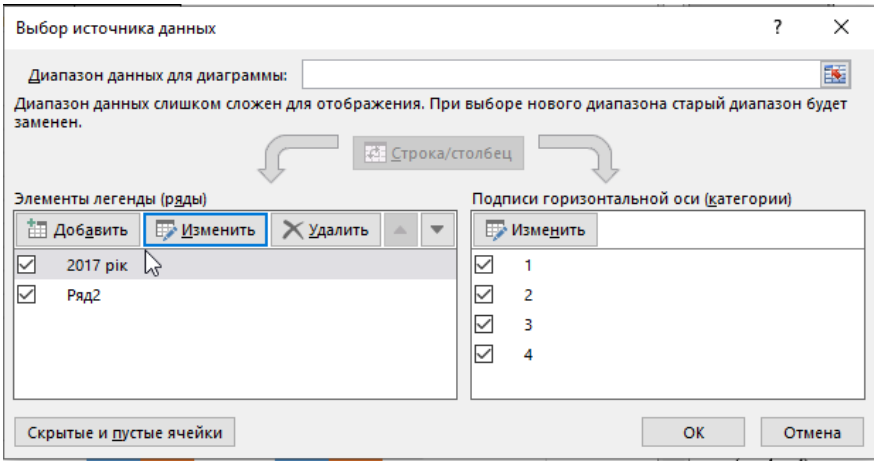
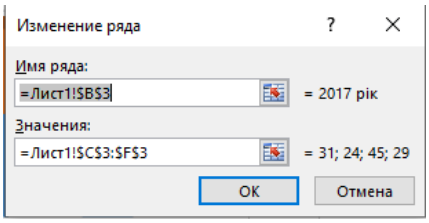


Рис. 2.10 Діалогове вікно вибору початкових даних

Для зміни імені ряду потрібно скористатися кнопкою **Вибрати дані**, яка розташована у групі **Дані** контекстної вкладки **Конструктор**. Ця кнопка відкриває діалог вибір початкових даних (рис. 2.10). Системні імена: **Ряд1**, **Ряд2** відображаються в списку **Елементи легенди (ряди)**. Після клацання на потрібному імені за допомогою кнопки **Змінити** у поле **Ім'я ряду** вводять назву ряду. Введення роблять з клавіатури або шляхом посилання на комірку робочого листа, яка містить ім'я даного ряду. В останньому випадку розміри вікна можна зменшити до розмірів поля за допомогою кнопки мінімізації . Кнопки **Добавити** та **Видалити** використовують для додавання (вилучання) рядів даних, за якими будується діаграма. Діючи за аналогією у цьому вікні задають назви категорій, котрі можна переглянути у списку **Підписи горизонтальної осі (категорії)**. Назви категорій

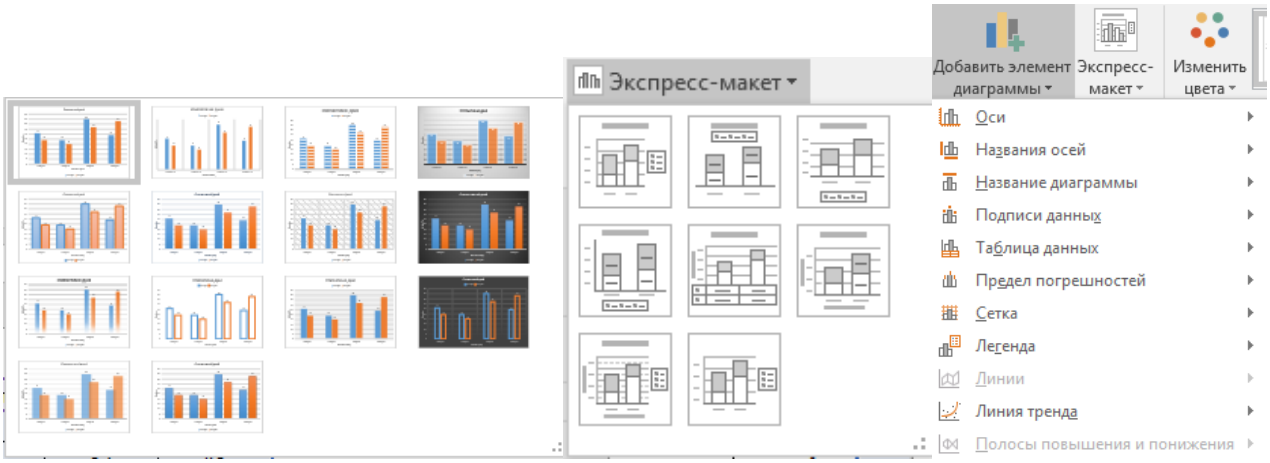


©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

вводять шляхом виділення клітинок таблиці з відповідними назвами, після клацання на кнопці **Змінити**.

При створенні діаграми Microsoft Office Excel визначає вісь, на якій виводяться ряди даних, в залежності від кількості рядків і стовпців аркуша, які містяться в діаграмі, розміщуючи більше число на горизонтальній осі – осі категорій. Однак якщо дані містять однакові рядки і стовпці даних листа, Excel відображає рядки даних на вертикальній осі, а стовпці даних - на горизонтальній осі. Для зміни представлення даних на діаграмі використовується кнопка **Рядок/Стовбець**, яка дозволяє швидко змінити спосіб відображення рядків і стовпців на діаграмі, переключивши рядки в стовпці або навпаки.

Для того, щоб прискорити роботу і не займатися форматуванням окремих елементів, можна скористатися колекцією оформлення з групи **Стилі діаграм** (рис. 2.11 мініатюра **а**)



а) **б)** **в)**

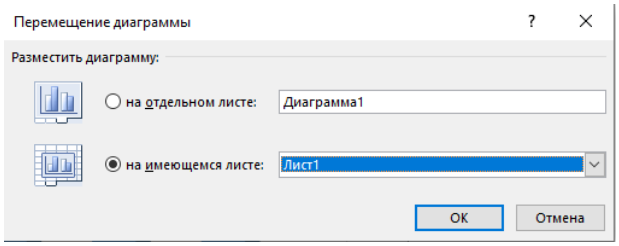
Рис. 2.11 Редагування елементів діаграми

Крім типу діаграми, істотним і важливим є наявність певних елементів діаграми і їх розміщення, які визначаються макетом діаграми. Тому перш, ніж приступити до форматування елементів діаграми, доцільно вибрати її макет. Вибір макета діаграми здійснюється за допомогою колекції (рис. 2.11 мініатюра **б**), який відкривається за допомогою кнопки **Експрес- макет**, що розташована на вкладці **Конструктор** у групі **Макети діаграм**.

Мініатюри макетів відповідають обраному типу діаграми і показують, які елементи і де будуть представлені на діаграмі. Діаграма на аркуші інтерактивно змінюється при переміщенні покажчика миші по мініатюрах макетів. Остаточно макет вибирається клацанням миші по мініатюрі. Але і обраний макет завжди можна змінити додаванням, видаленням або переміщенням його елементів.

Додати елементи на макет діаграми можна за допомогою кнопки **Додати елемент діаграми**, яка відкриває панель з переліком елементів (рис. 2.11 мініатюра **в**). Праворуч від назви елемента знаходиться трикутник, який відкриває колекцію з мініатюрами що уточнюють положення конкретного елемента на діаграмі.

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua



За замовчуванням діаграма створюється на тому-самому аркуші, що і таблиця, але буває зручніше (особливо якщо діаграма велика і містить багато елементів) помістити її на окремому аркуші. Найпростіше це зробити, вибравши команду **Перемістити діаграму** з контекстного меню або натиснути однойменну кнопку на вкладці **Конструктор** у групі **Розташування** (рис. 2.9). Відкриється діалог (рис. 2.85), в якому можна вибрати перемикач на окремому аркуші - в книгу буде додано новий аркуш з діаграмою, або вказати на наявному аркуші - діаграма буде переміщена на будь-який вже наявний аркуш, ім'я якого можна вибрати зі списку.

Створення спарклайнів в Excel 2016

С приходом Excel 2010 з'явилася нова можливість створювати міні-діаграми-графіки, так звані спарклайни. Спарклайн - це невелика діаграма, вміщена в одну клітинку. Вона дозволяє швидко відстежувати дані, що динамічно змінюються. Оскільки спарклайни вміщаються в одну клітинку, їх прийнято використовувати групами. У деяких джерела спарклайни називають інфолініями.

Хоча спарклайни багато в чому схожі з звичайними діаграмами, їх функціональність повністю відокремлена від функціональності діаграм в Excel. Наприклад, розміщення діаграми вимагає окремого шару і на діаграмі можна побудувати кілька рядів даних. Спарклайн відображається всередині осередку і показує тільки один ряд даних.

В Excel існує три типи спарклайнів: спарклайн-графік, спарклайн-гістограма і спарклайн виграшу / програшу.

Графік - схожий з діаграмою у вигляді графіка. Для даного типу спарклайну можливий варіант відображення маркера для кожної точки даних.

Гістограма – виглядає як і звичайна гістограма. Підписи даних у цьому випадку встановити не можна.

Виграш / програш - ділить всі дані на два типи: негативні і позитивні і відображає їх у вигляді верхнього і нижнього блоку.

Як правило, для кожного ряду даних будується один спарклайн, але при бажанні Ви можете створити будь-яку кількість спарклайнів і розмістити їх там, де необхідно. Перший спарклайн найпростіше створити в самому верхньому рядку з даними, а потім, використовуючи маркер автозаповнення, скопіювати його в усі інші ряди.

Щоб створити спарклайн, виберіть ряд даних, який потрібно візуалізувати. Потім на вкладці **Вставка** в групі **Спарклайни** потрібно вибрати один з трьох типів спарклайну: гістограма, графік або виграш / програш.

На екрані відобразиться діалогове вікно Створення спарклайну (рис. 2.12), де необхідно буде вказати діапазон даних і діапазон розташування самого спарклайну.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Як прищепило, графіки розташовуються поруч з даними для кращого сприйняття, але це не обов'язкова вимога.

Excel дозволяє вставляти спарклайн в клітинку, яка вже містять дані. Розташування спарклайнів при побудові має відповідати джерелу даних за кількістю рядків або кількості стовпців.

Excel надає достатній набір інструментів для зміни зовнішнього вигляду спарклайнв. Користувач може налаштувати відображення маркерів, задати колір, змінити тип і стиль спарклайну тощо. Для цього необхідно перейти на контекстну вкладку **Конструктор** (рис. 2.13), що з'являється, коли ви вибираєте клітинку з спарклайном.

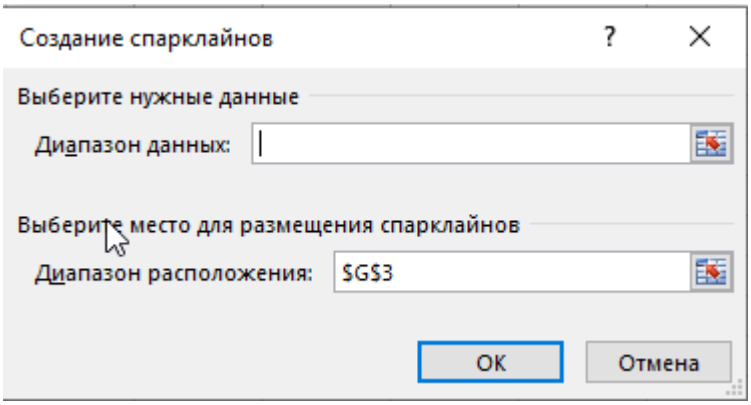


Рис. 2.12 Діалогове вікно Створення спарклайну

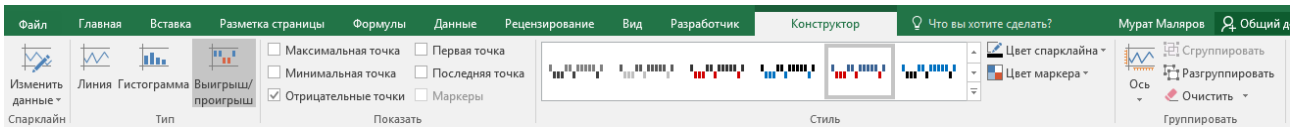


Рис. 2.13 Контекстна вкладка Конструктор (робота зі спарклайними)

За допомогою вкладки конструктор можливо виконання наступних операцій:

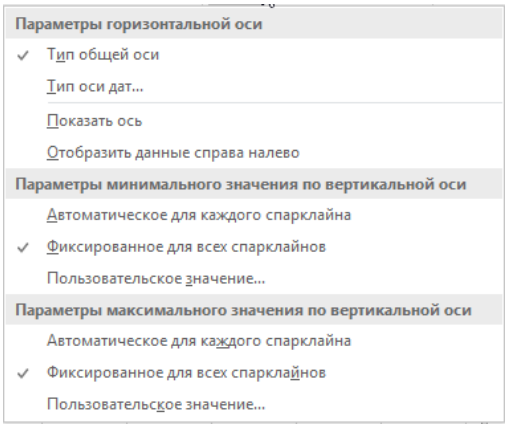
Відображення маркерів. Можливо акцентувати увагу на певних ділянках спарклайну за допомогою маркерів або точок, тим самим збільшуючи його інформативність. Наприклад, на спарклайнах з безліччю великих і малих значень дуже важко зрозуміти, яке з них є максимальним, а яке мінімальним. З увімкненими опціями **Максимальна точка** і **Мінімальна точка** зробити це набагато простіше.

Зміна стилю. При побудові спарклайну користувачу пропонується велика кількість стилів оформлення, вибрати які можливо у групі **Стиль**, при натисканні на стрілку меню, що випадає.

Зміна типу. Якщо зовнішній вигляд спарклайну на інформативний, то використовуючи групу Тип, його можливо змінити на бажаний. Кожен тип спарклайну призначений для певних цілей. Наприклад, спарклайн виграшу / програшу більше підходить для даних, де є позитивні або негативні значення.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua



Зміна діапазону відображення. За замовчуванням кожен спарклійн в Excel має такий масштаб, який відповідає максимальному і мінімальному значенню його вихідних даних. Максимальне значення розташовується вгорі клітинки, а мінімальне внизу. На жаль, це не демонструє величину значення, якщо порівнювати його з іншими спарклайнами. Excel дозволяє змінити зовнішній вигляд спарклайну, щоб їх можна було порівнювати один з одним. Для цього на вкладці **Конструктор** необхідно вибрати команду **Вісь**. З'явиться меню, що випадає. У цьому меню в параметрах для максимального і мінімального значень по вертикальній осі потрібно ввімкнути опцію Фіксована для всіх спарклайнів.

3. *Автоматизація завдань за допомогою макросів*

Для автоматизації дій, що часто повторюються в Microsoft Excel можна записати макрос. Наприклад припустимо, що на робочому листі введені дати в різних форматах, й потрібно застосувати до всіх з них один і той же формат. Це можна зробити за допомогою макросу. Користувач може записати макрос, який застосує потрібний формат дати, а потім запускати його при необхідності.

Макрос – це набір інструкцій, що дозволяють автоматизувати виконання часто повторюваних дій, які запрограмовані на певну послідовність та записані в середовищі програмування на мові Visual Basic for Applications (VBA).

При записи макросу записуються всі дії, описані в VBA коді. Ці дії можуть включати введення тексту або чисел, вибір клітинок або натискання кнопок команд на стрічці або в меню, форматування рядків або стовпців, а також імпорт даних із зовнішнього джерела. Додаток Visual Basic (VBA) - це підмножина потужної мови програмування Visual Basic, яке входить в більшість додатків Office. Незважаючи на те, що VBA забезпечує можливість автоматизації процесів між додатками Office, при записі макросів користувачу не потрібно знати код VBA або програмне програмування, якщо це не потрібно.

Важливо знати, що при записі макросу реєструються всі дії. Тому якщо при запису допущена помилка, наприклад натиснута не та кнопка, засіб запису макросів зареєструє цю дію. В такому випадку можна знову записати всю послідовність або змінити код VBA. Тому перед записом процесу слід добре потренуватися, чим точніше буде записана послідовність, тим ефективніше буде працювати макрос.

Макроси і засоби VBA знаходяться на вкладці **Розробник** (рис. 2.14), яка за замовчуванням прихована, тому спочатку потрібно додати її на стрічку команд, виконавши команду **Файл – Параметри**, та вибрати пункт **Налаштувати стрічку**.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

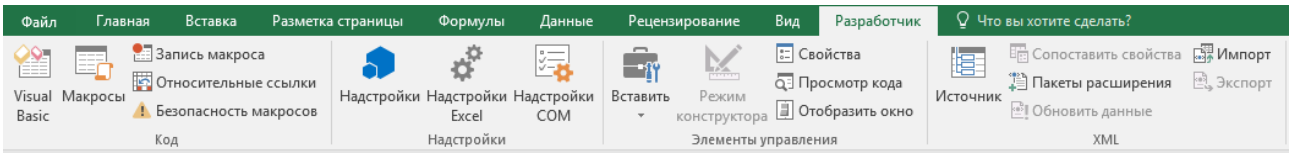


Рис. 2.14 Вкладка Розробник

Перед записом макросів корисно знати наступне:

Макрос, записаний для роботи з діапазоном Excel, буде виконуватися тільки для клітинок цього діапазону. Тому якщо користувач додає в діапазон новий рядок, макрос до нього застосовуватиметься не буде.

Якщо потрібно записати довгу послідовність завдань, то краще замість цього використовувати кілька дрібніших макросів.

У макросі можуть міститися і завдання, що не належать до Excel. Процес макросу може охоплювати інші додатки Office і інші програми, які підтримують Visual Basic для додатків (VBA). Наприклад, можна записати макрос, який спочатку оновлює таблицю в Excel, а потім відкриває Outlook для її відправки по електронній пошті.

Запис макросу за допомогою макрорекодеру.

Щоб записати макрос, потрібно перейти на вкладку **Розробник** та у групі **Код** натиснути на кнопку **Запис макросу**.

Відкриється діалогове вікно **Запис макросу** (рис. 2.15) де у **полі Ім'я макросу** потрібно ввести назву макросу. Бажано зробити ім'я зрозумілим, щоб можна було швидко знайти потрібний макрос.

Для швидкого запуску макросу можливо призначити поєднання клавіш, для цього у відповідне поле потрібно ввести будь-яку малу або велику літеру. Рекомендується використовувати поєднання клавіш з **Ctrl+Shift**, так як вони будуть заміняти собою стандартні сполучення клавіш в Excel, поки відкрита книга, що містить макрос.

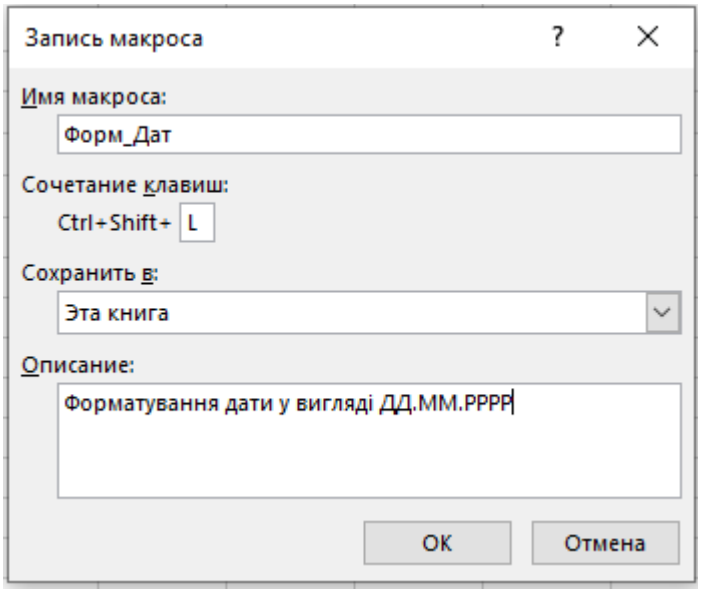


Рис. 2.15 Діалогове вікно Запис макросу

У списку **Зберегти** потрібно вибрати книгу, в якій потрібно зберегти макрос. Як правило, макрос зберігається в зазначеній книзі, але якщо потрібно, щоб макрос був доступний при використанні будь-яких файлів Excel, необхідно вибрати пункт **Особиста книга макросів**. Якщо вибрати команду **Особиста книга макросів**, Excel створить приховану особисту книгу макросів (з розширенням **.xlsb**), якщо вона ще не існує, і збереже макрос в цій книзі.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

У полі **Опис** при необхідності можна ввести короткий опис дій макросу. Хоча поле "Опис" є необов'язковим, рекомендується його заповнити. Крім того, бажано ввести зрозумілий опис, яке буде корисно всім, хто запускає макрос. Якщо у книзі багато макросів, описи допоможуть швидко визначити, для чого вони потрібні.

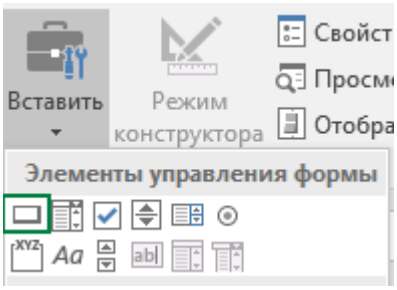
Щоб почати запис макросу, потрібно натисніть кнопку **ОК** та виконати дії, які потрібно записати. Для зупинки макросу потрібно натиснути кнопку **Зупинити запис**, (вкладка **Розробник**, група **Код**)

> Першим символом імені макросу повинна бути буква. Наступні символи можуть бути буквами, цифрами або знаками підкреслення. В імені макросу не повинно міститися пробілів; в якості роздільників слів слід використовувати знаки підкреслення. Якщо використовується ім'я макросу, що є посиланням на клітинку, може з'явитися повідомлення про помилку, яке вказує на неприпустиме ім'я макросу.

Робота з макросами, записаними в Excel

Щоб переглянути макроси, пов'язані з книгою, на вкладці **Розробник** потрібно натиснути кнопку **Макроси**. При цьому відкриється діалогове вікно Макрос , в якому потрібно вибрати назву необхідного макросу та натиснути кнопку **Виконати**.

> Макроси не можна скасувати. Перш ніж вперше запускати записаний макрос, необхідно зберегти книгу або створити її копію, щоб запобігти внесенню небажаних змін. Якщо результати виконання макросу виявилися небажаними, можна закрити книжку, не зберігаючи її.



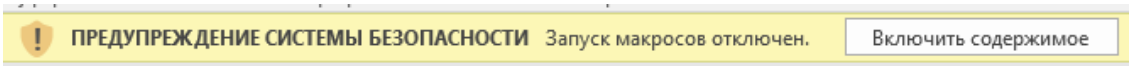
Також для запуску створеного макросу можливо призначити кнопку елемента управління форми. Для цього на вкладці **Розробник** (рис. 2.14) в групі **Елементи керування** натисніть кнопку **Вставити**, а потім в розділі **Елементи управління форми** натисніть значок кнопки.

Після всього цього необхідно намалювати кнопку на робочому аркуші при затиснутій лівій кнопці миші. Після закінчення процесу малювання включиться автоматично вікно, де необхідно буде вибрати той макрос, який повинен виконуватися при натисканні на цю кнопку.

Щоб змінити властивості елемента управління, необхідно клацнути на ньому правою кнопкою миші і вибрати пункт **Формат елемента керування**.

Нагадаємо, що макроси служать для автоматизації часто виконуваних завдань, що дозволяє заощадити час за рахунок скорочення обсягу роботи з клавіатурою і мишкою. Багато з них створені з використанням мови Visual Basic для додатків (VBA). Однак деякі макроси представляють загрозу безпеці. Зловмисник може впровадити в файл потенційно небезпечний макрос, здатний викликати зараження комп'ютера або мережі організації вірусом.

Тому при відкритті файлу, що містить макроси, з'являється жовта панель повідомлень із позначкою щита і кнопкою **Включити вміст**.



Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Якщо відомо, що макрос надійшов з надійного джерела, необхідно у області Панель повідомлень натиснути кнопку **Включити вміст**, при цьому файл відкриється як надійний документ.

Питання для самоконтролю

23. Яка послідовність уведення формули в комірку та її виконання?
24. Які існують зручні прийоми введення у формулу змінних?
25. Як можна відобразити формулу, а не результати розрахунків у комірці?
26. Яка послідовність команд включає/виключає режим автоматичного перерахунку формул?
27. Як можна копіювати формули в Excel?
28. Як задати абсолютну адресацію комірок на робочому аркуші?
29. Як виконати підсумок?
30. Як виконати обчислення за необхідною функцією?
31. Як виконати редагування формули в комірці?
32. Що таке абсолютне та відносне посилання?
33. Вкажіть де знаходиться поле авто розрахунку та які можливості воно надає?
34. Назвіть способи вставки функцій у комірку?
35. Які логічні функції Excel вам відомі?
36. Скільки вкладених функцій може містити логічна функція ЕСЛИ?
37. Які способи створення діаграм?
38. Які елементи діаграм існують?
39. Як задати діапазон комірок для створення діаграм? Для чого призначені кнопки мінімізації?
40. Що таке Ряд даних та Категорія? Як задати імена рядів даних?
41. Як здійснити форматування елементів діаграми?
42. У чому різниця між круговою діаграмою та гістограмою?
43. Дайте визначення макросу.
44. Яка послідовність створення макросу?
45. Як запустити створений макрос на виконання?

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Microsoft Excel 2016.
Використання електронних таблиць для створення та обробки баз даних

Зміст

Microsoft Excel 2016. Використання електронних таблиць для створення та обробки баз даних 44
1. *Бази даних в Microsoft Excel. Основні визначення та вимоги* 44
2. *Особливості створення списків даних з використанням контролю введення* 45
3. *Фільтрація даних.* 48
Правила формування критеріїв пошуку при фільтрації даних 48
Фільтрації в Excel з використанням інтелектуального фільтра. 49
Розширений фільтр та його застосування 51
4. *Сортування бази даних* 52
5. *Формування підсумкових даних* 54
6. *Консолідація списків* 55
7. *Зведені таблиці та діаграми* 56
Питання для самоконтролю 59

1. *Бази даних в Microsoft Excel. Основні визначення та вимоги*

Для створення повноцінних баз даних доцільно використовувати спеціалізовані програмні продукти, наприклад, Microsoft Access. Ці програми досить складні та потребують значних витрат часу на їх вивчення та опанування. Тому для створення простих баз даних (БД) варто скористатися Excel, який дозволяє створювати прості бази даних, що складаються з однієї таблиці.

Будемо вважати, що **база даних** – це сукупність відомостей про конкретні об'єкти або події, яка зберігається у електронному виді. Базами даних у Excel називають таблиці певного виду (їх ще називають **списками даних**), у яких зберігається подібна за своїми властивостями інформація у виді рядків (рис. 3.1). Списки даних створюють не стільки для обчислення даних, скільки для зберігання в зручному структурованому вигляді великих обсягів інформації. Наприклад, списками даних є розклади руху транспорту, адресні книги, прайс-листи тощо.

При роботі зі списками даних, рядки називають **записами**, а стовпці – **полями**. Одним з вимог до створення списків є те, що у полі повинні розміщуються дані тільки **одного типу**: числа, тексти або дати. Кожне поле (4) повинно мати назву. Зазвичай імена полів повинні бути **унікальними**, **інформативними** та **лаконічними**. Сукупність імен усіх полів створюють **заголовний рядок бази даних** (2), який для зручності повинен відрізнятися від інших записів кольором, шрифтом або обрамленням. При цьому не дозволяється розділяти порожніми клітинками назви полів та об'єднувати комірки. Кожна назва поля повинна розміщатися в однієї клітинці таблиці Excel. Якщо на аркуші розміщується декілька списків або якась інша інформація, то вона повинна бути відокремлена від списку порожніми рядками або стовпцями. Якщо у рядку електронної таблиці вводять назву бази (1), то її відокремлюють пустим рядком.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Починаючи з рядка, наступного за заголовним рядком, вводять записи бази даних (3). Запис може складатися з **різнотипних** даних, що змістовно зв'язані між собою. Кожен запис є окремою характеристикою об'єкта або події. Не дозволяється розділяти записи бази даних порожніми рядками.

Підсумуємо, основними правилами створення списків у Excel є:

- перший рядок бази даних повинен містити заголовки стовпців;
- кожний наступний рядок бази даних повинен містити хоча б одну заповнену даними клітинку;
- не слід застосовувати об'єднані комірки в просторі таблиці бази даних;
- кожен стовпець повинен містити тільки один тип даних - або текст, або числа, або дати;

Прізвище	Рік народження	Адреса	Середній бал
Герасімов	1982	Керчь	4
Сидоров	1981	Київ	4
Іванченко	1980	Харків	4,5
Гранін	1979	Ізмаїл	5
Іванов	1983	Донецьк	3
Сєдов	1980	Ізюм	4,8
Ісаков	1983	Харків	5

Рис. 3.1 Основні елементи бази даних

назва БД (1); заголовний рядок (2); запис (3); поле (4)

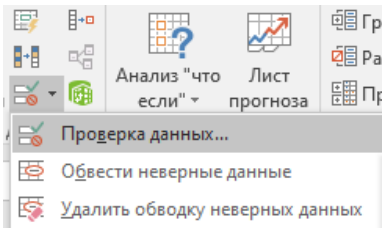
- навколо бази даних комірки повинні бути порожніми, або це повинен бути край робочого листа.
- бажано привласнити діапазону бази даних ім'я.

2. *Особливості створення списків даних з використанням контролю введення*

Безумовно, таблиця може бути заповнена даними звичайним способом, однак при цьому не виключена ймовірність помилок, які істотно впливають на роботу з базою даних в подальшому. Найбільш часто помилки виникають через те, що деякі дані чуються і пишуться по-різному. Наявність таких помилок призведе до того, що не всі потрібні дані будуть вибрані з таблиці. Щоб виключити подібні помилки та істотно прискорити заповнення таблиці даними, використовується механізм перевірки даних, який є особливо ефективним для обмежених наборів даних: назви районів чи причин пожеж, список співробітників, назви техніки тощо.

Ця операція називається перевіркою введення. Вона дає змогу контролювати тип інформації, яка вводиться у заздалегідь вибраний діапазон. При цьому можна задати виведення на екран попереджувального повідомлення перед введенням значення в комірку, яка перевіряється, а також повідомлення про помилку. Останнє буде з'являтися на екрані під час спроби введення в комірку значення, що не задовольняє накладеним обмеженням.

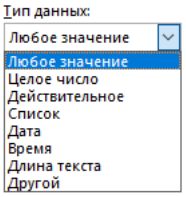
©НУЦЗ України, 2019
Http://www.nuczu.edu.ua



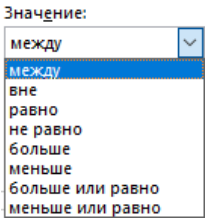
Для встановлення перевірки введення спочатку потрібно виділити діапазон клітиною у яких буде здійснюватися перевірка. Зазвичай це поле (нагадаємо, що у полі розміщуються дані одного типу), але може бути і рядок, і довільний діапазон. Далі виконується команда **Дані – Перевірка даних**.

З'являється вікно **Перевірка значень, що вводяться** із трьома вкладками. Активна вкладка **Параметри** (рис. 3.2) використовується для задання обмежень на тип і діапазон даних, які вводяться.

Для завдання умов перевірки потрібно вибрати тип даних, які можна вводити у клітинки. Дозволяється встановлення наступних типів:



- будь-яке значення (перевірка не проводиться);
- ціле або дійсне число;
- дата або час;
- довжина тексту (тільки для перевірки текстових значень);
- список.



При виборі будь-якого типу, крім останнього, у списку **Значення** потрібно вибрати оператор порівняння для обмеження діапазону можливих значень. Можна встановити такі оператори: **між**, **поза**, **дорівнює**, **не дорівнює**, **більше**, **менше** тощо. Після цього у вікні з'являються поля для введення конкретних значень обмежень. Якщо обраний оператор **між** або **поза**, то з'являється два поля, в інших випадках - одне.

При використанні типу **Список** потрібно заздалегідь створити список можливих значень, які можна вводити у вибраний діапазон. Списки повинні розташовуватися за межами основної таблиці як мінімум через один стовпець (можна використовувати окремий аркуш у книзі). У припущенні, що основна таблиця буде постійно збільшуватися не варто розташовувати списки під нею. Списки доцільно розташовувати в стовпець і під кожним залишати по кілька вільних клітинок, це дозволить в подальшому їх легко збільшувати.

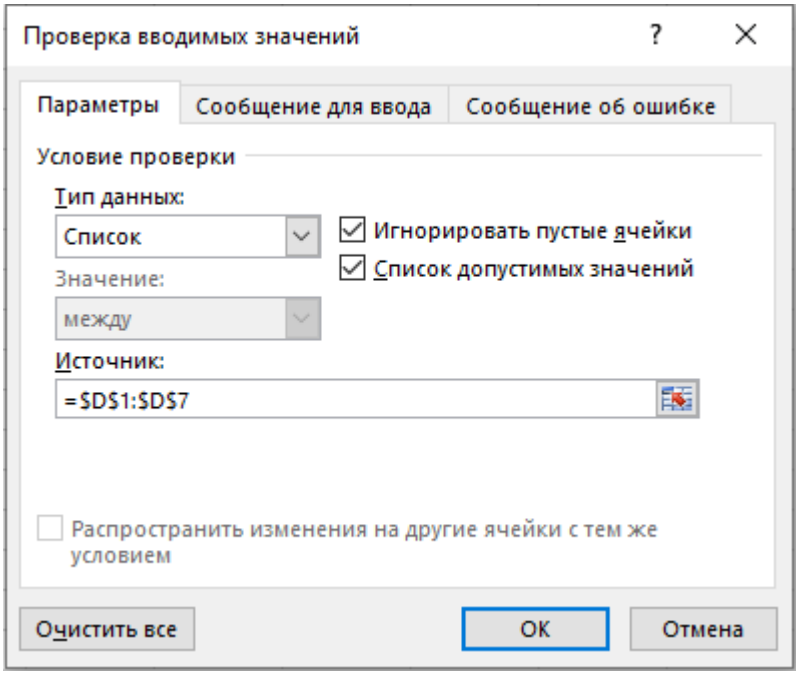


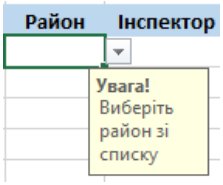
Рис. 3.2 Вікно Перевірка значень, вкладка Параметри

Перелік лекцій

Після вибору типу **Список** у вікні з'являється поле **Джерело**, в яке потрібно ввести ім'я діапазону, де розташований список. Доцільно захопити кілька порожніх клітинок нижче списку, що дасть можливість в подальшому вносити в список додаткові значення. В результаті в полі буде вказано діапазон з абсолютними посиланнями на клітинки, про що свідчить значок **$** (рис. 3.2). Тепер при введенні значень на робочому листі біля активної комірки з'являється кнопка списку, що розкривається, із переліком усіх можливих для неї значень.

Наступна вкладка **Повідомлення для вводу** (рис. 3.3) використовується для формування вікна повідомлення, яке буде з'являтися на екрані при активізації клітинки із діапазону, що перевіряється. Воно повинно мати заголовок і власне текст підказки, які вводять у відповідні поля вкладки.

Заповнення цих полів не є обов'язковим, але у цьому разі підказування не відображається на екрані. Проте, якщо повідомлення все ж таки буде сформовано, воно нагадає користувачеві про обмеження введення до того, як буде зроблена помилка.



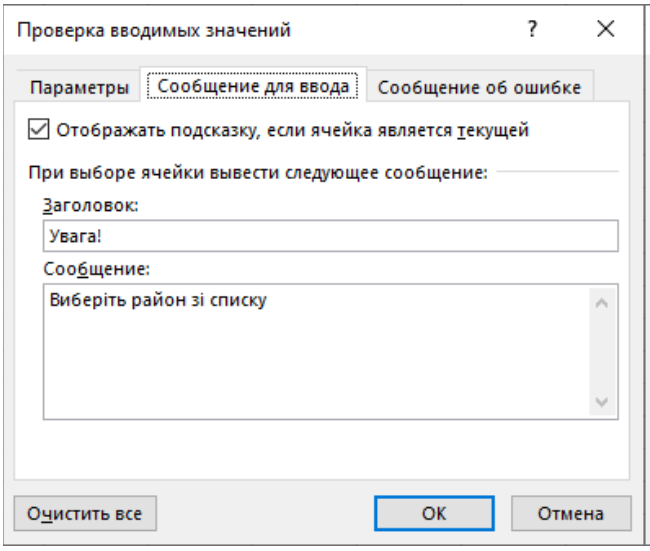


Рис. 3.3 Вікно Перевірка значень, вкладка Повідомлення для вводу

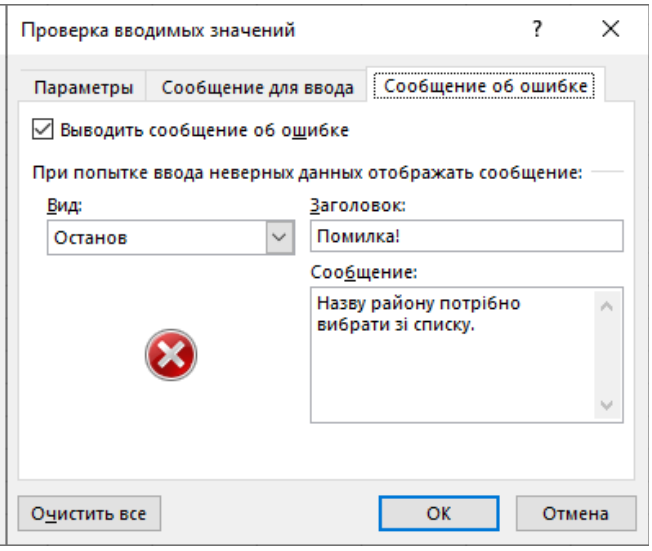
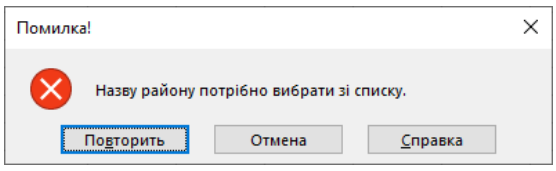


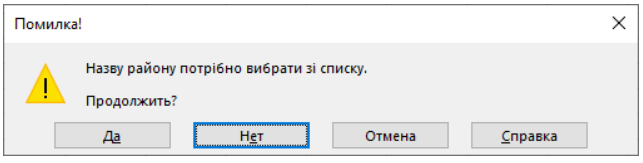
Рис. 3.4 Вікно Перевірка значень, вкладка Повідомлення про помилку

Остання вкладка **Повідомлення про помилку** (рис. 3.4) використовується для формування діалогового вікна повідомлення про помилку, яке з'являється при спробі помилкового введення. На цій вкладці зі списку **Вид**, вибирають потрібний тип вікна: **Зупин**, **Попередження** або **Повідомлення**. Призначений тип визначає вигляд діалогового вікна і подальші дії користувача:

- **Зупин** – цей вид діалогового вікна передбачає або повторне введення, або його скасування за допомогою кнопок **Повторити** та **Відмінити**. Введення помилкового значення у цьому випадку неможливе.

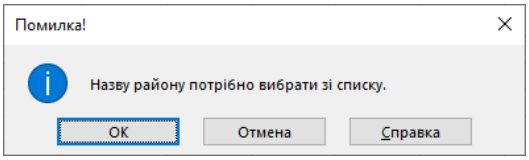


- **Попередження** – дає змогу підтвердити (дозволити) введення помилкового значення (кнопка **Так**),



Перелік лекцій

виправити помилку (кнопка **Ні**) або скасувати введення (кнопка **Відмінити**);



- **Повідомлення** – дає змогу дозволити введення неправильного значення (**ОК**) або скасувати введення (**Відмінити**). На відміну від попередніх цей тип не пропонує виправити помилку, залишаючи цю дію за користувачем.

Аналізована вкладка має також текстові поля для введення заголовка діалогового вікна і повідомлення про помилку. Заповнення цих полів не є обов'язковим, і у цьому разі система сама формує діалогове вікно про помилку за типом **Зупин**. Воно містить наступний текст: **Це значення не відповідає обмеженням по перевірці даних, встановленим для цієї клітинки**.

3. Фільтрація даних.

База даних може включати велику кількість записів, з яких для виконання конкретного завдання потрібні тільки деякі. Відображення всіх даних на аркуші ускладнює вирішення завдання. Виділення підмножини з набору даних називається фільтрацією. Для проведення фільтрації користувач повинен сформувати критерії пошуку.

Правила формування критеріїв пошуку при фільтрації даних

Критерій пошуку – це умови, яким повинні задовольняти записи БД, які потрібно знайти. Під час фільтрації йде порівняння умови, яка сформована у критерію, з записами бази даних. Результат перевірки критерію є два стани: Істина (приймає значення логічної 1) та неправда (приймає значення логічного 0).

Критерій пошуку складається з назви і значення, які з'єднуються операцією порівняння. У якості назви критерію виступають імена полів БД, значенням критерію може бути числова або текстова константа, дата або час.

Для пошуку інформації у текстових полях можна використовувати спеціальні шаблони, до яких відносять ***** (заміняє собою **будь-яку** кількість символів у даній позиції) та **?** (заміняє собою **один** символ у даній позиції).

Критерії пошуку можуть бути заданими за наступними принципами.

Принцип збігу, який передбачає **повну ідентичність** значення критерію з аналізованим при пошуку значенням поля. При цьому назва та значення поєднуються операцією **дорівнює**. (наприклад, Рік народження дорівнює 1980).

Селективний спосіб. При цьому способі вибір запису передбачається в тому випадку, коли значення його поля **належить інтервалу**, визначеному у критерії. Зв'язати назву і значення критерію можна однією з таких операцій порівняння: більше (>), більше або дорівнює (>=), менше (<), менше або дорівнює (<=), не дорівнює (<>). Такий спосіб використовують для числових значень, а також значень типу дата або час (наприклад, середній бал >8).

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Для пошуку текстових значень інтервал задають за допомогою шаблонів ***** та **?**, при цьому назва та значення поєднуються операцією дорівнює (наприклад, прізвище дорівнює С*).

Якщо пошук даних потрібно зробити за кількома критеріями одночасно, то їх об'єднують знаками логічних операторів **ТА** (логічне множення), **АБО** (логічне додавання).). Такий вираз моє назву **логічної функції**.

X	Y	X та Y	X або Y
0	0	0	0
1	0	0	1
0	1	0	1
1	1	1	1

Якщо один з критеріїв пошуку позначити як **X**, а другий – як **Y**, то для визначення результату перевірки умов складного критерію можна скористатися таблицею істинності логічних функцій. З якої видно, що операція логічного **ТА** дає значення **Істина** тільки в тому випадку, якщо значення **усіх** її критеріїв (умов пошуку) є істиною, операція логічного **АБО** дає значення **Істина** в тих випадках, якщо значення **хоча б одного** з її критеріїв є істиною.

Фільтрації в Excel з використанням інтелектуального фільтра.

У Excel є спеціальна програма автоматизації введення критеріїв, яка називається **Автофільтр**. Застосування автофільтру здійснюється командою **Дані - Фільтр**. При цьому для кожного стовпця таблиці автоматично створюється набір фільтрів, який складається з даних цього стовпця.

Дата	Район	Інспектор
12.сен	московський	Сергій Гринько
14.сен	шевченківський	Олег Петров
12.окт	московський	Сергій Гринько

Якщо режим автофільтру включено у всіх полях заголовного рядка бази з'являється кнопка викликання списку, що відображаються праворуч від заголовки кожного стовпчика.

Якщо натиснути кнопку фільтра, то її значок зміниться з трикутника на воронку і відкриється діалог з варіантами фільтрів в якому містяться всі наявні (можливі) значення цього поля (рис. 3.5). За замовчанням використовується варіант фільтрації **Виділити все**, що позначає, що записи бази даних повинні відображатися без фільтрації.

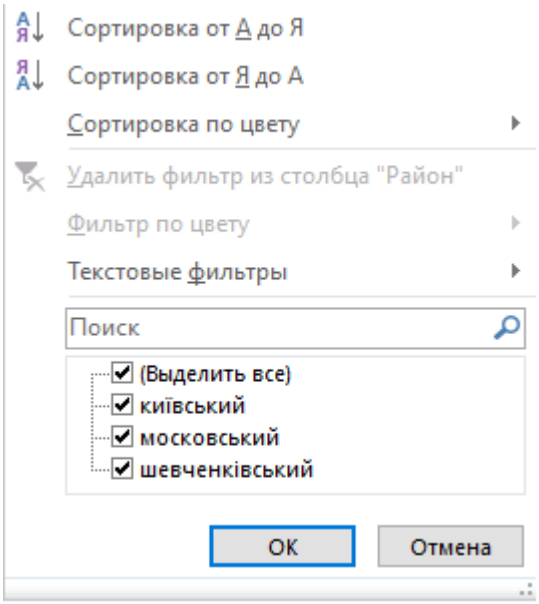


Рис. 3.5 Діалог з варіантами фільтрів

Список можливих значень дає змогу шукати та відображати дані за принципом збігу, наприклад, залишити на екрані всі записи, що стосуються певного району або дати.

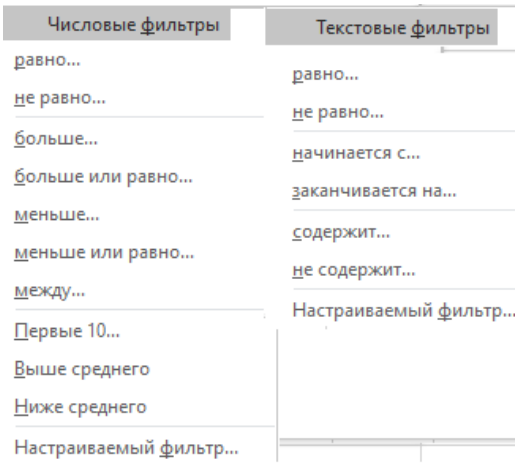
При застосуванні фільтра записи, які не потрапляють в відібрану підмножину, приховуються і не відображаються на аркуші. Загальна кількість записів бази даних і

Перелік лекцій

число відібраних записів відображається в рядку стану. Відфільтрована база даних може використовуватися при друку (друкуються тільки записи, які відносяться до відфільтрованої підмножини). Також при побудові діаграм вона будується на основі відфільтрованих даних . В останньому випадку зміна критеріїв фільтрації автоматично змінює вид діаграми.

Початкова база відновлюється за допомогою команди **Виділити все** списку, що розкривається, вибраного для фільтрації поля або за командою **Дані - Очистити**.

Для використання селективного способу завдання критеріїв використовуються додаткові фільтри.



Для стовпців, що містять цифрові дані список фільтрів відкривається за допомогою опції **Числові фільтри**. Відповідно для стовпців, що містять текстові дані опція має назву **Тексові фільтри**.

Стовпці з числами або текстом, до яких застосовуються фільтри розпізнаються Excel автоматично.

Окрім звичайних операторів порівняння для числових фільтрів в цьому списку присутня команда **Перші 10**, яку використовують для забезпечення виводу на екран визначеної кількості найбільших або найменших значень вибраного поля.

Дана команда активізує відповідне діалогове вікно (рис. 3.6), у якому кількість указують за допомогою поля-лічильника (за замовчанням 10), а які саме значення треба показати (найменші або найбільші) - за допомогою першого списку, що розкривається, у другому списку вибирають параметр: **елементів списку** або **% від кількості елементів**.

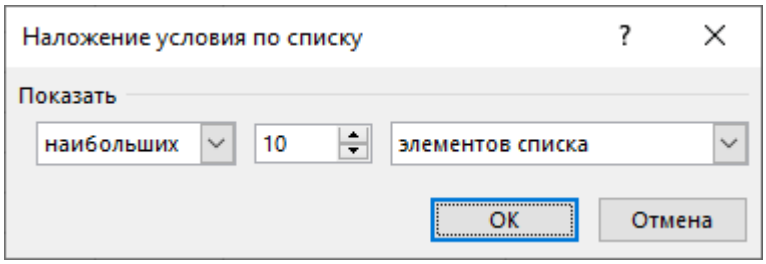


Рис. 3.6 Вікно накладення умови за списком

Для використання більш складних умов пошуку, як для числових, так і для текстових фільтрів використовується команда **Настроюваний фільтр** котра активізує діалогове вікно **Користувальницький фільтр** (рис. 3.7). Тут можна задати одну або дві умови вибірки для одного поля бази. Шукати дані можна як за принципом збігу, так і селективним способом.

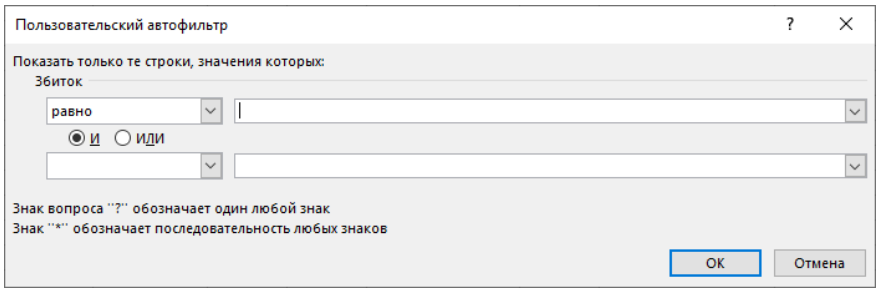


Рис. 3.7 Вікно Користувальницький фільтр

Перелік лекцій

У вікні відображається ім'я вибраного поля і є дві пари полів-списків. Перша пара використовується для задання одного критерію. У перше поле вводять оператор порівняння, а в друге – значення. Другу пару полів-списків використовують, якщо задаються відразу два критерії. Між ними розташовані два перемикачі **Та** і **Або**, які задають логічну функцію зв'язку.

Фільтрування записів бази даних може мати багатоступеневий характер. Його можна продовжити і застосувати до результату фільтрації, задаючи послідовно значення критеріїв пошуку для інших полів. Очевидно, що вони зв'язуються оператором логічного **Та**.

Вимкнення режиму автофільтра виконується повторним виконанням команди **Дані - Фільтр**. Якщо на поля бази даних були накладені якісь умови пошуку, то ці умови очищуються та усі записи відновлюються.

Розширений фільтр та його застосування

Результати автофільтрації записів бази даних відображаються на екрані, заміщаючи базу. Інші записи бази при цьому зберігаються, але вони стають для користувача невидимими.

Проте Excel передбачає можливість розмістити результати пошуку в будь-якому вільному від бази діапазоні робочого листка, а також створити спеціальний діапазон для великої кількості умов-фільтрів, які можуть знадобитися користувачеві під час роботи з базою даних. Так реалізується трохи інша технологія пошуку, яка називається розширеним фільтруванням. Розміщення блоків результату і критеріїв щодо бази даних довільне.

Блок результату за структурою, як правило, аналогічний початковій базі, але деяких її полів в ньому може не бути.

Блок критеріїв містить імена полів і значення, які потрібно знайти. Імена полів у блоку критеріїв мають бути абсолютно ідентичними іменам полів в базі; тому, щоб уникнути зовні не помітних помилок, їх краще скопіювати. Якщо в клітинки, які розташовані відразу ж під ім'ям поля, вводять кілька значень, то пошук буде здійснюватися за кількома критеріями одночасно. Причому вони зв’язуються логічним оператором **Або** і призначаються для пошуку в одному полі запису.

Район
київський
московський

Кілька критеріїв також можна розмістити по горизонталі в одному рядку. У цьому разі в клітинки, які розташовані під іменами полів, вводять одне значення. Кожний з таких критеріїв може використовуватися окремо, а сусідні - одночасно і призначатися для пошуку як в одному полі запису, так і в кількох полях. У будь-якому випадку імена потрібно вводити щоразу, коли задаються значення для вибірки. Зв'язуються такі умови пошуку оператором **Та**.

Район	Причина
московський	Підпал

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Технологія розширеного пошуку реалізується за командою **Дані – Додатково**. Перед її запуском на робочому листку з базою даних заздалегідь створюють блок критеріїв. Потрібно також заздалегідь створити блок результату, якщо планується виведення результату вибірки в окреме місце робочого листка. Як правило, його формують копіюванням заголовного рядка бази в рядок листка, який розташовано на 2-3 рядки нижче останнього запису бази. Розширений фільтр запускають командою **Дані – Додатково**, яка викликає на екран однойменне діалогове вікно (рис. 3.8).

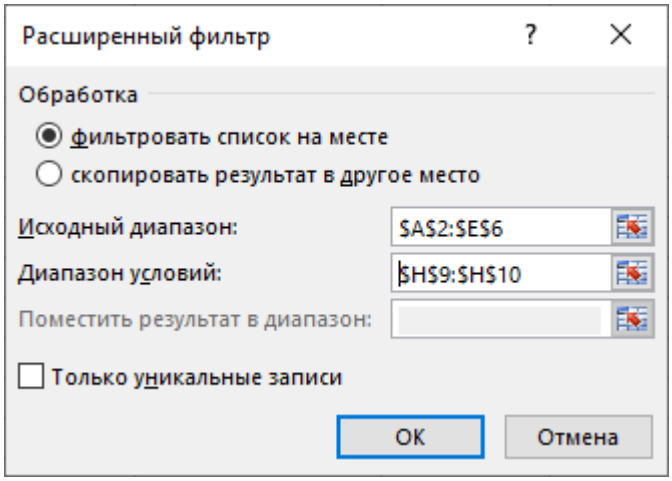


Рис. 3.8 Вікно Розширеного фільтру

У вікні задають:

- режим обробки бази даних (скопіювати результат в інше місце робочого листа або залишити на місці розміщення бази);
- діапазон розміщення початкової бази даних та діапазон умов, де розміщені вибрані користувачем критерії пошуку;
- діапазон, куди потрібно вивести записи, які будуть вибрані в результаті пошуку (діапазон розташування заголовного рядка блока результату). Цей діапазон задають у тому випадку, якщо встановлений перемикач **скопіювати результат в інше місто**.

Абсолютні адреси перерахованих діапазонів зручно вводити у відповідні поля вікна виділенням їх комірок на робочому листку.

> При виконанні будь-яких операцій з базою даних перед виконанням відповідної команди потрібно встановити рамку виділення у будь-яку комірку бази даних для коректного розпізнання таблиці-списку.

4. *Сортування бази даних*

Людині, як правило, легше працювати з даними, упорядкованими за визначеними правилами. Наприклад, шукати відомості про пожежі за конкретним районом міста легше, коли весь список упорядкований за алфавітом.

Записи бази даних, а також і дані довільного діапазону робочого листка, можна упорядкувати (відсортувати) за зростанням (убуванням) значень вибраних ключів. Як ключі сортування можуть виступати назви полів списку, найменування стовпчиків (A, B, C, D, ...) або номери рядків електронної таблиці. Сортування може здійснюватися за одним, двома або більше ключами одночасно.

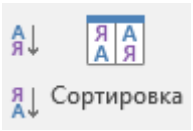
У результаті сортування за зростанням (убуванням) ключа тексти розташовуються за алфавітом від А до Я (в оберненому порядку від Я до А). Порядок розташування чисел і дат очевидний: числа - від меншого до більшого (від більшого до меншого), а дати - від більш ранньої до більш пізньої (від більш пізньої до більш ранньої).

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Необхідність повторного сортування випливає з вимог поставленої задачі, коли вміст двох і більше комірок першого ключового стовпця (рядка) збігається. При цьому впорядкування продовжується за другим ключем, характер зміни якого не залежить від першого, тобто перший ключ може бути таким, що зростає, а другий – та-ким, що спадає, і навпаки.

За замовчанням встановлені режими сортування даних в рядках таблиці і без урахування регістрів літер.



Швидке сортування даних за зростанням (убуванням) одного ключа здійснюється двома однойменними кнопками, які розташовані на вкладці **Дані**, група **Сортування та фільтр**. Якщо впорядковується база, то в якості ключа сортування встановлюється то поле, у якому знаходилася рамка виділення. Якщо це довільний виділений діапазон, то його рядки з даними впорядкуються за зростанням (спаданням) значень першого стовпця.

Для більш розширеного встановлення налаштувань, сортування виконується командою **Дані – Сортування**, яка активізує однойменне вікно (рис. 3.9).

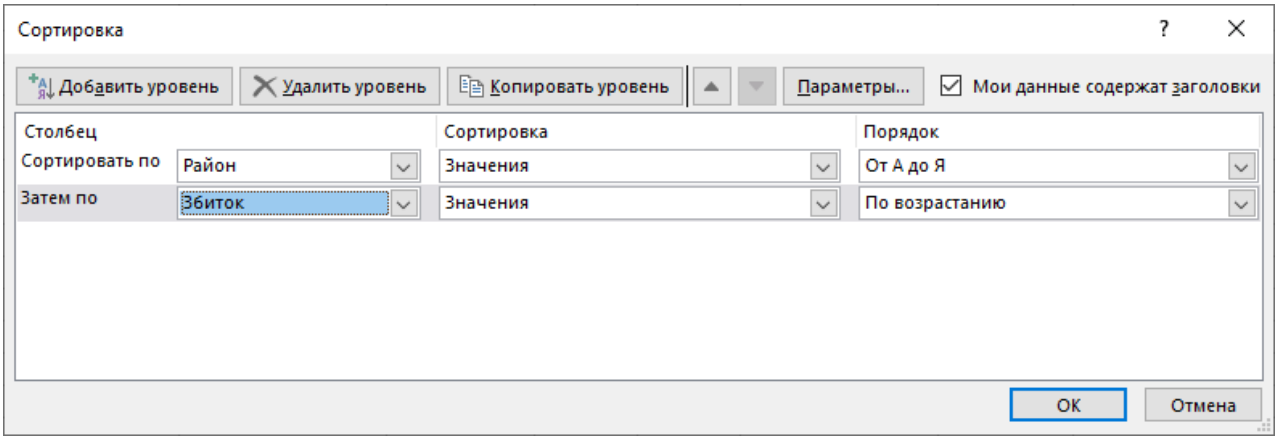


Рис. 3.9 Вікно Сортування

Якщо впорядковується база даних, на робочому листку автоматично виділяються усі її записи без заголовного рядка. У вікні **Сортування** перемикач **Мої дані містять заголовки** буде активним, при цьому ключами сортування є імена полів бази (значення комірок першого над виділеним діапазоном рядку).

Для встановлення режимів сортування, для кожного с ключів за допомогою списків, що розкриваються, послідовно вибирають назву поля та порядок сортування. Характер сортування для числових полів визначається **за зростанням** або **за спаданням**, для текстових полів – **від А до Я** чи навпаки, а для дати – **від старих до нових** та навпаки. Окрім цього можливо встановити нестандартний характер сортування за допомогою списків автозаповнення, які перераховані у вікні **Списки** (**Файл – Параметри – Додатково –** кнопка **Змінити списки**). Кількість вибраних ключів регулюється за допомогою кнопок **Добавити рівень** (або **Видалити рівень**) та випливає з вимог поставленої задачі. Кількість ключів-рівнів сортування не обмежена, але на практиці більш ніж трьома не користуються.

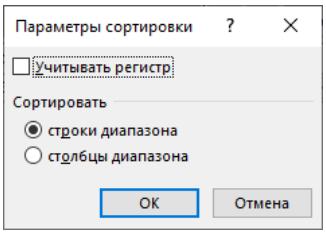
Якщо обрано тільки один ключ сортування, то в результаті сортування всі записи бази розташуються в порядку зростання або убування значень цього поля.

Перелік лекцій

Необхідність сортування за двома ключами виникає, коли декілька значень **першого** ключа збігаються.

Після сортування по двох ключах, значення першого ключа розташуються у виді груп з повторюваними значеннями, а другого ключа - в порядку зростання (убування) у межах **груп повторюваних значень** першого ключа. Відповідно необхідність використання третього ключа виникає тоді коли декілька значень **другого** ключа повторюються.

Додаткові можливості Excel щодо сортування даних надає використання кнопки **Параметри**. Клацання на неї викликає на екран діалогове вікно **Параметри сортування**. Перемикачі вікна дають змогу відсортувати дані з урахуванням станів регістру символів та що сортувати: стовпці або рядки діапазону.



5. *Формування підсумкових даних*

Зручним засобом аналізу даних бази є підведення підсумків. За його допомогою, згрупувавши початкові дані за певною ознакою, можна обчислити проміжні та остаточні підсумки. Групування здійснюється сортуванням бази за полями (ключами), для яких є намір отримати підсумки. Як наслідок, записи з однаковими значеннями вибраного поля будуть об'єднаними в блоки. Так, наприклад, для обчислення значень загальних збитків у кожному районі міста і по місту записи потрібно відсортувати за полем **Район**; для отримання підсумків за кожною датою – за полем **Дата**.

Власне обчислення підсумків проводиться через команду **Проміжний підсумок**, вкладка **Дані** група **Структура**. Вона активізує діалогове вікно **Проміжні підсумки** (рис. 3.10) з трьома списками.

У першому списку (**При кожній зміні в**) вибирають підсумкове поле по якому підводять проміжні підсумки та яке перед цим повинно бути **обов'язково відсортоване**, у другому - функцію, що використовують для обчислення підсумків (доступні варіанти **Сума**, **Кількість**, **Середнє**, **Максимум**, **Мінімум**, **Добуток** тощо), а в третьому - одне або кілька полів, де підбивають підсумки. Требі бути уважним та не вибирати текстові поля для яких не можуть бути розраховані Сума або Середнє.

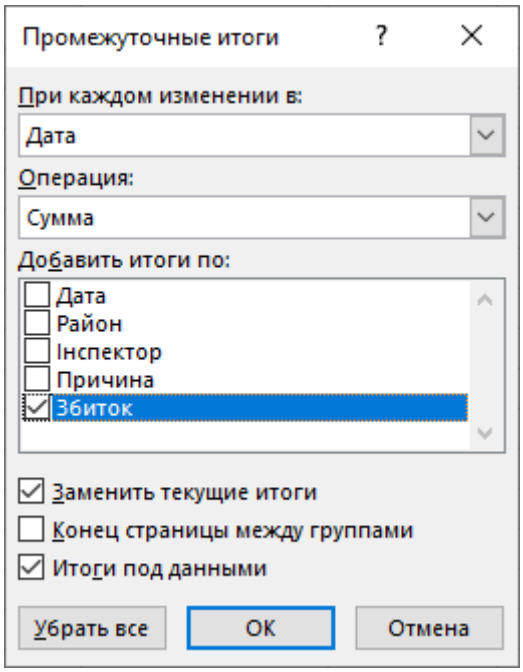
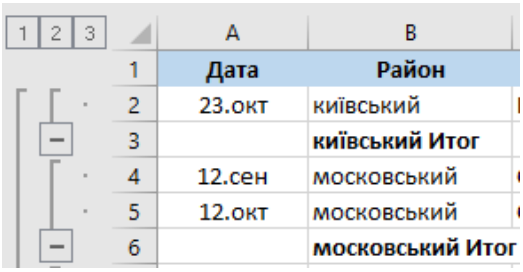


Рис. 3.10 Вікно Проміжні підсумки

Після підведення проміжних підсумків у рядку з іменами стовпців електронної таблиці з'явилися три кнопки з номерами 1, 2 і 3.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua



	A	B
1	Дата	Район
2	23.окт	київський
3		**київський Итог**
4	12.сен	московський
5	12.окт	московський
6		**московський Итог**

За їх допомогою на екрані можна відобразити або тільки загальний підсумок по усій базі даних, або тільки підсумки по кожній групі і загальний підсумок, або усі початкові дані і підсумки. Маніпулюючи кнопками зі знаками **–** поруч із номерами рядків з підсумками, можна сховати групи записів з початковими даними. Знак **–** при цьому змінюється на знак **+**. Клацання мишею на останньому призводить до появи записів, що були сховані.

Використовуючи інші функції зі списку **Операція** вікна **Проміжні підсумки**, можна послідовно вводити до бази даних різні підсумкові рядки і вилучати їх. Для додавання нового підсумкового рядка до тих, що вже є, треба вимкнути прапорець **Замінити поточні підсумки** (рис. 3.10). Початковий стан бази даних відновлюють за допомогою кнопки **Убрати все** даного вікна.

6. *Консолідація списків*

Працюючи зі списками даних, часто приходиться мати справу з однотипними списками. Такі списки, звичайно, містять відомості про ті ж самі об'єкти, але за різні тимчасові або звітні періоди. Наприклад, один список може містити відомості про пожежі по областях України, що виникли з різних причин, за один рік, а другий список - аналогічні відомості, але за інший рік. Буває необхідним ці відомості об'єднати в один загальний список, щоб підвести результати по кожної області за увесь період. Наприклад, визначити середнє значення кількості пожеж по кожній області України, в залежності від причини, за два звітних роки. Такий список можна сформувати за допомогою операції, яка називається консолідацією списків.

Консолідація - це операція, яка дозволяє в одній комірці загального списку об'єднати значення з множини комірок списків-джерел. При цьому є можливість забезпечити оновлення даних у загальному списку, якщо дані в списках-джерелах будуть змінюватися.

Значення об'єднуються, відповідно до вибраної математичної функції. Найчастіше для підведення результатів використовують такі функції: **сума**, **середнє**, **максимум**, **мінімум**, **добуток**.

> Щоб консолідація була можливою, назви полів списків-джерел повинні цілком збігатися. Причому перше поле цих списків має бути текстовим (назва конкретного об'єкта), а інші поля - числовими.

Якщо передбачається створювати оновлення даних у загальному списку, потрібно, щоб початкові дані списки розміщувалися в окремих робочих книгах та зберігалися у файлах з короткими інформативними іменами. У тому випадку коли оновлення даних не потрібно, початкові списки можуть дути розміщені в одній книзі на різних листах або навіть на одному листі в різних діапазонах клітинок.

Для виконання консолідації спочатку потрібно відкрити усі файли з початковими списками.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

На вільному листку поточної або нової книги потрібно вибрати місце розташування загального списку, який буде створено, і встановити рамку виділення у його верхню ліву клітинку.

Виконати команду **Дані –Консолідація** (група **Робота з даними**), що призведе до відкриття однойменного діалогового вікна (рис. 3.11).

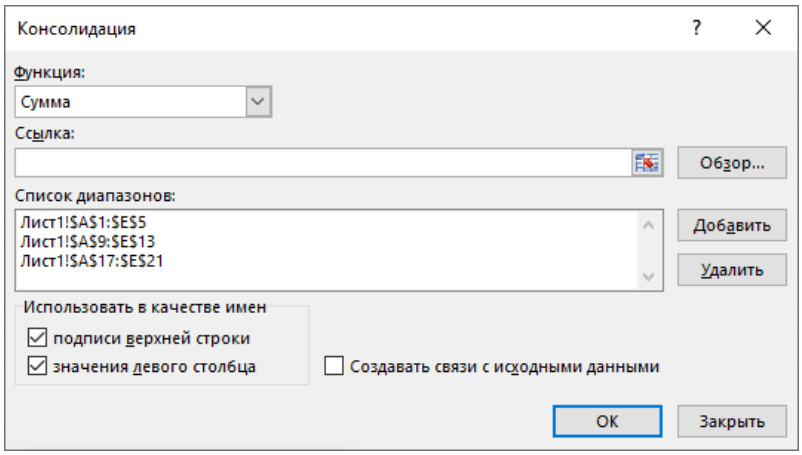


Рис. 3.11 Вікно Консолідація

Зі списку **Функція**, що розкривається, функцію, яку потрібно використовувати для опрацювання даних та ввести в поле **Посилання** діапазон розташування першого початкового списку даних, які консолідуються. Для цього спочатку потрібно активізувати лист із списком і виділити діапазон клітинок, який містить весь список із заголовним рядком. Використовуючи вікно **Консолідація** добавити діапазон у **Список діапазонів**. Аналогічно добавити усі списки, що консолідуються

Встановити всі перемикачі в групі **Використовувати у якості імені**. Загальний список в цьому випадку буде мати заголовний рядок і перший стовпець із назвами об'єктів.

> При консолідації даних заголовок першого текстового поля початкових списків не копіюється автоматично в загальний список. У разі потреби його копіюють або вводять вручну.

Для автоматичного оновлення загального списку при змінах у списках-джерелах даних, потрібно встановити прапорець **Створювати зв'язки з початковими даними**.

> Зв'язки не можна використовувати, якщо початковий та загальний списки знаходяться на одному листку робочої книги. Після установки зв'язків не можна добавляти нові списки-джерела і змінювати структуру початкових списків, які уже входять в консолідацію.

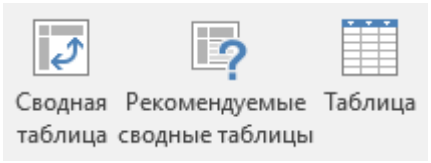
Підсумкова таблиця супроводжується кнопками з номерами 1, 2 у рядку з іменами стовпців електронної таблиці. Якщо активна кнопка з номером 1, на екрані відображаються тільки підсумкові результати по кожному об'єкту. Після клацання на кнопці з номером 2 на екрані відображаються всі значення, що стосуються кожного об'єкта, і підсумкові результати. Ліворуч від рядків з підсумками можуть розташовуватися кнопки зі знаками **–** або **+** . Маніпулюючи цими кнопками, можна виключати (включати) певні групи записів і отримувати нові підсумкові результати.

7. Зведені таблиці та діаграми

Докладний аналіз даних передбачає можливість отримання зведеної інформації із записів бази даних і такої інформації, яка прямо в базі даних не міститься. Фактично мова йде про поєднання засобів фільтрації і підсумкових обчислень. Результати

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

такого аналізу можуть бути представлені у вигляді зведеної таблиці або зведеної діаграми. При створенні зведеної діаграми автоматично створюється і зведена таблиця, яка містить дані, на основі яких будується діаграма.



Первинна побудова зведеної таблиці виконується за допомогою Майстра зведеної таблиці, який запускається командою **Вставка – Зведена таблиця** (група **Таблиці**). Але перед цим необхідно виділити будь-яку клітинку в таблиці, яка відноситься до бази даних.

На першому кроці роботи Майстра (рис. 3.12) автоматично вибирається розташування даних для аналізу. Якщо попередньо була виділена клітинка в початковій таблиці, то саме ця таблиця буде взята за основу для побудови зведеної таблиці. Про це свідчить діапазон клітинок в поле **Таблиця або Діапазон** і пунктирна лінія по контуру вихідної таблиці.

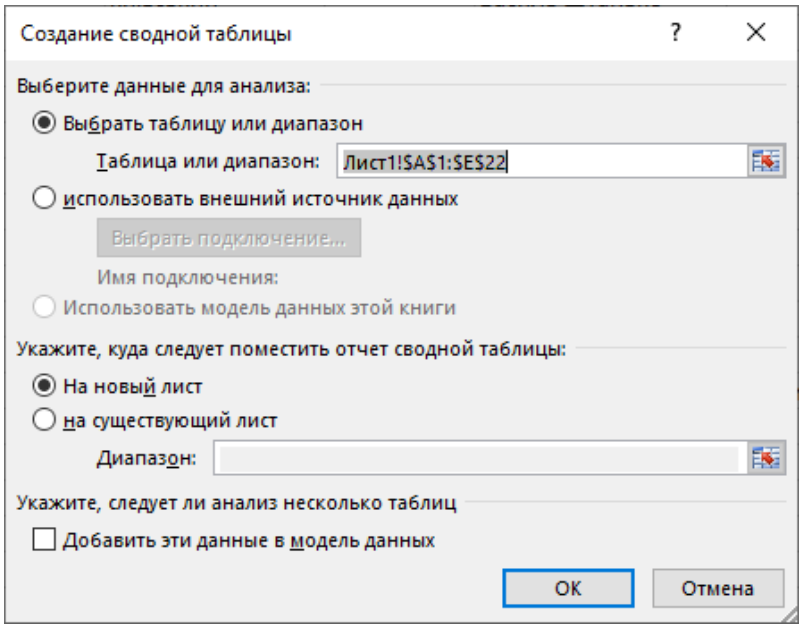


Рис. 3.12 Вікно Створення зведеної таблиці

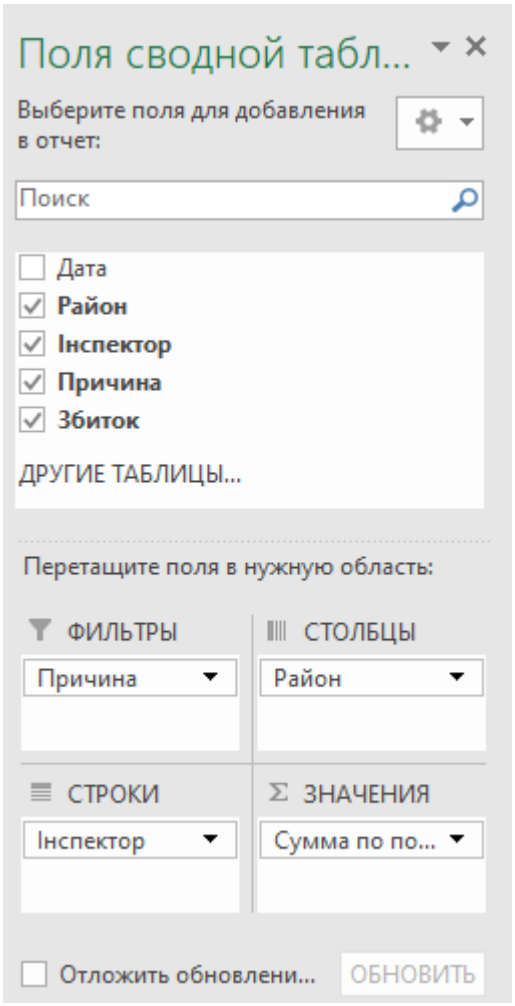
Користувач може вказати інший діапазон клітинок, виділивши тільки частину клітинок в вихідній таблиці або взагалі використати зовнішнє джерело даних, наприклад, таблицю з іншої робочої книги. У цьому ж вікні задається місце розташування зведеної таблиці. З огляду на великий розмір зведеної таблиці доцільно розміщувати її на новому аркуші.

На другому кроці створюється макет зведеної таблиці. У верхній частині вікна Майстра наводиться список стовпців початкової таблиці, а в нижній частині області зведеної таблиці. Зведена таблиця складається з чотирьох областей: **Фільтри**, **Стовпці**, **Рядки** і **Значення**.

Кожній області зведеної таблиці повинен відповідати один стовпець вихідної таблиці. Виняток робиться лише для поля **Фільтри**, яке можна залишити порожнім. Для поля **Значення** вибирається стовпець вихідної таблиці, який містить числові дані. Якщо у вихідній таблиці є велика кількість стовпців, то для побудови зведеної таблиці використовуються тільки ті стовпці, які необхідні для аналізу даних.

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Макет зведеної таблиці створюється перетягуванням назв стовпців з верхньої частини вікна в області нижньої частини вікна. Область **Фільтри** виступає зовнішнім фільтром. Дані в зведеній таблиці відносяться до значення, вибраного в цьому полі. Область **Рядки** містить значення, які визначають зміст рядків зведеної таблиці, і задає другий рівень фільтрації даних. Область **Стовпці** визначає зміст стовпців зведеної таблиці і визначає наступний рівень фільтрації даних. Область **Значення** визначає власне зміст таблиці. На відміну від всіх інших областей, до даних, які потрапляють в клітинку таблиці, застосовується функція для підсумкових обчислень (за замовчанням це **Сума**). Якщо необхідно змінити цю функцію, необхідно клацнути по кнопці праворуч і вибрати потрібну операцію зі списку на панелі **Параметри значення поля**.



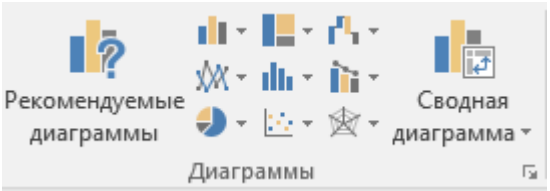
У міру перетягування стовпців на робочому аркуші Майстер автоматично будуватиме зведену таблицю, остаточний вигляд якої, наприклад, може бути таким, як на рис. 3.13. У наведеній зведеній таблиці відключені всі фільтри і тому в ній відображаються всі дані вихідної таблиці.

	A	B	C	D	E
4	**Названия строк**	**київський**	**московський**	**шевченківський**	**Общий итог**
5	Василь Штанько	1170	1500	390	3060
6	Олег Петров			4000	4000
7	Сегрій Гринько	2460	8154	2400	13014
8	**Общий итог**	**3630**	**9654**	**6790**	**20074**

Рис. 3.13 Зведена таблиця

Зведена діаграма являє собою діаграму, побудовану на основі зведеної таблиці з такими-самими можливостями аналізу даних, але у більш наочному вигляді. Якщо відразу створюється зведена діаграма, то попередньо автоматично створюється зведена таблиця.

Для створення зведеної діаграми на основі існуючої зведеної таблиці необхідно виділити будь-яку її клітинку і виконати команду Вставка – Зведена діаграма (група Діаграми). Excel автоматично створить зведену діаграму.



Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Приклад зведеної діаграми побудованої на основі зведеної таблиці (рис. 3.13) наведено на рис. 3.14

Робота зі зведеною діаграмою аналогічна роботі зі зведеною таблицею і полягає у виборі параметрів фільтрів за допомогою відповідних кнопок. Форматування зведеної діаграми здійснюється тими ж прийомами, що і звичайної діаграми. Ці прийоми детально розглянуті в лекції присвяченій графічному аналізу даних у Excel.

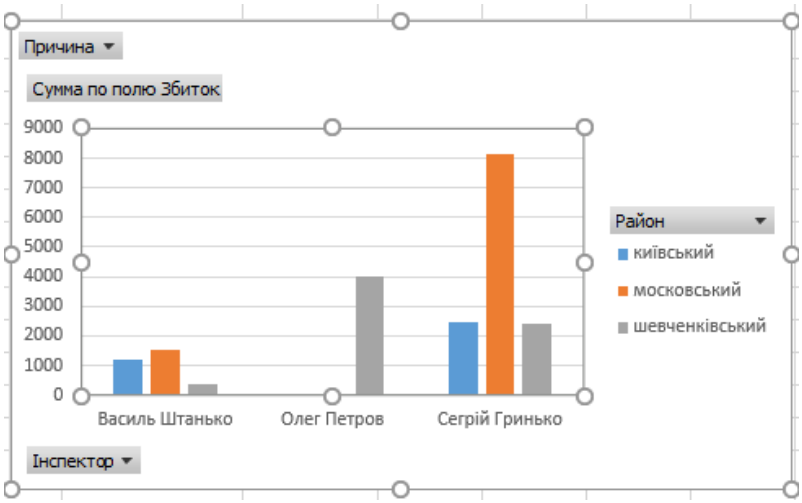


Рис. 3.14 Зведена діаграма

Питання для самоконтролю

1. Що таке база даних у MS Excel? З яких елементів вона складається?
2. Назвіть правила створення БД?
3. Дайте визначення що таке поле, запис, заголовний рядок бази даних.
4. Для чого використовується перевірка вводу у БД?
5. Яку команду потрібно виконати, щоб встановити перевірку введення для поля БД?
6. Які бувають види повідомлень про помилку? Назвіть їх відмінності.
7. Який вид повідомлення про помилку дозволяє введення помилкових даних?
8. Що таке повідомлення для введення та коли воно з'являється на екрані?
9. З чого складається критерій пошуку. Що є результатом перевірки критерію?
10. Які шаблони використовуються при пошуку текстових фрагментів?
11. Для чого призначена команда Автофільтр та Розширений фільтр?
12. Які логічні функції використовуються при об'єднанні декількох критеріїв пошуку?
13. Які переваги та недоліки кожного виду фільтрів?
14. Що таке ключ сортування? З чого він складається?
15. Які існують способи сортування?
16. Як здійснити сортування за списком?
17. Коли доцільно проводити сортування по другому та третьому ключу?
18. Як розміщуються дані, якщо встановлено характер сортування за зростанням? За спаданням?
19. Коли виникає необхідність підведення проміжних підсумків?
20. Яка операція обов'язково передує підведенню проміжних підсумків?
21. Що таке консолідація?
22. За якими командами підводяться проміжні підсумки та робиться консолідація?
23. Що таке зведена таблиця, для чого вона застосовується?
24. Наведіть структуру зведеної таблиці.
25. Яку команду потрібно виконати для створення зведеної таблиці?
26. Як побудувати зведену діаграму?

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Microsoft Access 2016.
Реляційні бази даних – створення та їх обробка за допомогою СУБД

Зміст

Microsoft Access 2016. Реляційні бази даних – створення та їх обробка за допомогою СУБД . 60
1. *Реляційна модель даних. Основні визначення та вимоги.* 60
Структура даних в реляційної моделі даних 61
Опис основних прийомів нормалізації відношень у базі даних 63
2. *Система управління базами даних MS Access 2016* 66
Загальна характеристика MS Access 2016 66
Робота з СУБД MS Access 2016 68
Питання для самоконтролю 82

1. *Реляційна модель даних. Основні визначення та вимоги.*

Реляційна модель даних - створена Едгаром Коддом логічна модель даних, що описує:

- структури даних у вигляді наборів відносин, що змінюються в часі;
- теоретико-множинні операції над даними: об'єднання, перетин, різницю і декартове множення;
- спеціальні реляційні операції: селекція, проекція, з'єднання і поділ;
- спеціальні правила, що забезпечують цілісність даних.

> Едгар Франк «Тед» Кодд - (23 серпня 1923 -18 квітня 2003 року) - британський вчений, роботи якого заклали основи теорії реляційних баз даних. Працюючи в компанії IBM, він створив реляційну модель даних. У 1970 видав роботу «*A Relational Model of Data for Large Shared Data Banks*», яка вважається першою роботою по реляційної моделі даних.

Реляційна модель даних - це спосіб розгляду даних, тобто припис для способу представлення даних за допомогою набору таблиць і для способу роботи з таким представленням за допомогою операторів. Ця модель пов'язана з трьома аспектами даних: структурою (об'єкти), цілісністю і обробкою даних (оператори).

> У 2002 журнал Forbes помістив реляційну модель даних в список найважливіших інновацій останніх 85 років.

Цілями створення реляційної моделі даних є:

- забезпечення більш високого ступеня незалежності від даних;
- створення міцного фундаменту для вирішення семантичних питань і проблем несуперечності і надмірності даних;
- розширення мов управління даними за рахунок включення операцій над множинами.

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Структура даних в реляційної моделі даних

Реляційна модель даних передбачає структуру даних, обов'язковими об'єктами якої є: відношення; атрибут; домен; кортеж; ступінь; кардинальність та первинний ключ.

Відношення - це проста плоска (двовимірна) таблиця, що складається з стовпців і рядків. **Атрибут** - це поіменований стовпець відношення (тобто таблиці). **Домен** - це набір допустимих значень для одного або декількох атрибутів. **Кортеж** - це рядок відношення (тобто таблиці). **Ступінь** визначається кількістю атрибутів, яке містить відношення. **Кардинальність** - це кількість кортежів, яке містить відношення. **Первинний ключ** - це унікальний ідентифікатор для таблиці.

Поставимо у відповідність формальний термін реляційної моделі даних та неформальне значення:

формальний термін	**неформальний термін**
відношення	таблиця
атрибут	стовпець
кортеж	рядок або запис
ступінь	кількість стовпців
кардинальне число	кількість рядків
первинний ключ	унікальний ідентифікатор
домен	загальна сукупність допустимих значень

Кожне відношення повинно підкорятися загальним правилам та властивостям, серед яких: унікальне ім'я відношення (кожна таблиця повинна мати ім’я, що не повторюється); унікальне ім'я атрибута (назви стовбців таблиць не повторюються); немає однакових кортежів (у таблицях немає однакових строк); кортежі не впорядковані зверху вниз (строки ніяк не відсортовані); атрибути не впорядковані зліва направо (стовбці теж не відсортовані); всі значення атрибутів атомарні (значення у клітинках таблиці нероздільні - нормалізоване відношення); кожен атрибут має однаковий тип (всі елементи в стовпці належать до одного типу та мають однакову довжину).

Таким чином, **реляційна база даних** - це набір нормалізованих відношень між якими встановлені зв’язки за допомогою ключів відношень.

Переваги реляційної моделі є:

- модель даних відображає інформацію в найбільш простий для користувача формі;
- заснована на розвиненому математичному апараті, який дозволяє досить лаконічно описати основні операції над даними;
- дозволяє створювати мови маніпулювання даними не процедурного типу;
- можливе маніпулювання даними на рівні вихідної БД і можливість зміни.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

До недоліків можна віднести найповільніший доступ до даних та трудомісткість розробки.

Ключі відношень в реляційній моделі даних

Ключі відношень можуть бути наступними: суперключ; потенційний ключ; первинний ключ; зовнішній ключ; сурогатний ключ.

Ключ відношень - це підсхема вихідної схеми відношень (таблиць), що складається з одного або декількох атрибутів (полів таблиці), для яких декларується умова унікальності значень в кортежі відношення (рядку таблиці).

Ключ відношень може бути простим або складеним. Простий ключ - це ключ, що складається з одного атрибута і не більше. Складений ключ – ключ, що складається з двох і більше атрибутів.

Суперключ - це атрибут або безліч атрибутів, яке єдиним чином ідентифікує кортеж даного відношення. Він може включати додаткові атрибути. Суперключ не має властивість ненадлишкових.

Потенційний ключ - це сукупність атрибутів відношення, що повинен бути унікальним та ненадлишковим. Унікальність: в таблиці немає двох різних рядків з однаковими значеннями в нашому потенційному ключі. Ненадлишковість: не можна прибрати один з стовпців з ключа, так, щоб він не втратив унікальності. У відношенні може бути більше одного потенційного ключа.

Первинний ключ (primary key) - це один з потенційних ключів відношення, обраний в якості основного ключа. Припустимо оголошення одного і тільки одного первинного ключа.

Зовнішній ключ (foreign key) - це ключ, оголошений в базовому відношенні, який при цьому посилається на первинний того ж самого або якогось іншого базового відносення.

Сурогатний ключ - це службовий атрибут, доданий до вже наявних інформаційних атрибутів відносення. Призначення сурогатного ключа - служити первинним ключем відношення. Значення цього атрибута генерується штучно.

Типи зв'язків в реляційній моделі даних

Всього існує **три** типи зв'язків: Один до одного (1 – 1); Один до багатьох (1 – ∞); Багато до багатьох (∞ – ∞).

Зв'язок «**Один до одного**»

Зв'язок один до одного утворюється, коли ключ одного відношення (ідентифікатор) присутній в іншій таблиці, в якій теж є ключем або властивостями атрибута задана його унікальність (одне і теж значення не може повторюватися в різних кортежах).

На практиці зв'язок «один до одного» спостерігається не часто. Наприклад, він може виникнути, коли потрібно розділити дані однієї таблиці на кілька окремих таблиць з метою безпеки.

Зв'язок «**Один до багатьох**»

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

У типі зв'язків один до багатьох запису першого відношення (таблиці) відповідає декілька записів в іншому відношення (таблиці).

Зв'язок «Багато до багатьох»

Якщо декільком записам з однієї таблиці відповідає декілька записів з іншої таблиці, то такий зв'язок називається «багато до багатьох» і організовується за допомогою проміжної таблиці.

Опис основних прийомів нормалізації відношень у базі даних

Нормалізація - це процес організації даних в базі даних, що включає створення таблиць і встановлення відносин між ними відповідно до правил, які забезпечують захист даних і роблять базу даних більш гнучкою, усуваючи надмірність і неузгоджені залежності.

Надмірність даних призводить до непродуктивного витрачання вільного місця на диску і ускладнює обслуговування баз даних. Наприклад, якщо дані, що зберігаються в кількох місцях, буде потрібно змінити, в них доведеться внести одні і ті ж зміни в усіх цих місцях. Зміна адреси студента набагато легше реалізувати, якщо в базі даних ці відомості зберігаються тільки в одній таблиці (*Студенти*) і ніде більше.

Що таке «неузгоджені залежності»? Користувач, якому потрібно дізнатися, наприклад, адресу якогось студента, цілком обґрунтовано буде шукати його в таблиці *Студенти*, але шукати в ній номер телефону або номер кабінету куратора цього студента, не має сенсу. Номер телефону та кабінету пов'язані з куратором (залежить від нього), а не з студентом, тому ці відомості слід зберігати в таблиці *Співробітники*. Неузгоджені залежності можуть ускладнювати доступ до даних, так як шлях до даних при цьому може бути відсутнім або бути неправильним.

Існує кілька правил нормалізації баз даних. Кожне правило називається «нормальною формою». Якщо виконується перше правило, говорять, що база даних представлена в «першій нормальній формі». Якщо виконуються три перших правила, вважається, що база даних представлена в «третій нормальній формі». Є й інші рівні нормалізації, однак для більшості додатків досить нормалізувати бази даних до третьої нормальної форми.

Перша нормальна форма:

- Усуньте повторювані групи (поля) в окремих таблицях.
- Створіть окрему таблицю для кожного набору пов'язаних даних.
- Визначте кожен набір пов'язаних даних за допомогою первинного ключа.

Не використовуйте кілька полів в одній таблиці для зберігання схожих даних. Наприклад, для спостереження за товаром, який закуповується у двох різних постачальників, можна створити запис з полями, що визначають код першого постачальника і код другого постачальника. Замість цього можна помістити всі відомості про постачальників в окрему таблицю *Постачальники* і зв'язати товари з постачальниками за допомогою кодів товарів або постачальників з товарами за допомогою кодів постачальників.

Друга нормальна форма

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

- Створіть окремі таблиці для наборів значень, що повторюються у декількох записах.
- Зв'яжіть ці таблиці за допомогою зовнішнього ключа.

Записи можуть залежати тільки від первинного ключа таблиці (складеного ключа, якщо необхідно). Візьмемо для прикладу адреса клієнта в системі бухгалтерського обліку. Ця електронна адреса необхідний не тільки таблиці Customers, але і таблицями Orders, Shipping, Invoices, Accounts Receivable і Collections. Замість того щоб зберігати адресу клієнта як окремий елемент в кожній з цих таблиць, зберігайте його в одному місці: або в таблиці Customers, або в окремій таблиці Addresses.

Третя нормальна форма

- Усуньте поля, які не залежать від ключа.

Значення, що входять до запису та не є частиною ключа цього запису, не належать таблиці. Якщо вміст групи полів може повторюватися більш ніж до однієї записи в таблиці, варто помістити ці поля в окрему таблицю.

Інші форми нормалізації

Крім описаних нормальних форм, існує четверта нормальна форма, яку також називають нормальною формою Бойса-Кодда (BCNF), і п'ята нормальна форма, але на практиці вони застосовуються рідко. Недотримання цих правил може призвести до погіршення архітектури бази даних, але на функціональності це позначитися не повинно.

Розглянемо нормалізацію на прикладі. До нормалізації існує наступна таблиця.

Name-Stud	Class1	Class2	Class3	Curator	Cur-Room
Дозоров	history, physics	electronics	chemistry	Петренко	412
Козін	philosophy	Informatics	physics	Іванов	216
Ранаєв	mechanics	history	chemistry	Саєнко	612

Це таблиця з вигаданими даними про закріплення студентів за факультативами, які вони відвідують. Також в таблицю занесено прізвище куратора та номер його робочого кабінету.

Приведемо таблицю до першої нормальної форми. Для цього усунемо повторювані групи полів. Таблиці повинні мати тільки два виміри. Наявність полів *Class1*, *Class2* і *Class3* в наведених вище записах свідчить про невдале проектування таблиці.

Електронні таблиці часто включають третій вимір, але в таблицях баз даних воно використовуватися не повинно. Розглянути цю проблему можна також за допомогою відношення «один до багатьох». Тому не потрібно включати в одну таблицю елементи, що представляють обидві сторони даного відношення.

Замість цього треба створити іншу таблицю в першій нормальній формі, усунувши повторювану групу (*Class#*).. Так як один студент відвідує кілька факультативів, то їх слід вказати в окремій таблиці.

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Також для створюваного відношення треба визначити первинний ключ. Зверніть увагу, що для першого студента (*Дозоров*) У полі *Class1* визначено два факультативи (*history*, *physics*), що суперечить вимогам реляційної моделі даних (всі значення у клітинках таблиці повинні бути нероздільними). Таким чином після приведення до першої нормальної форми таблиця має наступний вигляд:

ID_Stud	Name-Stud	Class#	Curator	Cur-Room
1022	Дозоров	history	Петренко	412
1022	Дозоров	physics	Петренко	413
1022	Дозоров	electronics	Петренко	414
1022	Дозоров	chemistry	Петренко	415
4123	Козін	philosophy	Іванов	216
4123	Козін	Informatics	Іванов	217
4123	Козін	physics	Іванов	218
4312	Ранаєв	mechanics	Саєнко	612
4312	Ранаєв	history	Саєнко	613
4312	Ранаєв	chemistry	Саєнко	614

Друга нормальна форма: усунення надлишкових даних

Зверніть увагу на те, що в наведеній вище таблиці кожне значення *Name-Stud* співвідносяться з кількома значеннями *Class#*. Значення *Class#* функціонально не залежать від значень *Name-Stud*, а це означає, що дане відношення не нормалізовано до другої нормальної форми. Усі дані, що повторюються (прізвища студентів та назви факультативів) виносимо до окремих таблиць. Другу нормальну форму представляють три наступні таблиці.

Таблиця *Students*

ID_Stud	Name-Stud	Curator	Cur-Room
1022	Дозоров	Петренко	412
4123	Козін	Іванов	216
4312	Ранаєв	Саєнко	612

Таблиця *Registration*

ID_Stud	Class#
1022	103-13
1022	101-19
1022	101-23
1022	101-18
4123	103-14
4123	106-16
4123	101-19
4312	101-12
4312	103-13
4312	101-18

Таблиця *Elective*

Class#	Class
101-12	mechanics
103-13	history
103-14	philosophy
101-25	mathematics
106-16	Informatics
101-23	electronics
101-18	chemistry
101-19	physics

Третя нормальна форма: усунення даних, що не залежать від ключа.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

В останньому прикладі для таблиці *Students* значення *Cur-Room* (номер кабінету куратора) функціонально залежать від атрибута *Curator*. Вирішити цю проблему можна, перемістивши даний атрибут з таблиці *Students* в таблицю *Faculty* та спираючись на використання первинного ключа

Таблиця *Students*

ID_Stud	Name-Stud	ID_Cur
1022	Дозоров	004-56
4123	Козін	001-45
4312	Ранаєв	005-34

Таблиця *Faculty*

ID_Cur	Curator	Cur-Room	Dept
004-56	Петренко	412	ПБ
001-45	Іванов	216	ЦЗ
005-34	Саєнко	612	ТЕБ

2. *Система управління базами даних MS Access 2016*

База даних створена за допомогою СУБД MS Access - це засіб для зберігання, організації і пошуку інформації. В основі кожної бази даних завжди лежить набір таблиць. **Моделлю бази даних** називається сукупність типів об'єктів цієї бази даних. Залежно від можливостей використання, бази даних діляться на мережеві і настільні. **Мережеві бази даних** використовують мережу Інтернет або локальну мережу, **настільні бази даних** призначені для офісної роботи на персональних комп'ютерах. Розглянемо базу даних на прикладі СУБД MS Access 2016. Файл Access 2016 це не звичайний файл, а набір об'єктів, які повинні взаємодіяти між собою і підкорятися певним принципам розробки, інакше база даних не буде працювати або працюватиме з помилками.

Загальна характеристика MS Access 2016

Система управління базами даних MS Access 2016 має наступні характеристики:

- MS Access 2016 являє собою **інтегроване середовище**, тобто в його рамках можна виконувати будь-які дії, у тому числі редагування, підключення зовнішніх об'єктів, оформлення звітів і друк;
- MS Access підтримує роботу з реляційною моделлю БД. Нагадаємо, що у реляційній базі даних записи, що утримуються в різних таблицях, зв'язані між собою певними відносинами. Це забезпечує гнучкість роботи і швидкодію бази даних;
- в MS Access можна завантажити тільки **одну активну базу даних**;
- об'єктами моделі бази даних MS Access є: таблиці, запити, звіти, форми, модулі, макроси. Наявність всіх цих об'єктів у кожній базі даних не обов'язкова;
- кожен об'єкт, створений користувачем усередині бази даних, може бути змінений або вилучений;
- всі елементи бази даних MS Access 2016 зберігаються усередині одного файлу з розширенням ACCDB.

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Таблиці становлять основну частину бази даних, у них зберігається вся інформація. Напряму з таблицями працюють тільки програмісти і розроблювачі бази даних. Стовпці таблиці називаються **полями** (атрибутами). У полях зберігаються відомості тільки певного типу. Кожне поле обов'язково повинне мати ім'я. Рядки таблиці називаються **записами** (кортежами). У записах зберігається вся інформація про окремий елемент таблиці. Перетинання поля і запису називається **коміркою**. В комірках таблиці може зберігатися тільки фактична інформація. Формули в комірках таблиці зберігати не можна. Для того, щоб розрізняти записи і зв'язувати між собою таблиці, кожна таблиця повинна мати спеціальне поле, що називається **ключовим полем**. У ключовому полі задається основний (первинний) ключ запису. У якості первинного ключа можна використовувати будь-яку унікальну характеристику запису. Наприклад, номер запису або унікальну послідовність цифр або символів.

Запити - це основний робочий інструмент бази даних. Вони є засобами для вибірки даних за заданим критерієм, обновлення і ведення бази даних. Найпоширеніший вид запитів - запити на вибірку. Результат такого запиту має вигляд звичайної таблиці. Відмінність полягає в тім, що таблиці запитів заповнюються не вручну, а формуються відповідно до заданих умов відбору з даних вихідних таблиць (джерел) або інших запитів. При обробці запиту MS Access переглядає джерела (таблиці, запити), у яких зберігаються фактичні дані, і відбирає з них всі записи, які відповідають вимогам запиту. При цьому із самими джерелами нічого не відбувається, у таблицю запиту передаються тільки копії цих записів. Самі результати запитів не зберігаються, зберігається тільки структура запиту. Тому запити це не звичайні, а віртуальні таблиці. У запитах, на відміну від звичайних таблиць, можна робити обчислення.

Форми - це самий складний по виконанню, часто використовуваний і функціонально багатий об'єкт бази даних. У завдання форм входять: зручне подання інформації на екрані, введення і редагування інформації, керування базою даних. Для різних завдань використовують різні типи форм. Часто форми мають вигляд бланків з полями для введення. Джерелами інформації для форм можуть служити таблиці або запити. MS Access зв'язує кожну форму (крім вільних форм) з її джерелом. Це дозволяє забезпечити взаємно однозначну відповідність даних у формі і її джерелі.

Звіт - це кінцевий документ, призначений для виведення інформації на друк, проведення підсумкових розрахунків і зручного перегляду інформації на екрані. Звіти бувають прості, з угрупованням і зведені звіти. **Простий звіт** швидко створюється за допомогою засобу **Звіт**, він містить всі дані із джерела інформації. Дизайн звіту можна змінювати в режимі редагування. **Звіти з угрупованням** і **зведені звіти** створюються в режимі майстра звітів. Перші дозволяють відбирати потрібні поля, групувати і сортувати дані. Другі додатково дозволяють узагальнювати дані декількох звітів.

Модуль - це програма мовою Visual Basic for Applications (VBA, Visual Basic для приложений), призначена для розширення можливостей бази даних. **Макрос** - це набір з однієї або декількох макрокоманд, що виконують деякі операції в певній послідовності. Макроси звичайно використовуються для автоматизації часто виконуваних завдань і дозволяють замінити послідовність дій одним натисканням кнопки. Основним компонентом макросу є **макрокоманда** - замкнута інструкція, що

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

самостійно або в комбінації з іншими макрокомандами, визначає виконувані в макросі дії. Макроси дозволяють додавати функціональні можливості у форми, звіти і елементи керування без необхідності написання окремого модуля.

Кількість таблиць у базі даних визначається кількістю об'єктів, інформацію про які потрібно в ній зберігати. Наприклад, інформація про студентів, замовлення, постачальників тощо. Кожен об'єкт інформації характеризується атрибутами - більше детальною інформацією. Наприклад, об'єкт студент має такі атрибути як анкетні дані, адреса, місце проживання, інформацію про батьків тощо. Атрибути задаються полями таблиці. Формально для роботи бази даних досить однієї таблиці. Але робота з однією великою таблицею незручна через перевантаженість інформацією. Реляційні бази даних дозволяють розбити всю інформацію на серію відносно невеликих (по кількості полів) таблиць і зв'язати ці таблиці між собою. Такий підхід забезпечує гнучкість роботи бази даних, зменшує час пошуку, перегляду і обробки даних (час відгуку).

При роботі з MS Access кожна створена таблиця повинна мати ключове поле. Для їх створення в MS Access є три типи первинних ключових полів: лічильник, простий ключ і складений ключ. Поле типу **лічильник** формує порядковий номер запису. Це найпростіше ключове поле. Воно завжди задається MS Access за замовчуванням. У якості **простого первинного ключа** може використовуватися будь-яке поле, що має унікальне значення (наприклад, номер рахунку, номер залікової книжки, інвентарний номер). **Складений ключ** складається з декількох полів. Він використовується тоді, коли жодне з полів не є унікальним. Наприклад, адреса.

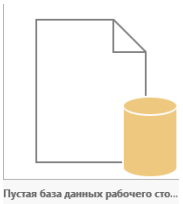
Зв'язки між таблицями встановлюють стосунки між співпадаючими значеннями в ключових полях. Їх установлюють після завдання ключових полів, але до заповнення таблиць. Як і в реляційній моделі в MS Access використовують наступні типи зв'язків:

зв’язок із відношенням "один - до - багатьох”. Кожному запису у таблиці 1 може відповідати декілька записів у таблиці 2, а запис у таблиці 2 не може мати більш одного відповідного йому запису в таблиці 1. Наприклад, кожна жінка може мати кілька дітей, але кожна дитина має тільки одну матір.

зв’язок із відношенням "багато - до - багатьох". Одному запису в таблиці 1 може відповідати декілька записів у таблиці 2, але одному запису в таблиці 2 теж може відповідати декілька записів у таблиці 1. Така схема реалізується тільки за допомогою третьої (допоміжної) таблиці. Наприклад, кожна фірма може виробляти багато товарів, але і кожен товар може вироблятися декількома фірмами.

зв’язок із відношенням "один - до - одного”. У цьому випадку запис у таблиці 1 може мати тільки один пов'язаний з ним запис у таблиці 2 і навпаки. Такий зв'язок найчастіше використовують для поділу широких таблиць.

Робота з СУБД MS Access 2016

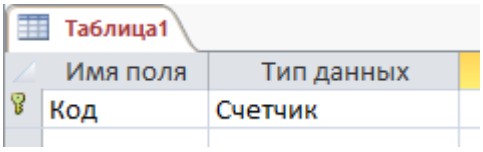


На відміну від інших додатків, MS Access при створенні бази даних відразу вимагає задати її ім'я і адресу. При подальшій роботі з базою даних всі зміни автоматично зберігаються. Існує два способи створити нову базу даних: у режимі конструктора і за допомогою майстра баз даних. **Майстер баз даних** пропонує ряд шаблонів. Це дуже зручно, тому

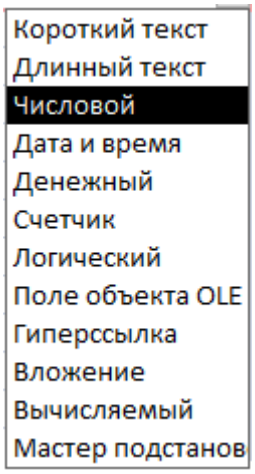
©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

що бази даних у цьому випадку нормалізовані і гарно оформлені. З іншої сторони рідко вдається підібрати потрібний шаблон. **Конструктор** дозволяє побудувати базу даних вручну за своїм бажанням. Цей спосіб кращий.

Для **створення таблиць** використовується вкладка **Створення** група **Таблиці**. Створювати таблиці можна трьома способами: у режимі конструктора (рекомендується) та шляхом введення даних (може виникнути помилка при визначенні типу даних). При створенні таблиці в режимі конструктора вводяться імена всіх полів таблиці, вибирається тип даних для кожного поля, задаються властивості поля і при збереженні задається ім'я таблиці. До імен полів пред'являються наступні вимоги: довжина імені не більше 64 символів, не можна починати ім'я із пробілу, не можна використовувати символи крапка, знак оклику, квадратні дужки, у межах таблиці імена полів повинні бути унікальними.



Оскільки кожна таблиця повинна містити первинний ключ, MS Access за замовчуванням створює його під ім'ям **Код** і задає тип ключового поля **Лічильник**. Для створення ключового поля іншого типу (простий ключ або складений ключ) потрібно спочатку видалити поле типу **Лічильник**, а потім створити нове ключове поле. Зробити це можна тільки в режимі конструктора.



На відміну від інших пакетів, MS Access призначений для зберігання і обробки великих масивів інформації. Тому дуже важливим стає обсяг займаною інформацією пам'яті і способи обробки даних. Ці характеристики залежать від типу даних. **Тип даних** для кожного поля вибирається залежно від того, яка інформація буде в ньому зберігатися. Кожен тип даних має свої параметри форматування (властивості поля). В MS Access існує 10 типів даних. Задаються вони в режимі конструктора таблиць.

Короткий текст використовується для вводу буквено-цифрових даних (імена, заголовки тощо). Розмір до 255 символів.

Довгий текст (колишня назва Memo), призначений для великих обсяги буквено-цифрових даних (речення та абзаци). Наприклад, опис товару. До 1 ГБ, але елементи керування, у яких може відображатися довгий текст, вміщуватимуть лише перші 64 000 символів.

Число. Числові дані. Розмір 1, 2, 4, 8 або 16 байт.

Дата й час. Дати й години. Розмір 8 байтів.

Грошовий. Грошові дані з 4 знаками після коми. Займає 8 байтів.

Лічильник. Унікальне значення, яке програма Access створює для кожного нового запису.Розмір4 байти (16 байт для ідентифікатора реплікації).

Логічний. Логічні значення ("Так" або "Ні"). В Access числове значення –1 відповідає значенню "Так", а 0 – "Ні". Займає байт.

Об’єкт OLE. Зображення, графічні об’єкти або інші об’єкти ActiveX з інших програм на платформі Windows. Розмір до 2 ГБ.

Перелік лекцій

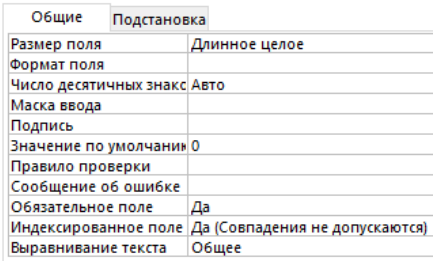
©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Гіперпосилання. Адреса посилання на документ або файл в Інтернеті, інтрамережі, локальній мережі або на локальному комп'ютері. До 8192 символів (кожна частина типу даних "Гіперпосилання" може містити до 2048 символів).

Вкладення. Ви можете вкладати зображення, документи, електронні таблиці або діаграми та інші файли. Кожне поле типу "Вкладення" може містити довільну кількість вкладень на запис, яка обмежується лише граничним розміром файлу бази даних. До 2 ГБ.

Обчислюваний. Ви можете створити вираз, який використовує дані з одного або кількох полів. На основі виразу можна призначити різні типи даних результату.

Майстер підстановок. Елемент **Майстер підстановок**, доступний у стовпці "Тип даних" у режимі конструктора, – це не тип даних. Якщо його вибрати, майстер допоможе вам визначити просте або складне поле підстановки. Просте поле підстановки використовує вміст іншої таблиці або список значень, щоб перевірити вміст одного значення в кожному рядку. У складному полі підстановки можна зберігати кілька значень одного типу даних у кожному рядку. Розмір залежить від типу даних поля підстановки.



Властивості полів визначають спосіб зберігання даних, їхнього редагування і відображення на екрані. Властивостей поля багато і задати їх можна в нижній частині вікна конструктора таблиць. Властивість **розмір поля** установлюється тільки для текстових і числових полів і для текстових полів характеризує максимальну кількість символів, а для числових полів відводить пам'ять, що потрібна для їх збереження. Для текстових полів потрібно пам'ятати, що один символ займає 1 байт пам'яті. Для числових полів:

формат байт використовується для зберігання цілих чисел у діапазоні від 0 до 255 і займає 1 байт пам'яті, **формат ціле** використовується для зберігання цілих чисел у діапазоні від -32 768 до +32 767 і займає 2 байти пам'яті, **формат довге ціле** використовується для зберігання цілих чисел від -2,1 до 2,1 мільярди і займає 4 байти пам'яті;

формат одинарне із плаваючою крапкою використовується для зберігання дробових чисел від -3,4E38 до +3,4E38. Мантиса містить 6 знаків. Займає 4 байти пам'яті;

формат подвійне із плаваючою крапкою використовується для зберігання дробових чисел від -1,8E308 до +1,8E308. Мантиса містить 10 знаків. Займає 8 байт пам'яті;

формат дійсне використовують для зберігання чисел від -1028 до +1028 з 28 значущими цифрами. Займає 12 байт пам'яті;

формат код реплікації застосовують при процедурі реплікації бази даних, тобто створенні декількох її копій. Займає 16 байт пам'яті.

Для полів типу числовий, грошовий, дата | час і логічний існує припустимий список форматів відображення даних. Формат **Грошовий** включає тільки два види

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

грошових одиниць. Для того, щоб при грошовому форматі замість рублів відображалися гривні, потрібно переналагодити операційну систему (**Панель керування - Мова і регіональні стандарти -** у **Регіональних параметрах** вибрати **Український)**.

Властивість **число десяткових знаків** установлюється для полів типу числовий і грошовий. Можна встановити будь-яку кількість цифр після коми (до 15 знаків). У цьому випадку потрібно задати у властивості **Формат поля** формат **Фіксований**. Опція **Авто** використовується для відображення тільки значущих цифр після коми. У цьому випадку потрібно задати у властивості **Формат поля** формат **Звичайний**.

Маски введення встановлюються тільки для полів типу текстовий або дата | час. Для текстових полів маски введення використовуються для введення паролів. Маски встановлюють зразок введення даних у поле, що дозволяє зменшити ймовірність помилки введення. Установлюється маска за допомогою майстра **Створення масок введення**. На першому кроці вибирається вид маски. Кнопка "Список" дозволяє доповнити стандартні маски користувальницькими. На другому кроці можна змінити вид заповнювача в масці.

Властивість **Підпис** використовується, якщо у звіті або формі потрібно вказати не ім'я поля, а інше позначення.

Властивість **Значення за замовчуванням** означає, що дані, уведені в поле з такою властивістю, будуть з'являтися автоматично в кожному записі. Їх можна потім виправляти. Цей формат використовується у випадку, коли в полі завжди (або майже завжди) зберігається те ж саме значення.

Властивість **Умова на значення** дозволяє задати обмеження на значення при веденні даних. Це допомагає здійснювати контроль правильності введення.

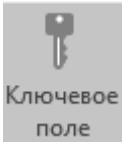
Властивість **Повідомлення про помилку** задає текст повідомлення, що буде видано при порушенні умови на значення в зазначеному полі.

Властивості **Значення за замовчуванням** і **Умова на значення** можна формувати за допомогою **Будівника виразу**.

У властивості **Обов'язкове поле** дозволяється вибір із двох значень - Так і Ні. Обов'язкові поля повинні бути заповнені даними.

Властивість **Індексоване** привласнюється тим полям, по яких часто здійснюється пошук або сортування даних. Індексування полів прискорює ці процеси, але сповільнює відновлення бази даних. Можливі два варіанти індексації: дозволити або не дозволити введення повторюваних значень.

Призначення ключового поля здійснюється за замовчуванням або в режимі конструктора. За замовчуванням встановлюється тільки поле типу лічильник. Для зміни ключового поля в режимі конструктора потрібно видалити поле типу лічильник, виділивши його клацанням по маркеру ліворуч від імені поля, і клацнути по кнопці **Видалити рядки**. Потім виділити поле, яке потрібно зробити ключовим, і клацнути по кнопці **Ключове поле**. Ліворуч від імені поля з'явиться символ ключа.



Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Редагування структури об'єктів бази даних виконується в режимі конструктора. Редагування вихідних даних виконується в режимі таблиці. Редагування структури таблиць включає зміну імен, типів і властивостей полів, їхнє додавання і видалення, зміну ключового поля. Щоб відкрити таблицю в режимі конструктора потрібно спочатку відкрити її в прямому режимі подвійним клацанням по її імені в області **Перехід**. Потім виконати команду **Головна - Режими - Режим - Конструктор** або відкрити контекстне меню і виконати команду **Конструктор**. Для корегування полів досить просто внести зміни. Для вставки поля потрібно виділити поле перед місцем вставки і клацнути по кнопці **Вставити рядки** із групи **Сервіс**. Для видалення поля потрібно виділити його і натиснути **Delete** або клацнути по кнопці **Видалити рядки** із групи **Сервіс**.

Зв'язки між таблицями встановлюються по ключових полях до заповнення таблиць і дозволяють працювати з декількома таблицями як з однією таблицею. Для встановлення зв'язків в MS Access крім первинних ключів використовуються також зовнішні ключі. **Зовнішній ключ** це поле (або поля), що містить посилання на первинний ключ в іншій таблиці. Зазвичай з первинним ключем однієї таблиці зв'язується зовнішній ключ іншої таблиці. Назви ключових полів можуть бути різними, але типи даних і зміст зовнішнього і первинного ключа повинні збігатися. Якщо дві таблиці зв'язуються відношенням один до багатьох, то таблиця на стороні один називається **головною** таблицею, а друга таблиця називається **підлеглою** таблицею. Головна таблиця обов'язково повинна зв'язуватися з підлеглою таблицею через свій первинний ключ. Наприклад, у базі даних є дві таблиці з інформацією про замовників: **Замовники** і **Договори**. Зв'язок між таблицями встановлюється по загальному полю **Замовник**. Для таблиці **Замовники** це первинний ключ, для таблиці **Договір** - зовнішній. У цьому зв'язку таблиця **Замовники** головна, а таблиця **Договори** - підлегла.

При створенні зв'язків між таблицями важливо забезпечити цілісність даних. **Цілісність даних** це система правил, використовуваних MS Access для підтримки зв'язків між записами в таблицях і забезпечення захисту від випадкового видалення або зміни зв'язаних даних. Якщо користувач не хоче включати забезпечення цілісності даних, то він бере на себе відповідальність за можливу невідповідність зв'язаних даних. Установити цілісність даних можна при виконанні ряду умов: у головній таблиці зв'язане поле повинне бути первинним ключем, зв'язані поля повинні мати один тип даних (виключення - поле лічильника можна зв'язувати із числовим полем із властивістю довге ціле), обидві таблиці повинні належати одній базі даних. У загальному випадку можна зв'язувати таблиці з різних баз MS Access. Автоматичне забезпечення цілісності даних накладає обмеження на можливість зміни зв'язаних даних. Для подолання цих обмежень, тобто для забезпечення зміни зв'язаних даних, не порушуючи їхньої цілісності, існують дві можливості:

каскадне відновлення зв'язаних полів. Якщо дозволити цю можливість, то при зміні значення в ключовому полі головної таблиці, автоматично змінюються і відповідні значення зв'язаних записів у підлеглій таблиці (таблицях). Змінювати значення зв'язаних записів у підлеглій таблиці можна тільки в межах значень із головної таблиці;

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

каскадне видалення зв'язаних записів. Дозволяє при видаленні запису в головній таблиці автоматично видаляти і всі зв'язані записи в підлеглих таблицях. Видалення записів у підлеглій таблиці не змінює головну таблицю.

Зв'язки можна встановлювати між таблицями і запитами. Перед створенням зв'язку потрібно **закрити всі об'єкти бази даних**. Установлюються зв'язки у вікні **Схема даних**. Відкривається це вікно у вкладці **Робота з базами даних** у групі **Відношення** клацанням по кнопці **Схема даних**.

Якщо зв'язки встановлюються вперше, вікно буде порожнім і автоматично відкриється діалогове вікно Додавання таблиці (рис. 4.1), у якому відображається перелік всіх таблиць і запитів бази даних. Якщо вікно не відкрилося, його можна викликати кнопкою **Відобразити таблицю** у групі **Зв'язки**. Необхідні таблиці або запити вибираються кнопкою **Добавити**, після чого це вікно необхідно закрити.

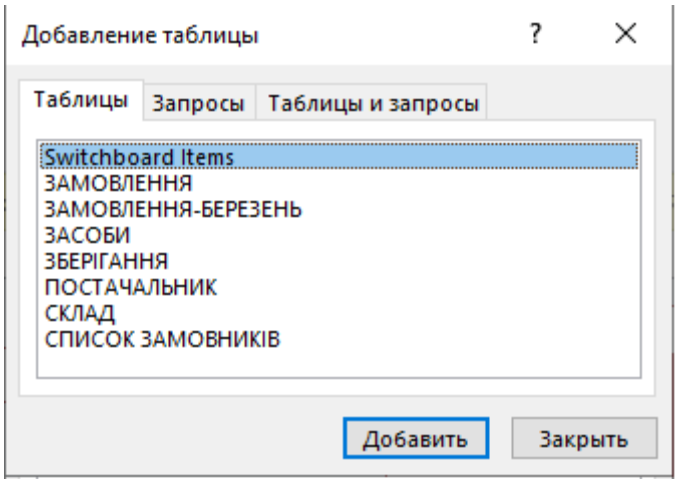


Рис. 4.1 Вікно Додавання таблиці

Для встановлення зв'язку потрібно вибрати поле в одній таблиці (запиті) і перетягнути його на відповідне поле в другій таблиці (запиті). Для установки декількох зв'язків потрібно перед перетаскуванням потрібно натиснути клавішу **Ctrl**.

У вікні **Зміна зв'язку** задати забезпечення цілісності даних, поставивши прапорці **Забезпечення цілісності даних**, **Каскадне відновлення зв'язаних полів** і **Каскадне видалення зв'язаних записів**. Натиснути клавішу **Створити**. Якщо зв'язок установлюється по первинних ключах обох таблиць або у обох таблицях загальні поля мають значення властивості **Індексоване** - **Так (Збіги не допускаються**), то зв'язок має тип один до одного. Якщо в головній таблиці це первинний ключ, а у зв'язаній таблиці використовується зовнішній ключ, то встановлюється зв'язок один до багатьох. Для зміни зв'язку потрібно зробити подвійне клацання по лінії цього зв'язку у вікні **Схема даних**. У вікні **Зміна зв'язків** задати новий зв'язок. Перегляд і видалення зв'язків здійснюються також у вікні **Схема даних**. Для перегляду всіх зв'язків бази даних потрібно натиснути кнопку **Всі зв'язки** в групі **Зв'язки**. Для перегляду зв'язків певної таблиці потрібно виділити цю таблицю і

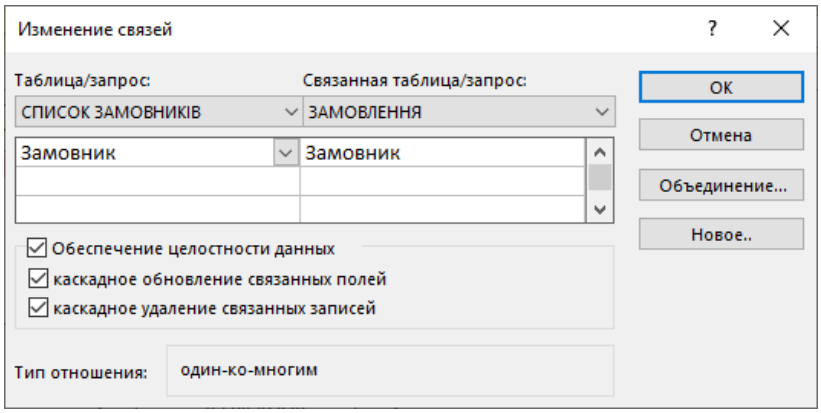


Рис. 4.2 Вікно Зміна зв'язку

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

натиснути кнопку **Прямі зв'язки**. Для видалення зв'язку досить клацанням виділити її лінію у вікні **Схема даних** і натиснути **Delete**. Знаки (+ | -) у головній таблиці дозволяють відкрити (закрити) для перегляду і редагування підлеглу таблицю для відповідного запису.

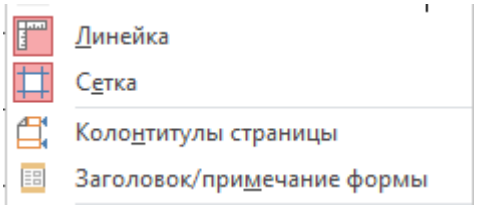
Форми призначені для зручного введення інформації в таблиці (форми введення), редагування і перегляду даних (переглядові форми), а також для керування базою даних (кнопкові форми). На будь-яку форму можна поміщати елементи керування і створювати спеціальні поля для проведення розрахунків. В MS Access кнопкові форми часто заміняються **Областю переходів**. Всі відкриті об'єкти відображаються в одному вікні. Для перемикання між ними служать вкладки. Часто функції форм введення, переглядових і кнопкових форм сполучаються в одній формі. Переглядові форми і форми уведення завжди мають один або кілька джерел інформації, якими можуть служити таблиці або запити. Якщо джерел інформації декілька, вони повинні бути зв'язані між собою. Всі зміни, внесені у форму, автоматично переносяться в пов'язані з нею таблиці або запити і навпаки. Крім звичайних форм, можна створювати порожні форми, на які потім наносяться елементи керування і інші об'єкти, наприклад малюнки. Порожні форми не мають джерела інформації. Кожна форма незалежно від її призначення має наступну структуру:

заголовок форми використовується для відображення назви форми, інструкцій з роботи з нею, розміщення елементів керування;

верхній і нижній колонтитули використовується для відображення інформації, що повинна повторюватися, на кожному аркуші форми вгорі або внизу відповідно;

область даних використовується для відображення записів і розміщення елементів керування;

примітка форми використовується для відображення інструкцій з роботи з формою і елементів керування.



За замовчуванням на екран виводиться тільки область даних. Відобразити на екрані інші області можна за допомогою контекстного меню командами **Колонтитули сторінки** і **Заголовок / Примітка форми**. Для зручності редагування на всіх областях форми командою **Сітка** можна відображати координатну сітку.

В MS Access 2016 існує багато способів створення форм: за допомогою інструментів (Форма, Розділена форма, Кілька елементів, Порожня форма), майстрів форм, у режимі конструктора і за допомогою диспетчера кнопкових форм. **Інструменти** використовуються для створення простих форм, **Майстер форм** - для створення гарних форм введення і переглядових форм. Створену форму можна потім поліпшити в режимі конструктора. Для створення кнопкових форм використовують диспетчер кнопкових форм.

Для створення форми за допомогою інструмента **Форма** потрібно виділити джерело даних для форми і виконати команду **Створення – Форма** (група **Форми**). При цьому у формы містяться всі поля джерела даних і відображається тільки один

Перелік лекцій

запис. Форму можна змінювати в режимі макета або конструктора. Для головної таблиці створюється складна форма, для підлеглої або незв'язаної таблиці - проста.

Для створення форми за допомогою інструмента **Розділена форма** потрібно виділити джерело даних для форми і виконати команду **Створення – Інші форми – Розділена форма**. У створеній формі одночасно відображаються дані в режимі форми і у режимі таблиці. У режимі форми відображається один запис, у режимі таблиці - декілька. При виділенні поля в одній частині форми, виділяється таке ж саме поле в іншій частині. Дані можна додавати, змінювати або видаляти в кожній частині форми. Можна скористатися нижньою табличною частиною форми, щоб швидко знайти запис, а потім переглянути або змінити запис у верхній частині форми.

Для створення форми за допомогою інструмента **Кілька елементів** потрібно виділити джерело даних для форми і виконати команду **Створення – Інші форми – Кілька елементів**. Цей інструмент використовується для відображення кількох записів одночасно. Форма зовні нагадує таблицю, але, як будь- яку форму, її можна набудовувати. Наприклад, прикрашати, вирівнювати, додавати малюнки, емблеми, діаграми і елементи керування.

Для створення форми за допомогою інструмента **Порожня форма** потрібно виконати команду **Створення - Порожня форма**. Цей інструмент дозволяє створити форму з обраними полями. В області **Список полів** вибираються поля для форми і подвійним клацанням або перетаскуванням містяться у форму.

Інструменти **Форма**, **Розділена форма** і **Кілька елементів** не дозволяють вибирати поля для форми.

Майстер форм забезпечує можливість вибору полів форми і зміни їхньої послідовності. Можна включити у форму поля з декількох зв'язаних таблиць або запитів. Майстер пропонує ряд шаблонів оформлення форми. Викликається майстер форм командою **Створення - Майстер форм.** У режимі діалогу майстер форм пропонує користувачеві вибрати джерело даних і поля, вибрати вид форми, стиль і задати ім'я форми. Майстер форм дозволяє створювати форми декількох видів:

в один стовпець - кожен запис розташовується на окремій сторінці, кожне поле - в окремому рядку. Для перегляду даних треба перегорнути сторінки форми за допомогою лічильника. Форми в один стовпець зазвичай використовуються для введення даних;

стрічкова - на сторінці міститься кілька записів, один під іншим. Стрічкові форми зазвичай використовуються для перегляду і редагування даних;

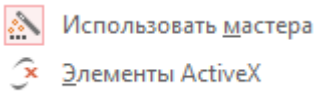
таблична - має вигляд звичайної таблиці. Можна задавати лінії сітки по горизонталі або вертикалі, тип ліній і колір ліній, тло і інші параметри. Зазвичай використовується для перегляду і редагування даних;

вирівняна - кожен запис розташовується на окремій сторінці, але поля розміщаються в лінію. Звичайно використовується для введення даних.

Недоліком майстра форм є те, що він обмежує можливості дизайну форми і не дозволяє розміщувати на ній елементи керування. Тому побудована за шаблоном форма часто вимагає доробки і редагується в режимі **макета** або **конструктора**. У цих

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

режимах можна змінювати розміри об'єктів, їхнє розташування, кольорову гаму, шрифти і т.д. Виконується редагування командою **Властивості** контекстного меню або за допомогою кнопок на панелі інструментів. Треба пам'ятати, що при виклику контекстного меню для редагування конкретного елемента форми покажчик миші встановлюється на цей елемент, для редагування загальних властивостей форми (підпису, розділових ліній, смуг прокручування і інше) покажчик миші встановлюється на вільне місце форми.



Елементи керування базою даних можна розміщати тільки на формах і звітах, найчастіше вони встановлюються на формах. Створюються елементи керування тільки в режимі конструктора форми (звіту) за допомогою майстрів. За замовчуванням програми майстри активовані. Але якщо при створенні елемента керування не відкривається діалогове вікно майстра, потрібно активувати його вручну. Для цього досить клацнути по елементу **Використовувати Майстри** (рис. 4.3) у списку елементів.

Для встановлення елементів керування потрібно відкрити форму в режимі конструктора і у вкладці **Конструктор** у групі **Елементи керування** вибрати потрібний елемент. В MS Access 2016 можна встановлювати наступні елементи керування.

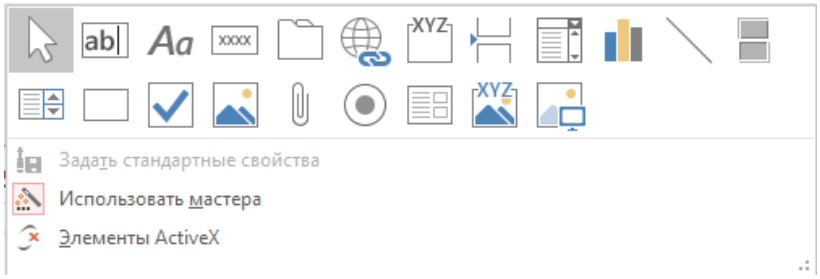


Рис. 4.3 Елементи керування для форм

Поле - додає у форму текстове поле. Це поле можна зв'язати з полем таблиці.

Напис - для створення підпису до інших елементів форми.

Кнопка - служить для виконання певних дій (відкрити або закрити об'єкт, переміститися по записах, викликати інший додаток і т.д.) Ці дії можуть задаватися макросами.

Список - формує розкритий список. Для перегляду великого списку використовуються смуги прокручування.

Поле зі списком - формує поле уведення зі списком, що розкривається.

Підлегла форма/звіт - створює підлеглу форму або звіт.

Лінія - малює лінію.

Прямокутник - малює прямокутник. Для лінії і прямокутника за допомогою палітри, розташованої в тій самій групі, задаються товщина і вигляд ліній.

Вільна і приєднана рамки об'єкта - формують рамку для впровадженого або убудованого об'єкта відповідно.

Прапорець - кнопка для вибору одного з параметрів, що не виключають один одного.

Перемикач - кнопка для вибору одного із взаємовиключних параметрів.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Група перемикачів - кнопки для групи перемикачів, з яких включеним може бути тільки один.

Вимикач - кнопка, що перебуває тільки у двох станах: включений - виключений.

Вкладка - створює форму із двома вкладками.

Додати вкладку - додає у форму нові вкладки.

Вставити діаграму або малюнок.

Розрив сторінки - формує багатосторінкові форми.

Вставити гіперпосилання - вставляє посилання на еЬ сторінку, електронну пошту, програму, файл.

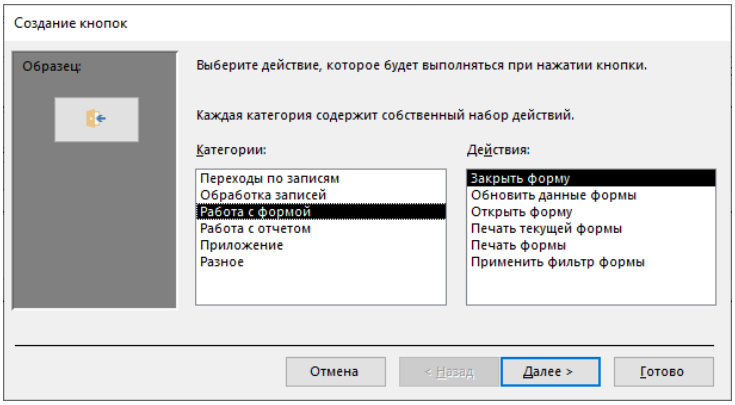
Вкладення - вставляє зовнішній об'єкт.

Вибір об'єкта - виділяє окремі об'єкти на формі.

Виділити все - виділяє всі об'єкти на формі.

Використати майстри - активізує програми майстра для створення елементів керування.

Розглянемо роботу майстра для створення елементів керування на прикладі створення кнопки. Для створення кнопки потрібно клацнути по елементу **Кнопка** на панелі елементів, потім клацнути по тому місцю форми, де повинна бути розміщена ця кнопка.



З'являється діалогове вікно **Створення кнопок**. Тут вибирається одна з шести можливих категорій кнопок: переходи по записах, обробка записів, робота з формою, робота зі звітом, додаток і різне. Кожній категорії відповідає набір можливих дій і малюнок на кнопці. Оскільки одним з найпоширеніших завдань керування є передача керування іншим формам, виберемо категорію **Робота з формою**. У ній передбачені дії по відкриттю або закриттю форми, роботі з фільтрами для відбору інформації, друкуванню форми і відновленню даних. Виберемо відкриття форми. У наступному вікні з'являється список всіх наявних у базі форм. Вибираємо форму для відкриття. Далі майстер кнопок пропонує створити фільтр для відображуваних даних. Є дві можливості: відображати всі записи (нижній прапорець) або відображати тільки відібрані записи (верхній прапорець). Якщо задати режим відображення відібраних записів з'явиться вікно, у якому потрібно задати умову відбору. У віконці ліворуч відображені всі поля форми, на яку ставиться кнопка. У віконці праворуч - всі поля форми що відкривається. Потрібно виділити необхідне поле праворуч і ліворуч і з'єднати їх за допомогою кнопки **<->**. Потім вибирається назва кнопки. Можна на кнопці відобразити малюнок, але для кнопок відкриття рекомендується писати на кнопці ім'я форми що відкривається. На наступному кроці майстер кнопок пропонує задати ім'я кнопки. На формі воно не відображається, його

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

можна побачити в режимі конструктора за командою **Властивості**. Зрозуміле ім'я полегшує подальші посилання на кнопку. Якщо ім'я не задавати, то MS Access просто пронумерує цю кнопку. Недосвідченим користувачам краще не давати ім'я кнопкам, тому що при редагуванні кнопки, пов'язаному з її видаленням і створенням заново, MS Access щоразу створює нову підпрограму на мові Visual Basic for Applications і дає їй ім'я кнопки. Наявність декількох підпрограм з одним ім'ям приводить до помилки. Створення кнопки без установлення фільтра на відбір записів здійснюється аналогічно, у цьому випадку кроки, пов'язані із установленням фільтра відсутні і відразу пропонується вибрати напис на кнопці. Рекомендується на кнопці, що пов'язана з роботою з певним об'єктом, писати ім'я об'єкта або ім'я дії. Якщо кнопка викликає стандартну операцію (закрити, обновити тощо) краще використати стандартний малюнок. Редагування оформлення і інших властивостей кнопки виконується в режимі конструктора командою **Властивості** контекстного меню. Можна змінити ім'я кнопки, її розміщення і розмір, колір, шрифт і накреслення тексту на кнопці. Аналогічно створюються і редагуються інші елементи керування: вмикачі, перемикачі, прапорці тощо.

Поля у формах і звітах MS Access бувають трьох типів: зв'язані, вільні і що обчислюють. **Зв'язане поле** містить посилання на якесь поле таблиці або запиту. **Вільне поле** не зв'язане з жодним джерелом інформації. **Поле, що обчислює**, містить формули і використовується для розрахунків. Додаються поля всіх видів у режимі конструктора. Особливість створення нового поля полягає в тому, що при додаванні поля фактично додається два взаємозалежних елементи: напис і елемент керування. Додавати зв'язані поля можна тільки з таблиці (таблиць), що є джерелом інформації для форми (звіту) або пов'язаних з нею таблиць. Для додавання нового зв'язаного поля потрібно відкрити форму (звіт) у режимі конструктора і виконати команду **Додати поля** із групи **Сервіс**. У верхній області екрана відобразиться список всіх полів таблиці - джерела і пов'язаних з нею таблиць. З цього списку подвійним клацанням потрібно вибрати поле для вставки. У нижній області екрана відображаються поля незв'язаних таблиць. Для їхньої вставки можна прямо в процесі вставлення створити зв'язок. Якщо зв'язок установити не можна, то поле теж не можна вставити. Для зв'язаних полів додається два взаємозалежних елементи: напис і елемент керування - поле. **Напис** задає найменування поля так, як це поле називається в джерелі інформації. **Елемент керування - поле** містить посилання на це поле. Напис можна змінювати, не міняючи змісту поля. Для цього потрібно клацнути по напису і написати нову назву, наприклад, "Виробник" замість "Постачальник". Зміна вмісту поля приводить до зміни його значення в таблиці. Для додавання нового **вільного поля** використовується елемент **Поле** панелі елементів. При цьому створюється напис **Поле №** та елемент управління **Вільний**. Напис змінюють на потрібний напис. У вільне поле вводяться дані. **Поля, що обчислюють**, створюються у вільних полях, у які вводиться формула. Існують наступні правила написання формул:

у формулах можна використовувати числа, імена полів, стандартні математичні операції + (плюс), - (мінус),* (помножити), / (поділити) і круглі дужки;

формули починаються зі знака рівності =;

імена полів поміщують у квадратні дужки;

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

для складних формул використовують будівник виражень, що викликається командою **Властивості** контекстного меню;

якщо формула не поміщується у полі, потрібно нажати "гарячі" клавіші **Shift + F2** і в області, що відкрилася, ввести формулу.

Поля, що обчислюють, редагуються і форматуються так само, як і звичайні поля, командами контекстного меню або командою **Властивості** цього меню. Для завдання кількості знаків після коми в полях числового типу, використовується формат поля **Фіксований**.

Для пошуку даних в MS Access використовуються команди **Знайти** і **Замінити**, засоби обробки даних **Сортування** і **Фільтри**, і об'єкт бази даних **Запити**. Всі засоби пошуку, крім запитів, використовуються в режимах таблиці або форми. Запити створюються в режимі конструктора або за допомогою майстра запитів. Команди **Знайти** і **Замінити** працюють у покроковому режимі і звичайно застосовуються як допоміжні команди перед використанням фільтрів.

Команди **Знайти** і **Замінити** використовуються для пошуку і заміни певного запису і знаходяться у вкладці **Головна**, групі **Знайти**. Для пошуку у вікні **Пошук і заміна** вибирається вкладка пошук. У поле **Зразок** вводиться текст що шукається, у полі **Пошук з** вибирається зі списку таблиця. У полі **Збіг** вибирається зі списку потрібний шаблон пошуку. В MS Access підтримується три шаблони:

шаблон **поля цілком** знаходить записи, у яких перебуває тільки зазначений текст. Наприклад, якщо в зразку зазначене **Іванов**, то буде шукатися тільки це прізвище;

шаблон **с початку поля** шукає записи, що починаються із зазначеного тексту. Наприклад, результатом пошуку за цим шаблоном буде **Іванов**, І**ванов**а, **Іванов**ський, **Іванов**ич, але не буде знайдено Сел**иванов**ский;

шаблон **с будь-якою частиною поля** знаходить всі записи, в які включений текст. Наприклад, у цьому випадку в результат пошуку потрапить слово Сел**иванов**ский теж.

Режими перегляду при пошуку задаються в поле **Перегляд**: режим **Усе** означає переглядати всі записи, режим **Нагору** означає переглядати записи нагору від поточного запису і режим **Униз** означає переглядати записи униз від поточного запису. Прапорець **Із урахуванням регістра** дозволяє враховувати розходження між прописними і заголовними буквами. Для початку пошуку натискається клавіша **Знайти далі**. Команда **Знайти** знаходить записи по черзі, для продовження пошуку потрібно натискати клавішу **Знайти далі** доти, поки не з'явиться повідомлення про кінець пошуку. Заміна запису виконується аналогічно пошуку. Відмінність у тім, що вибирається вкладка **Заміна** у вікні пошуку і у поле **Замінити на** вводиться текст заміни. Для заміни використовуються клавіші **Замінити** або **Замінити все**.

MS Access за замовчуванням автоматично сортує записи по ключовому полю за зростанням. Для зміни сортування потрібно в режимі таблиці або форми помістити курсор у потрібне поле і виконати наступні дії - вкладка **Головна -** група **Сортування і фільтр** та скористатися кнопками сортування. Обмежень на кількість полів для сортування немає. Якщо сортування встановлюється по декількох полях одночасно,

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

то пріоритети сортування задаються зліва направо. Для скасування сортування використовується кнопка **очистити всі сортування**.

Фільтри використовуються для перегляду підмножини обраних даних. На відміну від команд **Знайти** і **Замінити** вони дозволяють побачити всі дані, що цікавлять, одночасно. Фільтри можна застосовувати до таблиці, запиту або форми. В MS Access існує три методи фільтрації: **фільтр за виділеним фрагментом** (за виділенням), **звичайний фільтр** і **розширений фільтр**. Перші 2 методи простіші і частіше використовуються. Результати фільтрації в цих методах відображаються на місці вихідної таблиці. **Фільтр за виділеним фрагментом** використовується при наявності тільки одного критерію відбору. Його можна застосовувати для фільтрації по декількох критеріях, але для цього доводиться використати фільтр кілька разів підряд, накладаючи умови фільтрації, одну на другу. Для використання цього фільтра потрібно: відкрити таблицю (форму, запит) у режимі таблиці; знайти в таблиці будь-який запис, у якому є значення, що служить критерієм відбору (це можна зробити командою **Знайти**) і задати умову (критерій) відбору. Є три режими відбору даних у фільтрі за виділеним фрагментом:

поставити курсор в елемент поля - відбираються записи, у яких значення в полі фільтрації збігаються з виділеним значенням;

виділити частину поля, починаючи з першого символу - відбираються записи, у яких значення в поле фільтрації починаються з виділеного значення;

виділити будь-яку частину поля, що не включає перший символ - відбираються записи, у яких значення в поле фільтрації включають виділене значення.

Для встановлення фільтра за виділенням є два способи. У першому способі клацають по кнопці **Виділення** в групі **Сортування і фільтр** вкладки **Головна**. Для текстових полів можна відбирати: **записи, що збігаються**, або **не співпадаючі з виділеним значенням**; **записи що утримують** або **не утримують** його **як свою частину**. Для числових полів можна відбирати записи **рівні** або **не рівні** виділеному значенню, **менші або рівні** або **більші або рівні** його, і **перебувають у заданому діапазоні.** У другому способі клацають по кнопці **Фільтр** тієї ж групи. У вікні, що з'явилося, можна сортувати дані і установлювати фільтр. Для текстових полів в області ліворуч ставляться індикатори у виділюваних значеннях. У правій області можна відбирати записи, що **збігаються** або **не співпадаючі** з виділеними значеннями, що **починаються** або **не починаються** з виділеного фрагмента, що **містять** або **не містять** його як свою частину і що **закінчуються** або **не закінчуються** на фрагменті. Для числових полів можливості не змінюються.

Звичайний фільтр використовується при завданні декількох умов відбору. Умови фільтрації можуть реалізовуватися за допомогою операції логічного множення (**AND**) або операції логічного додавання (**OR**). У першому випадку відбираються записи, що задовольняють всім умовам відбору одночасно, у другому випадку відбираються записи, що збігаються хоча б з однією із заданих умов відбору. Критерії відбору у звичайному фільтрі можуть бути досить складними і включати логічні вираження (більше, менше, дорівнює та інші). Для виклику вікна звичайного фільтра потрібно клацнути по кнопці **Змінити фільтр**, що розміщена під кнопкою **Додатково** в групі **Сортування і фільтр**. У бланку фільтра, що з'явився, клацнути по полю

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

фільтрації. У списку, що з'явився, вибрати критерій відбору або ввести свій критерій. Кількість полів фільтрації не обмежена. Для пошуку записів, що мають у поле фільтрації порожні або непусті значення, потрібно ввести в це поле вираження **Is Null** (для порожніх) або **Is Not Null** (для заповнених записів). Для реалізації умови "Або" потрібно клацнути по вкладці "Або". Таких умов може бути декілька. Для установки фільтра досить клацнути по кнопці **Застосувати фільтр**. Ця кнопка також перебуває в групі **Сортування і фільтрація.**

Розширений фільтр на додаток до можливостей звичайного фільтра дозволяє виконувати сортування по різних полях і використовувати більш складні умови відбору, створені за допомогою будівника виражень. Пріоритет сортувань - зліва направо. Щоб викликати розширений фільтр клацають по кнопці **Додатково \ Розширений фільтр**. На екрані відображається бланк фільтра. У верхній області бланка відображені всі поля таблиці. Нижня частина бланка повинна бути порожньою. Якщо там щось є, необхідно очистити цю частину бланка. Для завдання умов фільтрації і сортувань виконуються наступні дії: у вікні фільтра потрібно подвійним клацанням (або методом перетаскування) додати в порожній бланк фільтра (нижня частина) поле або поля фільтрації; у рядку **Сортування** потрібного поля вибрати порядок сортування; у рядку **Умова відбору** вказати значення або ввести вираження; нажати кнопку **Застосувати фільтр**.

При закритті таблиці будь-який встановлений фільтр автоматично зберігається. Для того, щоб не зіпсувати вихідну таблицю, перед її закриттям потрібно зняти фільтр. Для скасування фільтра робиться клацання по кнопці **Застосувати фільтр** у групі **Сортування і фільтр**. Для збереження результатів фільтрації перед скасуванням фільтра потрібно зберегти їх як запит. Для цього виконати команду **Файл – Зберегти як Зберегти об’єкт як**. У вікні Збереження задати ім'я об'єкта, що зберігається, і вибрати тип **запит**. Виконати збереження і створити новий об'єкт.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Питання для самоконтролю

1. Що таке реляційна база даних?
2. Яку структуру та розширення має файл бази даних, створений у Microsoft Access?
3. Для чого призначені таблиця, форма, запит, звіт?. Дайте визначення цих понять.
4. Якими способами можна створювати таблиці у базі даних Microsoft Access?
5. Які обмеження накладаються на імена полів таблиці?
6. Для чого використовується режим таблиці та режим конструктора? В чому між ними різниця?
7. Що таке нормалізація таблиць. Яка таблиця має назву нормалізованої?
8. Для чого призначено поле типу довгий текст та Вложення ?
9. Що таке первинній ключ?
10. Назвіть способи, за допомогою яких можливо відкрити схему даних?
11. Що повинно бути однаковим у полях, які зв'язуються?
12. Які типи зв'язків існують у реляційних базах даних?
13. Якими способами можна зв'язати таблиці у Microsoft Access?
14. Що таке підтримання цілісності даних, каскадне оновлення та каскадне видалення зв'язаний полів?
15. Що таке сортування записів? Що таке ключ сортування?
16. Скільки ключів сортування може бути у Access? Коли доцільно проводити сортування по другому та третьому ключу?
17. Як розміщуються числа, текст та дати, якщо встановлено характер сортування за зростанням? За спаданням?
18. Як переглянути записи за певним критерієм?
19. Як застосувати або вилучити фільтр? Як очистити бланк розширеного фільтру?
20. Які команди містить підменю команди Додатково?
21. Що таке форма? Які дії вона дозволяє виконувати?
22. Які види автоматичних форм можливо створити у Microsoft Access?
23. Чим відрізняються між собою форма у стовпець, стрічкова, таблична? Для чого вони призначені.
24. Якими способами можна створювати форму у Microsoft Access?
25. Як створити на формі поле, що обчислюється?
26. Як викликати на екран вікно побудови виразів?
27. Які елементи керування можна вставити на форму?
28. Що таке запит? Для чого він використовується?
29. Які види запитів існують?
30. У чому різниця між запитом та фільтром?
31. Що таке параметричний запит? Для чого він використовується?
32. Що таке перехресний запит? Для чого він використовується?
33. Які способи створення запитів існують?

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

WAN Internet.
Локальні та глобальні комп'ютерні мережі. Мережа Internet

Зміст

WAN Internet. Локальні та глобальні комп'ютерні мережі. Мережа Internet 83
1. *Загальні поняття комп'ютерних мереж.* 83
Лінії зв'язку і канали передачі даних 84
Комунікаційне обладнання комп'ютерних мереж 88
2. *Класифікація мереж* 90
3. *Протоколи, стеки протоколів, модель TCP/IP* 96
Протоколи, інтерфейси, стеки протоколів 96
Мережева модель TCP / IP 97
Формат адрес у стеці TCP / IP 100
4. *Глобальна мережа Інтернет* 105
Поняття провайдера 106
Способи доступу або підключення до Інтернет 107
Адресація в глобальній мережі Інтернет – сервіс DNS 109
Сервіси мережі інтернет 110
Питання для самоконтролю 113

1. *Загальні поняття комп'ютерних мереж.*

Мережі - це системи, що формуються сполуками. Наприклад, дороги, що з'єднують групи людей, формують фізичну мережу. Зв'язки між вами і вашими друзями формують вашу особисту мережу. Веб-сайти, які дозволяють окремим користувачам зв'язуватися зі сторінками один одного, називаються соціальними мережами.

Комп'ютерні мережі - це сукупність вузлів, об'єднаних каналами передачі даних, що забезпечують користувачів засобами обміну інформацією і колективного використання ресурсів мережі: апаратних, програмних та інформаційних.

Комп'ютерні мережі також називаються інформаційно - обчислювальними мережами, обчислювальними мережами або мережами передачі даних.

Комп'ютерні мережі (КМ) складаються з різних пристроїв. Деякі пристрої можуть служити або в якості вузлів, або в якості периферійних пристроїв. Вузол (хост) - це будь-який пристрій, що відправляє і отримує інформацію по мережі. Принтер, підключений до ноутбука, є периферійним пристроєм. Якщо ж принтер підключений до мережі безпосередньо, він функціонує як вузол.

Комп'ютерні мережі забезпечують:

- Розподіл даних (Data Sharing). Дані в КМ зберігаються на центральному вузлі (сервері) і можуть бути доступні на робочих станціях, тому на кожному робочому місці не треба мати накопичувачі для зберігання однієї і тієї ж інформації.
- Розподіл інформаційних і технічних ресурсів (Resource Sharing):
  - логічні диски та інші зовнішні пристрої для зберігання інформації (накопичувачі на CD-ROM, DVD, ZIP і так далі);
  - каталоги (папки) та файли, що в них містяться;

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

- підключені до ПК пристрої: принтери, модеми та інші зовнішні пристрої (що дозволяє економно використовувати ресурси).

– Розподіл програм (Software Sharing). Всі користувачі КМ можуть спільно мати доступ до програм (мережевим версіями), які централізовано встановлюються в мережі.

– Обмін повідомленнями по електронній пошті (Electronic Mail). Всі користувачі мережі можуть оперативно обмінюватися інформацією між собою за допомогою передачі повідомлень.

Весь комплекс програмно-апаратних засобів, що входять до складу мережі може бути описаний **багатошаровою моделлю**, що складається з шарів:

– комп'ютери або комп'ютерні платформи;
– комунікаційне обладнання;
– операційні системи;
– мережеві додатки.

Комп'ютери. В основі будь-якої мережі лежить апаратний шар стандартизованих комп'ютерних платформ. В даний час широко використовуються комп'ютерні платформи різних класів - від персональних комп'ютерів до мейнфреймів і суперЕОМ. Комп'ютери підключаються до мережі за допомогою мережевої карти.

Комунікаційне обладнання. До другого прошарку належить комунікаційне обладнання, яке відіграє не менш важливу роль, ніж комп'ютери.

Комунікаційне обладнання мереж можна розділити на три групи:

– середовище передачі даних;
– мережеві адаптери (карти);
– проміжне комунікаційне обладнання (трансивери, повторювачі, концентратори, комутатори, мости, маршрутизатори і шлюзи).

Операційні системи. Третім шаром, що створює програмну платформу мережі, є операційні системи. Залежно від того, які концепції управління локальними і розподіленими ресурсами покладені в основу мережевої ОС, залежить ефективність роботу всієї мережі.

Мережеві додатки. Четвертий шар - це мережеві додатки. До мережевих додатків відносяться такі додатки як мережеві бази даних, поштові програми, системи автоматизації колективної роботи тощо.

Лінії зв'язку і канали передачі даних

Для побудови комп'ютерних мереж застосовуються лінії зв'язку, що використовують різну фізичну середу. Як фізичне середовище в комунікаціях використовуються: метали (в основному мідь), надпрозоре скло (кварц) або пластик і ефір. Фізичне середовище передачі даних може являти собою кабель "вита пара", коаксіальні кабель, волоконно-оптичний кабель і навколишній простір.

Лінії зв'язку або **лінії передачі даних** - це проміжна апаратура і фізичне середовище, по якій передаються інформаційні сигнали (дані).

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

В одній лінії зв'язку можна утворити кілька каналів зв'язку (віртуальних або логічних каналів), наприклад шляхом частотного або тимчасового поділу каналів. **Канал зв'язку** - це засіб **односторонньої** передачі даних. Якщо лінія зв'язку монопольно використовується каналом зв'язку, то в цьому випадку лінію зв'язку називають каналом зв'язку.

Канал передачі даних - це засоби **двостороннього** обміну даними, які включають в себе лінії зв'язку та апаратуру передачі (прийому) даних. Канали передачі даних пов'язують між собою джерела інформації та приймачі інформації.

Залежно від фізичного середовища передачі даних лінії зв'язку можна розділити на:

– провідні лінії зв'язку без ізолюючих і екрануючих облаток;
– кабельні, де для передачі сигналів використовуються такі лінії зв'язку як кабелі "вита пара", коаксіальні кабелі або оптоволоконні кабелі;
– бездротові (радіоканали наземного і супутникового зв'язку), що використовують для передачі сигналів електромагнітні хвилі, які розповсюджуються по ефіру.

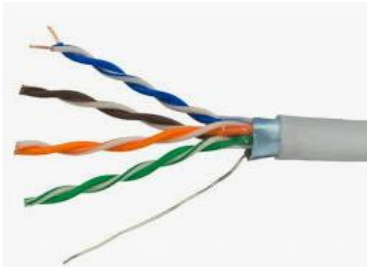
Провідні лінії зв'язку

Провідні (повітряні) лінії зв'язку використовуються для передачі телефонних і телеграфних сигналів, а також для передачі комп'ютерних даних. Ці лінії зв'язку застосовуються в якості магістральних ліній зв'язку.

По провідних лініях зв'язку можуть бути організовані аналогові і цифрові канали передачі даних. Швидкість передачі по провідних лініях "простої старої телефонної лінії" (POST - Primitive Old Telephone System) є дуже низькою. Крім того, до недоліків цих ліній відносяться перешкодозахищеність і можливість простого несанкціонованого підключення до мережі.

Кабельні лінії зв'язку

Кабельні лінії зв'язку мають досить складну структуру. Кабель складається з провідників, укладених в кілька шарів ізоляції. У комп'ютерних мережах використовуються три типи кабелів.



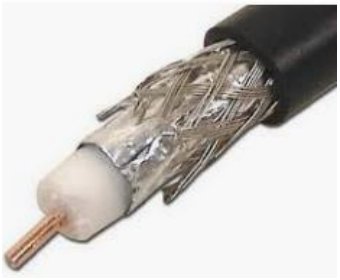
Вита пара (twisted pair) - кабель зв'язку, який представляє собою виту пару мідних проводів (або кілька пар проводів), укладених в екрановану оболонку. Пари проводів скручуються між собою з метою зменшення наведень. Вита пара є досить перешкодостійкою. Існує два типи цього кабелю: неекранована вита пара UTP і екранована вита пара STP.

Характерним для цього кабелю є простота монтажу. Даний кабель є найдешевшим і поширеним видом зв'язку, який знайшов широке застосування в найпоширеніших локальних мережах з архітектурою Ethernet, побудованих по топології типу "зірка". Кабель підключається до мережевих пристроїв за допомогою з'єднувача RJ45.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Кабель використовується для передачі даних на швидкості 10 Мбіт/с і 100 Мбіт/с. Вита пара зазвичай використовується для зв'язку на відстань не більше кількох сотень метрів. До недоліків кабелю "вита пара" можна віднести можливість простого несанкціонованого підключення до мережі.



Коаксіальний кабель (coaxial cable) - це кабель з центральним мідним дротом, який оточений шаром ізолюючого матеріалу для того, щоб відокремити центральний провідник від зовнішнього провідного екрану (мідної обплетення або шар алюмінієвої фольги). Зовнішній провідний екран кабелю покривається ізоляцією.

Існує два типи коаксіального кабелю: тонкий коаксіальний кабель діаметром 5 мм і товстий коаксіальний кабель діаметром 10 мм. У товстого коаксіального кабелю загасання менше, ніж у тонкого. Вартість коаксіальногокабелю перевищує номінальну вартість кручений пари і виконання монтажу мережі складніше, ніж кручений парою.

Коаксіальний кабель застосовується, наприклад, в локальних мережах з архітектурою Ethernet, побудованих по топології типу "загальна шина".

Коаксіальний кабель більш перешкодозахищений, ніж кручена пара і знижує власне випромінювання. Пропускна здатність - 50-100 Мбіт/с. Допустима довжина лінії зв'язку - кілька кілометрів. Несанкціоноване підключення до коаксіального кабелю складніше, ніж до кручений парі.

Кабельні оптоволоконні канали зв'язку. Оптоволоконний кабель (fiber optic) - це оптичне волокно на кремнієвій або пластмасовій основі, укладену в матеріал з низьким коефіцієнтом заломлення світла, який закритий зовнішньою оболонкою.

Оптичне волокно передає сигнали тільки в одному напрямку, тому кабель складається з двох волокон. На передавальному кінці оптоволоконного кабелю потрібно перетворення електричного сигналу в світловий, а на приймальному кінці зворотне перетворення.

Основна перевага цього типу кабелю - надзвичайно високий рівень перешкодозахищеності і відсутність випромінювання. Несанкціоноване підключення дуже складно. Швидкість передачі даних 3Гбіт/с. Основні недоліки оптоволоконного кабелю - це складність його монтажу, невелика механічна міцність і чутливість до іонізуючих випромінювань.

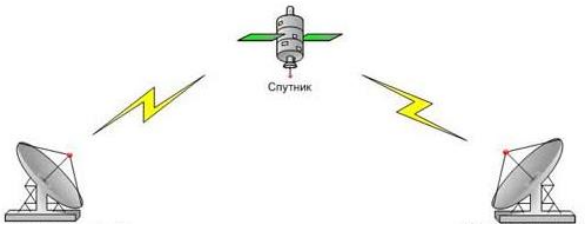
Бездротові (радіоканали наземного і супутникового зв'язку) канали передачі даних

Радіоканали наземного (радіорелейного і стільникового) і супутникового зв'язку утворюються за допомогою передавача і приймача радіохвиль і відносяться до технології бездротової передачі даних.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

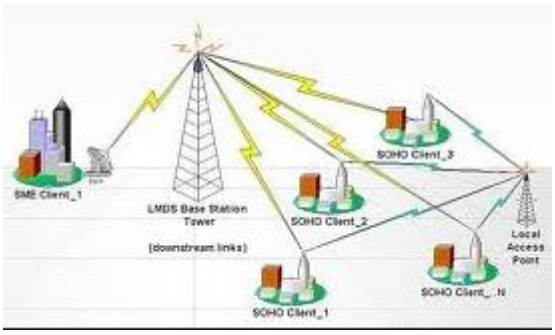
Радіорелейні канали передачі даних. Радіорелейні канали зв'язку складаються з послідовності станцій, які є ретрансляторами. Зв'язок здійснюється в межах прямої видимості, дальності між сусідніми станціями - до 50 км. Цифрові радіорелейні лінії зв'язку (ЦРРЗ) застосовуються в якості регіональних і місцевих систем зв'язку і передачі даних, а також для зв'язку між базовими станціями стільникового зв'язку.



Супутникові канали передачі даних. У супутникових системах використовуються антени СВЧ-діапазону частот для прийому радіосигналів від наземних станцій і ретрансляції цих сигналів назад на наземні станції. У супутникових мережах використовуються три основні типи супутників, які знаходяться на геостаціонарних орбітах, середніх або низьких орбітах. Супутники запускаються, як правило, групами. Рознесені один від одного вони можуть забезпечити охоплення майже всієї поверхні Землі. Доцільніше використовувати супутниковий зв'язок для організації каналу зв'язку між станціями, розташованими на дуже великих відстанях, і можливості обслуговування абонентів в самих важкодоступних точках. Пропускна здатність висока - кілька десятків Мбіт/с.

Стільникові канали передачі даних . Радіоканали стільникового зв'язку будуються за тими ж принципами, що і стільникові телефонні мережі. Стільниковий зв'язок - це бездротова телекомунікаційна система, що складається з мережі наземних базових приймально-передавальних станцій і стільникового комутатора (або центру комутації мобільного зв'язку).

Базові станції підключаються до центру комутації, який забезпечує зв'язок, як між базовими станціями, так і з іншими телефонними мережами і з глобальною мережею Інтернет. По виконуваних функцій центр комутації аналогічний звичайної АТС провідного зв'язку.



LMDS (Local Multipoint Distribution System) - це стандарт стільникових мереж бездротової передачі інформації для фіксованих абонентів. Система будується за стільниковим принципом, одна базова станція дозволяє охопити район радіусом кілька кілометрів (до 10 км) і підключити декілька тисяч абонентів. Самі БС об'єднуються один з одним високошвидкісними наземними каналами зв'язку або радіоканалами. Швидкість передачі даних до 45 Мбіт/с.

Радіоканали передачі даних WiMAX (Worldwide Interoperability for Microwave Access) аналогічні Wi-Fi. WiMAX, на відміну від традиційних технологій радіодоступу, працює і на відбитому сигналі, поза прямою видимістю базової станції. Експерти вважають, що мобільні мережі WiMAX відкривають набагато цікавіші перспективи для користувачів, ніж фіксований WiMAX, призначений для корпоративних замовників. Інформацію можна передавати на відстані до 50 км зі швидкістю до 70 Мбіт/с.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Радіоканали передачі даних MMDS (Multichannel Multipoint Distribution System). Ці системи здатна обслуговувати територію в радіусі 50-60 км, при цьому пряма видимість передавача оператора є не обов'язковою. Середня гарантована швидкість передачі даних складає 500 Кбіт/с - 1 Мбіт/с, але можна забезпечити до 56 Мбіт/с на один канал.

Радіоканали передачі даних для локальних мереж. Стандартом безпровідного зв'язку для локальних мереж є технологія Wi-Fi. Wi-Fi забезпечує підключення в двох режимах: точка-точка (для підключення двох ПК) і інфраструктурне з'єднання (для підключення кілька ПК до однієї точки доступу). Швидкість обміну даними до 11 Мбіт/с при підключенні точка-точка і до 54 Мбіт/с при інфраструктурному з'єднанні.

Радіоканали передачі даних Bluetooht - це технологія бездротового зв'язку, що дозволяє пристроям обмінюватися даними на невеликих відстанях (не більше 10 м). До одного пристрою з підтримкою Bluetooth можна підключити до семи інших пристроїв. Пристрої Bluetooth здатні обробляти голос і дані та працюють в радіочастотному діапазоні від 2,4 до 2,485 ГГц, (так званому діапазоні для використання в промислових, наукових і медичних цілях (Industrial, Scientific, and Medical - ISM). Стандарт Bluetooth включає в себе технологію адаптивної перебудови частоти (Adaptive Frequency Hopping - AFH), що дозволяє «перекидати» сигнали на різні частоти в діапазоні Bluetooth, знижуючи таким чином можливість виникнення перешкод при використанні декількох пристроїв Bluetooth. Швидкість передачі даних не перевищує 1 Мбіт/с

Комунікаційне обладнання комп'ютерних мереж

Основне комунікаційне обладнання: мережеві адаптери. **Мережевий адаптер** (мережева карта) - це пристрій двонаправленого обміну даними між ПК і середовищем передачі даних комп'ютерної мережі. Крім організації обміну даними між ПК і комп'ютерної мережею, мережевий адаптер виконує буферизацію (тимчасове зберігання даних) і функцію створення пари комп'ютера з мережевим кабелем.

Адаптери забезпечені власним процесором і пам'яттю. Карти класифікуються за типом порту, через який вони з'єднуються з комп'ютером: ISA, PCI, USB. Найбільш поширені з них - це мережеві карти PCI. Карта, як правило, встановлюється в слот розширення PCI, розташований на материнській платі ПК, і підключається до мережевого кабелю роз'ємами типу: RJ-45 або BNC.

Мережеві карти можна розділити на два типи: адаптери для клієнтських комп'ютерів; адаптери для серверів.

Залежно від застосовуваної технології комп'ютерних мереж Ethernet, Fast Ethernet або Gigabit Ethernet, мережеві карти забезпечують швидкість передачі даних: 10, 100 або 1000 Мбіт/с.

До **проміжного комунікаційного обладнання** комп'ютерних мереж відносять: трансивери (transceivers), повторювачі (repeaters), концентратори (hubs), комутатори (switches), мости (bridges), маршрутизатори (routers) і шлюзи (gateways). Проміжне комунікаційне обладнання комп'ютерних мереж використовується для посилення і перетворення сигналів, для об'єднання ПК в фізичні сегменти, для поділу

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

комп'ютерних мереж на підмережі (логічні сегменти) з метою збільшення продуктивності мережі, а також для об'єднання підмереж (сегментів) і мереж в єдину комп'ютерну мережу.

Фізична структуризація комп'ютерних мереж об'єднує ПК в загальну середу передачі даних, тобто утворює фізичні сегменти мережі, але при цьому не змінює напрямок потоків даних. Фізичні сегменти спрощують підключення до мережі великої кількості ПК.

Логічна структуризація розділяє загальне середовище передачі даних на логічні сегменти і тим самим усуває зіткнення (колізії) даних в комп'ютерних мережах. Логічні сегменти або підмережі можуть працювати автономно і в міру необхідності комп'ютери з різних сегментів можуть обмінюватися даними між собою. Протоколи управління в комп'ютерних мережах залишаються тими ж, які застосовуються і в неподільні мережах.

Трансивери і повторювачі забезпечують посилення і перетворення сигналів в комп'ютерних мережах. Концентратори і комутатори служать для об'єднання декількох комп'ютерів в необхідну конфігурацію локальної комп'ютерної мережі.

Концентратори є засобом фізичної структуризації комп'ютерної мережі, так як розбивають мережу на сегменти. Комутатори призначені для логічної структуризації комп'ютерної мережі, так як розділяють загальну середу передачі даних на логічні сегменти і тим самим усувають зіткнення.

Для з'єднання підмереж (логічних сегментів) і різних комп'ютерних мереж між собою в якості міжмережевого інтерфейсу застосовуються комутатори, мости, маршрутизатори і шлюзи.

Повторювачі - це апаратні пристрої, призначені для відновлення і посилення сигналів в комп'ютерних мережах з метою збільшення їх довжини.

Трансивери або приймачі - це апаратні пристрої, що служать для двобічної передачі між адаптером і мережним кабелем або двома сегментами кабелю. Основною функцією трансивера є посилення сигналів. Трансивери застосовуються і в якості конверторів для перетворення електричних сигналів в інші види сигналів (оптичні або радіосигнали) з метою використання інших середовищ передачі інформації.

Концентратори - це апаратні пристрої множинного доступу, які об'єднують в одній точці окремі фізичні відрізки кабелю, утворюють загальну середу передачі даних або фізичні сегменти мережі.

Комутатори - це програмно - апаратні пристрої, які ділять загальне середовище передачі даних на логічні сегменти. Логічний сегмент утворюється шляхом об'єднання декількох фізичних сегментів за допомогою концентраторів. Кожен логічний сегмент підключається до окремого порту комутатора.

Мости - це програмно - апаратні пристрої, які забезпечують з'єднання декількох локальних мереж між собою або кілька частин однієї і тієї ж мережі, що працюють з різними протоколами. Мости призначені для логічної структуризації мережі або для з'єднання в основному ідентичних мереж, що мають деякі фізичні відмінності. Міст

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

ізолює трафік однієї частини мережі від трафіку іншій частині, підвищуючи загальну продуктивність передачі даних.

Маршрутизатор. Це комунікаційне обладнання, яке забезпечує вибір маршруту передачі даних між декількома мережами, що мають різну архітектуру або протоколи. Маршрутизатор застосовують тільки для зв'язку однорідних мереж і в розгалужених мережах, що мають кілька паралельних маршрутів. Маршрутизаторами і програмними модулями мережевої операційної системи реалізуються функції мережевого рівня.

Шлюзи - це комунікаційне обладнання (наприклад, комп'ютер), що служить для об'єднання різнорідних мереж з різними протоколами обміну. Шлюзи повністю перетворюють весь потік даних, включаючи коди, формати, методи управління тощо.

Комунікаційне обладнання: мости, маршрутизатори і шлюзи в локальної комп'ютерної мережі - це, як правило, виділені комп'ютери зі спеціальним програмним забезпеченням.

2. *Класифікація мереж*

Залежно від відстаней між ПК розрізняють наступні комп'ютерні мережі:

– **локальні** комп'ютерні мережі (LAN - Local Area Networks) - комп'ютерні мережі, розташовані в межах невеликої обмеженій території (будівлі або в сусідніх будинках) не більше 10 - 15 км;

– територіальні комп'ютерні мережі, які охоплюють значну географічний простір. До територіальних мереж можна віднести **міські** (MAN - Metropolitan Area Network), **регіональні** (Regional computer network), **національні** (National computer network) і **глобальні** (WAN - Wide Area Network) мережі. Міські та регіональні мережі пов'язують абонентів району, міста або області. Глобальні мережі об'єднують абонентів, віддалених між собою на значну відстань, що знаходяться в різних країнах або континентах;

– **Персональні** мережі (PAN - Personal Area Network) це мережа, побудована «навколо» людини. PAN являє собою комп'ютерну мережу, яка підключає пристрої, такі як миші, клавіатури, принтери, смартфони і планшетні ПК, що знаходяться в межах досяжності окремого користувача. Всі ці пристрої підключаються до одного вузла, найчастіше за допомогою технології Bluetooth.

За адміністративним відносинам між вузлами можна виділити локальні мережі з **виділеними серверами** (серверні мережі), їх називаються ієрархічними і мережі без виділеного сервера (децентралізовані), так звані, **одноранговi** мережі.

У локальних мережах з виділеними серверами один з комп'ютерів є сервером, а інші ПК - робочими станціями. **Сервери** - це високопродуктивні комп'ютери з великої місткості пам'яті і з високошвидкісною мережевий картою, які відповідають за зберігання даних, організацію доступу до цих даних і передачу даних автоматизованого робочого місця або клієнтам. **Робочі станції** – це комп'ютери, з яких здійснюється доступ до інформації на сервері, називаються робочими станціями або клієнтами.

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

У одноранговых локальних мережах немає єдиного центру управління взаємодією робочих станцій і єдиного комп'ютера для зберігання даних. Одноранговая мережа - це сукупність рівноправних комп'ютерів, кожен з яких має унікальне ім'я і, як правило, пароль для входу в нього в момент завантаження ОС. Рівноправність ПК означає, що адміністратор кожного комп'ютера в локальній мережі може перетворити свій локальний ресурс в загальний і встановлювати права доступу до нього і паролі. Він же відповідає за збереження або працездатність цього ресурсу. Таким чином, одноранговая локальна мережа - це локальна комп'ютерна мережа (ЛКМ), в якій кожна робоча станція може розділити всі або деякі з своїх ресурсів з іншими робочими станціями мережі.

Але відсутність виділеного сервера не дозволяє адміністратору централізовано керувати всіма ресурсами локальної мережі. Кожна робоча станція може виконувати функції, як клієнта, так і сервера, тобто надавати ресурси іншим робочим станціям та використовувати ресурси інших робочих станцій.

До переваг одноранговой локальної мережі можна віднести низьку вартість; високу надійність, до недоліків: одноранговая мережа ефективна тільки при кількості одночасно працюючих станцій не більше 10, слабкий захист інформації, складність оновлення та зміни ПО робочих станцій.

Серверні локальні мережі (багаторівневі або ієрархічні). У таких мережах сервер забезпечує взаємодії між робочими станціями, виконує функції зберігання даних загального користування, організує доступ до цих даних і передає дані клієнтові. Клієнт обробляє отримані дані і надає результати обробки користувачеві. Необхідно відзначити, що обробка даних може здійснюватися і на сервері.

Локальні мережі з централізованим управлінням, в яких сервер призначений тільки зберігання і видачі клієнтам інформації за запитами, називаються **мережами з виділеним файл-сервером**. Системи, в яких на сервері поряд зі зберіганням здійснюється і обробка інформації, називаються **системами "клієнт-сервер"**.

Необхідно відзначити, що в серверних локальних мережах клієнту безпосередньо доступні тільки ресурси серверів. Але робочі станції, що входять до локальної мережі з централізованим управлінням, можуть одночасно організувати між собою одноранговую локальну мережу з усіма її можливостями.

Програмне забезпечення, що керує роботою локальної мережі з централізованим управлінням, складається з двох частин: мережевий операційної системи, яка встановлюється на сервері та програмного забезпечення на робочій станції, що представляє набір програм, що працюють під управлінням операційної системи, яка встановлена на робочій станції. При цьому на різних робочих станціях в одній мережі можуть бути встановлені різні операційні системи.

У великих ієрархічних локальних мережах в якості мережевих ОС використовуються UNIX та LINUX, які є більш надійними. Для локальних мереж середнього масштабу найбільш популярною мережевий ОС є Windows Server.

Залежно від способів використання сервера в ієрархічних мережах розрізняють сервери наступних типів:

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

– Файловий сервер. У цьому випадку на сервері знаходяться файли, що спільно оброблюються або програми, що спільно використовуються.

– Сервер баз даних. На сервері розміщується мережева база даних.

– Принт-сервер. До комп'ютера підключається досить продуктивний принтер, на якому може бути роздрукована інформація відразу з декількох робочих станцій.

– Поштовий сервер. На сервері зберігається інформація, що відправляється і отримується по локальній мережі.

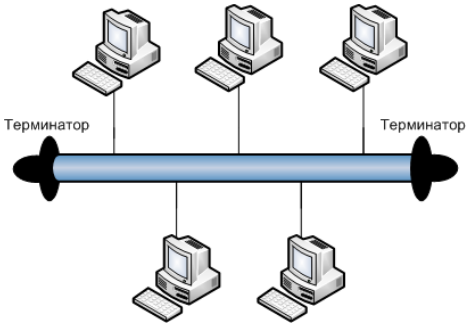
До переваг можна віднести: високу швидкість обробки даних; надійну систему захисту інформації та забезпечення таємності; простоту управління (у порівнянні з однорангових мережами); до недоліків: побудова мережі дорожче через наявність виділеного сервера та менш гнучка в порівнянні з рівноправною мережею.

Всі комп'ютери в локальній мережі з'єднані лініями зв'язку. Геометричне розташування ліній зв'язку щодо вузлів мережі і фізичне підключення вузлів до мережі називається **фізичною топологією**. Залежно від топології розрізняють мережі: шинної, кільцевої, зоряної, ієрархічної та довільної структури.

Розрізняють фізичну і логічну топологію. Логічна і фізична топології мережі незалежні один від одного. **Фізична топологія** - це геометрія побудови мережі, а **логічна топологія** визначає напрямки потоків даних між вузлами мережі і способи передачі даних.

В даний час в локальних мережах використовуються наступні **фізичні топології**:

– фізична "шина" (bus);
– фізична "зірка" (star);
– фізичне "кільце" (ring);
– фізична "зірка" і логічне "кільце" (Token Ring).



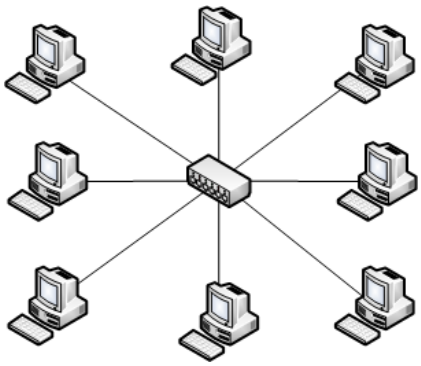
Шинна топологія. Мережі з шинної топологією використовують лінійний моноканал (коаксіальний кабель) передачі даних, на кінцях якого встановлюються кінцеві опори (так звані термінатори). Кожен комп'ютер підключається до коаксіального кабелю з допомогою Т-роз'єму (Т - конектор). Дані від передавального вузла мережі передаються по шині в обидві сторони, відбиваючись від кінцевих термінаторів. Термінатори запобігають відображення сигналів, тобто використовуються для гасіння сигналів, які досягають кінців каналу передачі даних.

Таким чином, інформація надходить на всі вузли, але приймається тільки тим вузлом, якому вона призначається. У топології логічна шина середовище передачі даних використовуються спільно і одночасно всіма ПК мережі, а сигнали від ПК поширюються одночасно у всіх напрямках по середовищі передачі. Так як передача сигналів в топології фізична шина є широкомовної, тобто сигнали поширюються одночасно у всіх напрямках, то логічна топологія даної локальної мережі є логічною шиною.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

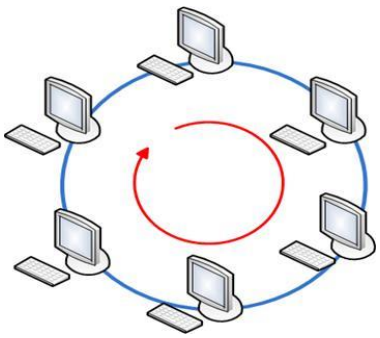
Переваги мереж шинної топології: відмова одного з вузлів не впливає на роботу мережі в цілому; мережу легко налаштовувати і конфігурувати; мережу стійка до несправностей окремих вузлів. Недоліки мереж шинної топології: розрив кабелю може вплинути на роботу всієї мережі; обмежена довжина кабелю і кількість робочих станцій; важко визначити дефекти з'єднань.



Топологія типу "зірка". У мережі побудованої по топології типу "зірка" кожна робоча станція з'єднується кабелем (витою парою) до концентратора або хабу (hub). Концентратор забезпечує паралельне з'єднання ПК і, таким чином, всі комп'ютери, підключені до мережі, можуть спілкуватися один з одним.

Дані від передавальної станції мережі передаються через хаб по всіх лініях зв'язку всім ПК. Інформація надходить на всі робочі станції, але приймається тільки тими станціями, яким вона призначається. Так як передача сигналів в топології фізична зірка є широкомовної, тобто сигнали від ПК поширюються одночасно у всіх напрямках, то логічна топологія даної локальної мережі є логічною шиною.

Переваги мереж топології зірка: легко підключити новий ПК; є можливість централізованого управління; мережу стійка до несправностей окремих ПК і до розривів з'єднання окремих ПК. Недоліки мереж топології зірка: відмова хаба впливає на роботу всієї мережі; велика витрата кабелю.



Топологія "кільце". У мережі з топологією кільце все вузли з'єднані каналами зв'язку в нерозривне кільце (необов'язково окружність), по якому передаються дані. Вихід одного ПК з'єднується з входом іншого ПК. Почавши рух з однієї точки, дані, в кінцевому рахунку, потрапляють на його початок. Дані в кільці завжди рухаються в одному і тому ж напрямку.

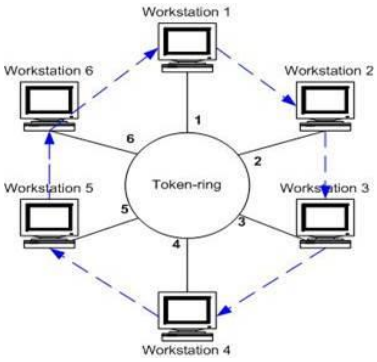
Приймаюча робоча станція розпізнає і отримує тільки адресований їй повідомлення. У мережі з топологією типу фізичне кільце використовується маркерний доступ, який надає станції право на використання кільця у визначеному порядку. Логічна топологія даної мережі - логічне кільце. Дану мережу дуже легко створювати і налаштовувати.

До основного недоліку мереж топології кільце є те, що пошкодження лінії зв'язку в одному місці або відмова ПК призводить до непрацездатності всієї мережі.

Як правило, в чистому вигляді топологія "кільце" не застосовується через свою ненадійність, тому на практиці застосовуються різні модифікації кільцевої топології.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua



Топологія Token Ring. Ця топологія заснована на топології "фізична кільце з підключенням типу зірка". У даній топології все робочі станції підключаються до центрального концентратора (Token Ring) як в топології фізична зірка. Центральний концентратор - це інтелектуальний пристрій, яке за допомогою перемичок забезпечує послідовне з'єднання виходу однієї станції зі входом іншої станції.

Іншими словами за допомогою концентратора кожна станція з'єднується тільки з двома іншими станціями (попередньої та подальшої станціями). Таким чином, робочі станції пов'язані петлею кабелю, по якій пакети даних передаються від однієї станції до іншої і кожна станція ретранслює ці послані пакети. У кожній робочій станції є для цього приймально-передавальний пристрій, який дозволяє управляти проходженням даних в мережі. Фізично така мережа побудована за типом топології "зірка".

В архітектурі Token Ring маркер передається від вузла до вузла по логічному кільцю, створеному центральним концентратором. Така маркерна передача здійснюється у фіксованому напрямку (напрямок руху маркера і пакетів даних представлено на малюнку стрілками синього кольору). Станція, що володіє маркером, може відправити дані іншої станції.

Для передачі даних робочі станції повинні спочатку дочекатися приходу вільного маркера. У маркері міститься адреса станції, яка послала цей маркер, а також адреса тієї станції, якій він призначається. Після цього відправник передає маркер наступної в мережі станції для того, щоб і та могла відправити свої дані.

Один з вузлів мережі (зазвичай для цього використовується файл-сервер) створює маркер, який відправляється в кільце мережі. Такий вузол виступає в якості активного монітора, який стежить за тим, щоб маркер ні загублений або зруйнований.

Переваги мереж топології Token Ring: топологія забезпечує рівний доступ до всіх робочих станцій; висока надійність, так як мережа стійка до несправностей окремих станцій і до розривів з'єднання окремих станцій. Недоліки мереж топології Token Ring: велика витрата кабелю і відповідно дорога розводка ліній зв'язку.

Локальні комп’ютерні мережі можуть відрізнятися мережевою архітектурою, що визначає топологію і метод доступу до середовища передачі даних, кабельну систему або середовище передачі даних, формат мережевих кадрів, тип кодування сигналів, швидкість передачі в локальній мережі. У сучасних локальних комп’ютерних мережах широкого поширення набули такі мережеві архітектури, як: Ethernet, Token-Ring, ArcNet, FDDI.

Мережеві технології локальних мереж **IEEE802.3 / Ethernet**. В даний час ця мережева технологія найбільш популярна в світі. Популярність забезпечується простими, надійними і недорогими технологіями. У класичній локальній мережі Ethernet застосовується стандартний коаксіальний кабель двох видів (товстий і тонкий). Однак все більшого поширення набула версія Ethernet, що використовує в якості середовища передачі виті пари, так як монтаж і обслуговування їх набагато простіше. У локальних мережах Ethernet застосовуються топології типу "шина" і типу

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

"пасивна зірка", а в мережі реалізується спосіб контролю доступу «множинний доступ з контролем несучої і виявленням конфліктів (CSMA / CD)».

У CSMA / CD всі пристрої прослуховують мережевий провідник і чекають, коли він буде вільний для відправки даних. Цей процес схожий на очікування сигналу готовності лінії на телефоні перед набором номера. Коли пристрій визначає, що ніякі інші пристрої не передають дані, воно намагається відправити дані. Якщо ніяке інший пристрій відправляє дані в цей же час, передача на комп'ютер призначення відбувається без проблем. Якщо в цей час інший пристрій виконає передачу, в провіднику виникне конфлікт.

У розвиток мережевої технології Ethernet створені високошвидкісні варіанти: **IEEE802.3u/Fast Ethernet** і **IEEE802.3z/Gigabit Ethernet**. Основна топологія, яка використовується в локальних мережах Fast Ethernet і Gigabit Ethernet, пасивна зірка. Локальні мережі Fast Ethernet і Gigabit Ethernet сумісні з локальними мережами, виконаними за технологією Ethernet, тому легко і просто з'єднувати сегменти Ethernet, Fast Ethernet і Gigabit Ethernet в єдину обчислювальну мережу.

Мережеві технології локальних мереж **IEEE802.5/Token-Ring**. Мережа Token-Ring передбачає використання середовища передачі даних, яка утворюється об'єднанням всіх вузлів в кільце. Мережа Token-Ring має зоряно-кільцеву топологію (основна кільцева і зоряна додаткова топологія). Для доступу до середовища передачі даних використовується маркерний метод (детермінований маркерний метод). Стандарт підтримує виту пару (екрановану і неекрановану) і оптоволоконний кабель. Максимальне число вузлів на кільці - 260, максимальна довжина кільця - 4000 м. Швидкість передачі даних до 16 Мбіт / с.

Мережеві технології локальних мереж **IEEE802.4 / ArcNet**. Як топології локальна мережа ArcNet використовує "шину" і "пасивну зірку". Підтримує екрановану і неекрановану виту пару і оптоволоконний кабель.

У мережі ArcNet для доступу до середовища передачі даних використовується метод передачі повноважень. Локальна мережа ArcNet - це одна з найстаріших мереж і користувалася великою популярністю. Серед основних переваг локальної мережі ArcNet можна назвати високу надійність, низьку вартість адаптерів і гнучкість.

Основним недоліків мережі є низька швидкість передачі інформації (2,5 Мбіт / с). Максимальна кількість абонентів - 255. Максимальна довжина мережі - 6000 метрів.

Мережеві технології локальних мережі **FDDI** (Fiber Distributed Data Interface). DDI- стандартизована специфікація для мережевої архітектури високошвидкісної передачі даних по оптоволоконних лініях. Швидкість передачі - 100 Мбіт / с. Ця технологія багато в чому базується на архітектурі Token-Ring і використовується детермінований маркерний доступ до середовища передачі даних.

Максимальна довжина кільця мережі - 100 км. Максимальна кількість абонентів мережі - 500. Мережа FDDI - це дуже високонадійна мережу, яка створюється на основі двох оптоволоконних кіл, що утворюють основний і резервний шляхи передачі

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

3. *Протоколи, стеки протоколів, модель TCP/IP*

Комп'ютерні мережі, як правило, складаються з різного обладнання різних виробників, і без прийняття всіма виробниками загальноприйнятих правил побудови ПК і мережевого устаткування, забезпечити нормальне функціонування мереж було б неможливо.

Тобто для забезпечення нормальної взаємодії цього обладнання в мережах потрібен єдиний уніфікований стандарт, який визначав би алгоритм передачі інформації в мережах. В сучасних обчислювальних мережах роль такого стандарту виконують **мережеві протоколи**.

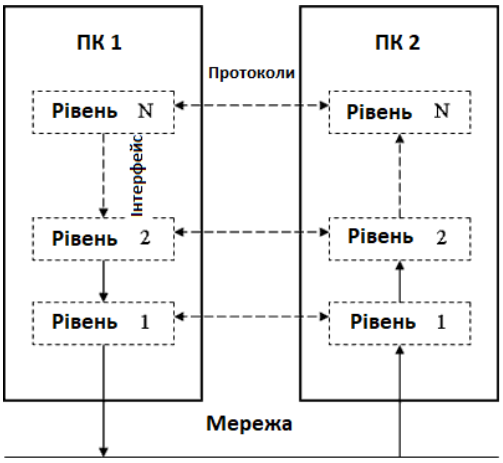
У зв'язку з тим, що описати єдиним протоколом взаємодії між пристроями в мережі не представляється можливим, то необхідно розділити процес мережевої взаємодії на ряд концептуальних рівнів (модулів) і визначити функції для кожного модуля і порядок їх взаємодії, застосувавши метод декомпозиції.

Використовується багаторівневий підхід методу декомпозиції, відповідно до якого безліч модулів, які вирішують приватні завдання, впорядковують за рівнями створюючи ієрархію, процес мережевої взаємодії можемо представити у вигляді ієрархічно організованої безлічі модулів.

Протоколи, інтерфейси, стеки протоколів

Багаторівневі представлення засобів мережевої взаємодії має свою специфіку, пов'язану з тим, що в процесі обміну повідомленнями беруть участь дві сторони, тобто необхідно організувати узгоджену роботу двох ієрархій, що працюють на різних комп'ютерах.

Обидва учасники мережевого обміну повинні прийняти безліч угод. Угоди повинні бути прийняті для всіх рівнів, починаючи від найнижчого - рівня передачі бітів - до найвищого, що реалізує сервіс для користувача. Декомпозиція передбачає чітке визначення функції кожного рівня і інтерфейсів між рівнями.



Взаємодія однойменних функціональних рівнів по горизонталі здійснюється за допомогою протоколом. **Протоколом** називається набір правил і методів взаємодії однойменних функціональних рівнів об'єктів мережевого обміну.

Взаємодії функціональних рівнів по вертикалі здійснюється через інтерфейси. Інтерфейс визначає набір функцій, які нижчий рівень надає вищому рівневі.

Таким чином, механізм передачі будь-якого пакета інформації через мережу від клієнтської програми, яка працює на одному комп'ютері ПК 1, до клієнтській програмі, що працює на іншому комп'ютері ПК 2, можна умовно представити у вигляді послідовної пересилки цього пакета зверху вниз від верхнього рівня, що забезпечує взаємодія з призначеним для користувача додатком, до нижнього рівня, організуючим

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

інтерфейс з мережею, його трансляції на комп'ютер ПК 2 і зворотної передачі верхнього рівня вже на ПК 2.

Комунікаційні протоколи можуть бути реалізовані як програмно, так і апаратно. Протоколи нижніх рівнів часто реалізуються комбінацією програмних і апаратних засобів, а протоколи верхніх рівнів - як правило, чисто програмними засобами. Протоколи реалізуються не тільки комп'ютерами, але і іншими мережевими пристроями - концентраторами, мостами, комутаторами, маршрутизаторами тощо. Залежно від типу пристроїв в ньому повинні бути вбудовані засоби, що реалізують той чи інший набір протоколів.

Ієрархічно організований набір протоколів, достатній для організації взаємодії вузлів в мережі, називається **стеком комунікаційних протоколів**. В мережі Інтернет базовим набором протоколів є стек протоколів TCP / IP.

Мережева модель TCP / IP

Стек протоколів TCP / IP - набір мережевих протоколів передачі даних, що використовуються в мережах, включаючи мережу Інтернет. Назва TCP / IP відбувається з двох найважливіших протоколів сімейства - Transmission Control Protocol (TCP) і Internet Protocol (IP), які були розроблені і описані першими в даному стандарті.

> У липні 1977 Гвинт Серф і Боб Кан вперше продемонстрували передачу даних з використанням TCP по трьом різним мережам. Пакет пройшов за наступним маршрутом: Сан-Франциско - Лондон - Університет Південної Каліфорнії. В кінці своєї подорожі пакет виконав 150 тисяч км, не втративши жодного біта. У 1978 році Серф, Постел і Дені Кохен вирішили виділити в TCP дві окремі функції: TCP і протокол Інтернету (Internet Protocol, IP). TCP був відповідальний за розбивку повідомлення на фрагменти (дейтаграми) і з'єднання їх в кінцевому пункті відправлення. IP відповідав за передачу (з контролем отримання) окремих дейтаграм. Ось так народився сучасний протокол Інтернету. А 1 січня 1983 року ARPANET перейшла на новий протокол. Цей день прийнято вважати офіційною датою народження Інтернету.

Протоколи працюють один з одним в стеці (stack, стопка) - це означає, що протокол, що розташовується на рівні вище, працює «поверх» нижнього, використовуючи механізми інкапсуляції. Наприклад, протокол TCP працює поверх протоколу IP. TCP (протокол управління передачею) - відповідає за перевірку коректної доставки даних від клієнта до сервера. Дані можуть бути втрачені в проміжній мережі. TCP додана можливість виявлення помилок або втрачених даних і, як наслідок, можливість запросити повторну передачу, до тих пір, поки дані коректно і повністю не будуть отримані. IP (інтернет протокол) - відповідає за передачу пакета даних від вузла до вузла. IP пересилає кожен пакет на основі адреси призначення (IP-адреса).

Основні характеристики TCP / IP:

- Стандартизовані протоколи високого рівня, що використовуються для добре відомих призначених для користувача сервісів.

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

- Використовуються відкриті стандарти протоколів, що дає можливість розробляти і допрацьовувати стандарти незалежно від програмного і апаратного забезпечення;
- Система унікальною адресації;
- Незалежність від використовуваного фізичного каналу зв'язку;

Стек протоколів TCP / IP включає в себе чотири рівні:

- прикладний рівень (application layer),
- транспортний рівень (transport layer),
- мережевий рівень (Internet layer),
- канальний рівень (link layer).

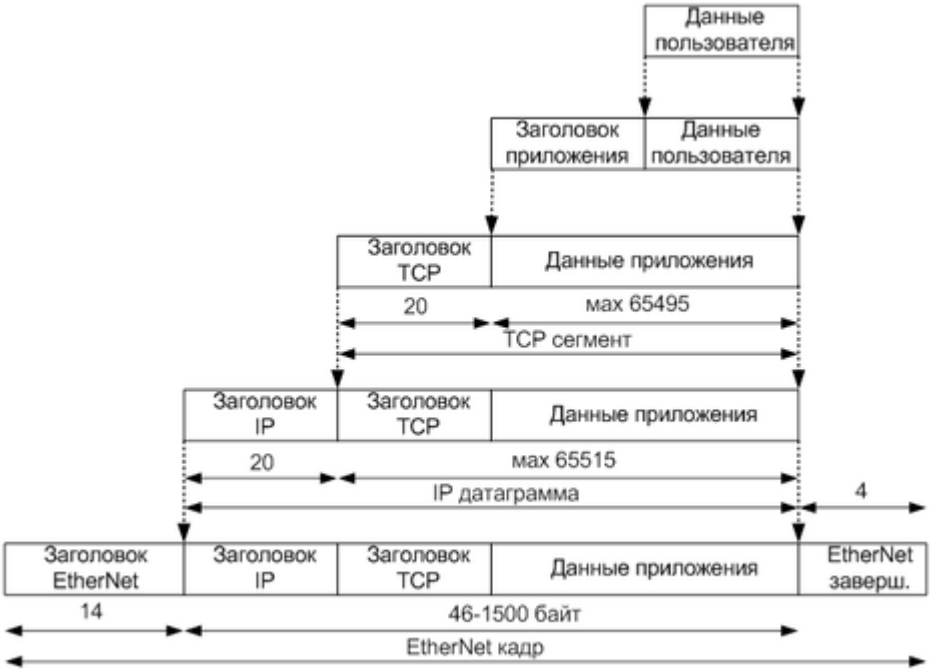
Коли пакет просувається по рівням зверху вниз - на кожному рівні до нього додається службова інформація у вигляді заголовка і можливо трейлера (інформації вміщеній в кінець повідомлення). Цей процес називається **інкапсуляція**. Службова інформація призначається для об'єкта того ж рівня на віддаленому комп'ютері. Її формат і інтерпретація визначаються протоколами даного рівня.

Якщо пакет просувається по рівню від низу до верху - він розділяється на заголовок і дані. Аналізується заголовок пакета, виділяється службова інформація і відповідно до неї дані перенаправляються до одного з об'єктів вищого рівня. Вищий рівень, в свою чергу, аналізує ці дані і також їх розділяє їх на заголовок і дані, далі аналізується заголовок і виділяється службова інформація і дані для вищого рівня. Процедура повторюється заново поки призначені для користувача дані, звільнені від всієї службової інформації, не дійдуть до прикладного рівня.

Не виключено, що пакет так і не дійде до прикладного рівня. Зокрема, якщо комп'ютер працює в ролі проміжної станції на шляху між відправником і отримувачем, тоді об'єкт, на відповідному рівні, при аналізі службової інформації визначить, що пакет на цьому рівня адресований не йому, в слідстві чого, об'єкт проведе необхідні заходи для перенаправлення пакета до пункту призначення або повернення відправнику з повідомленням про помилку. Але так чи інакше не буде здійснювати просування даних на верхній рівень.

Приклад інкапсуляції можна представити таким чином:

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua



Розглянемо кожні функції рівнів

Прикладний рівень. Поширеними прикладами додатків є програми: Telnet, FTP , HTTP, WWW, протоколи електронної пошти (SMTP, POP3).

Для пересилання даних іншому додатку, додаток звертається до того чи іншого модулю транспортного модуля.

Транспортний рівень. Протоколи транспортного рівня забезпечують прозору доставку даних між двома прикладними процесами. Процес, який одержує або відправляє дані, за допомогою транспортного рівня ідентифікується в таких межах номером, який називається номером порту.

Таким чином, роль адреси відправника і одержувача на транспортному рівні виконується номером порту. Аналізуючи заголовок свого пакета, отриманого від мережевого рівня, транспортний модуль визначає за номером порту одержувача по якому з прикладних процесів спрямовані дані і передає ці дані до відповідного прикладному процесу.

Номер порту одержувача і відправника записується в заголовок транспортним модулем, який відправляє дані. Заголовок транспортного рівня містить також і деяку іншу службову інформацію, і формат заголовка залежить від транспортного протоколу, що використовується.

Засоби транспортного рівня це функціональна надбудова над мережевим рівнем для вирішення двох основних завдання: забезпечення доставки даних між конкретними програмами, функціонуючими на різних вузлах мережі та забезпечення **гарантованої** доставки масивів даних довільного розміру.

В даний час в Інтернет використовуються два транспортних протоколу - **UDP**, що забезпечує **негарантовану** доставку даних між програмами, і **TCP**, що забезпечує **гарантовану** доставку з встановленням віртуального з'єднання.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Мережевий (міжмережевий) рівень. Основним протоколом цього рівня є протокол IP, який доставляє блоки даних (так звані дейтаграми) від однієї IP-адреси до іншого. IP-адреса є унікальним 32-х бітовим ідентифікатором комп'ютера, точніше його мережевого інтерфейсу. Дані для дейтаграми передаються IP модулю транспортним рівнем. IP модуль додає до цих даних заголовок, що містить IP-адресу відправника і одержувача, і іншу службову інформацію.

Таким чином, сформована дейтаграмма передається на рівень доступу до середовища передачі, для відправки по каналу передачі даних.

Не всі комп'ютери можуть безпосередньо зв'язатися один з одним, часто щоб передати дейтаграмму за призначенням потрібно направити її через один або кілька проміжних комп'ютерів по тому чи іншому маршруту. Завдання визначення маршруту для кожної дейтаграми вирішується протоколом IP.

Коли модуль IP отримує дейтаграмму з нижнього рівня, він перевіряє IP адреса призначення, якщо дейтаграмма адресована даному комп'ютера, то дані з неї передаються на обробку модулю вищого рівня, якщо ж адреса призначення дейтаграми чужий, то модуль IP може прийняти два рішення: знищить дейтаграмму або надіслати її далі до місця призначення, визначивши маршрут прямування, так роблять проміжні станції - маршрутизатори.

Також на границі мереж з різними характеристиками, може знадобитися розбити дейтаграмму на фрагменти, а потім зібрати їх в єдине ціле на комп'ютері одержувача - це також задача протоколу IP.

Також протокол IP може відправляти повідомлення, наприклад, в разі знищення дейтаграми. Більш ніяких засобів контролю коректності даних, підтвердження або доставки, попереднього з'єднання в протоколі немає, ці завдання покладені на транспортний рівень.

Канальний рівень (рівень доступу до середовища передачі). Функції цього рівня наступні: відображення IP-адрес в фізичні адреси мережі. Цю функцію виконує протокол ARP - протокол дозволу адрес (Address Resolution Protocol); інкапсуляція IP-дейтаграм в кадри для передачі по фізичному каналу і витяг дейтаграм з кадрів, при цьому не потрібно будь-якого контролю безпомилкової передачі, оскільки в стек TCP / IP такий контроль покладено на транспортний рівень або на сам додаток; визначення методу доступу до середовища передачі, тобто способу, за допомогою якого комп'ютери встановлює своє право на передачу даних; визначення представлення даних у фізичному середовищі; пересилання та прийом кадру.

Формат адрес у стеці TCP / IP

У стеці TCP / IP використовуються три типи адрес:

- Локальні або апаратні адреси (MAC-адрес, наприклад: F4-D2-32-12-43-FD);
- Мережевий адрес (IP-адреси, наприклад: 113.17.10.123);
- Символьні доменні імена, наприклад: www.nuczu.edu.ua.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Локальні адреси. У термінології TCP / IP під локальним адресом розуміється такий тип адреси, який використовується засобами базової технології для доставки даних в межах підмережі, що є елементом складовою великої мережі. У різних підмережах допустимі різні мережеві технології, різні стеки протоколів, тому при створенні стека TCP / IP передбачалася наявність різних типів локальних адрес. Якщо підмережею є локальна мережа, то локальна адреса - це MAC-адреса (Media Access Control).

MAC-адреса призначається мережевим адаптерам і мережевим інтерфейсам маршрутизаторів. MAC-адреси призначаються виробниками устаткування і є унікальними, так як управляються централізовано. Він відноситься до пристрою незалежно від того, до якої мережі він підключений. MAC-адресу має довжину 48 біт і може бути представлений в одному з трьох шістнадцятирічних форматів, показаних нижче:

11-A0-17-3D-BC-01	Дві шістнадцятирічні цифри, що розділені тире
11:A0:17:3D:BC:01	Дві шістнадцятирічні цифри, що розділені двокрапкою
11A0.173D.BC01	Чотири шістнадцятирічні цифри, що розділені крапкою

Мережева адреса. IP-адреси являють собою основний тип адрес, на підставі яких мережевий рівень передає пакети між мережами.

Наразі комп'ютер, як правило, може працювати з двома версіями IP-адрес. На початку 1990-х років виникла стурбованість з приводу нестачі мережевих адрес IPv4. Інженерна група з розвитку Інтернету (IETF) почала пошуки альтернативних рішень. Це призвело до розробки рішення, яке зараз відоме як протокол IP версії 6 (IPv6). В даний час адреси IPv6 використовуються паралельно з адресами IPv4 і вже починають їх витісняти.

Адреса IPv4 це унікальна 32-розрядна послідовність двійкових цифр, за допомогою яких комп'ютер однозначно ідентифікується в IP-мережі.

Для зручності роботи з IP-адресами 32- послідовність зазвичай поділяють на 4 частини по 8 біт (октети), кожен октет переводять в десяткове число і при запису поділяють ці числа точками. У такому вигляді IP-адреси займають набагато менше місця і набагато легше запам'ятовуються, відповідно до таблиці.

IP-адреса у вигляді 32-розрядної послідовності	00001010011001001100100000000001			
IP-адреса поділена на октети	00001010	01100100	11001000	00000001
Октети, що переведені в десяткове число	10	100	200	1
IP-адреса у вигляді десяткових чисел, розподілених крапками	10.100.200.1			

Всі IP-адреси можна розділити на дві логічні частини - номера мережі (NetID) і номера вузла мережі (номер хоста - HostID). Щоб визначити яка саме частина IP-

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

адреси належить до номера мережі, а яка - до номера хоста, залежить від значення перших біт адреси. Також, перші біти IP-адреси використовуються для того, щоб визначити до якого класу належить та чи інша IP-адреса.

Якщо адреса починається з 0, то мережу відносять до класу A і номер мережі займає один байт (8 біт), інші 3 байти інтерпретуються як номер вузла в мережі. Мережі класу A мають номери в діапазоні від 1 до 126. (Номер 0 не використовується, а номер 127 зарезервований для спеціальних цілей) Мереж класу A небагато, зате кількість вузлів в них може досягати 2^{24} = 16 777 216 вузлів.

Якщо перші два біти адреси рівні 10, то мережа відноситься до класу B. У мережах класу B під номер мережі і під номер вузла відводиться по 16 біт, тобто по 2 байти. Таким чином, мережа класу B є мережею середніх розмірів з максимальним числом вузлів 2^{16} = 65 536 вузлів.

Якщо адреса починається з послідовності 110, то це мережа класу C. У цьому випадку під номер мережі відводиться 24 біти, а під номер вузла - 8 біт. Мережі цього класу найбільш поширені, число вузлів в них обмежене 2^8, тобто 256 вузлами.

Якщо адреса починається з послідовності 1110, то він є адресою класу D позначає особливий, груповий адреса - multicast. Якщо в пакеті як адреса призначення вказана адреса класу D, то такий пакет повинні отримати всі вузли, яким визначено цю адресу.

Якщо адреса починається з послідовності 11110, то це означає, що дана адреса відноситься до класу E. Адреси цього класу зарезервовані для майбутніх застосувань.

Великі мережі отримують адреси класу A, середні - класу B, а маленькі - класу C.

Клас	Перші біти	Найменший номер мережі	Найбільший номер мережі	максимальна кількість вузлів в мережі
A	0	1.0.0.0	126.0.0.0	2^{24} -2
B	10	128.0.0.0	191.255.0.0	2^{16} -2

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

C	110	192.0.0.0	223.255.255.0	2^8 -2
D	1110	224.0.0.0	239.255.255.255.255	multicast
E	11110	240.0.0.0	247.255.255.255.255	зарезервовані

Спочатку механізм видачі діапазонів IP-адрес працював штатно, це було пов'язано з тим, що спочатку в організаціях була невелика кількість ПК і відповідно невеликі обчислювальні мережі. Але в зв'язку з подальшим бурхливим зростанням інтернету і мережевих технологій описаний підхід до розподілу IP-адрес став видавати збої, в основному пов'язані з мережами класу «B». Дійсно, організаціям, в яких число комп'ютерів не перевищувало кількох сотень (скажімо, 500), доводилося реєструвати для себе цілу мережу класу «B» (так як клас «C» тільки для 254 комп'ютерів, а клас «B» - 65534). Через що доступних мереж класу «B» стало, просто на просто, не вистачати, але при цьому великі діапазони IP-адрес пропадали даремно.

Так як, традиційна схема ділення IP-адреси на номер мережі (NetID) і номер вузла (HostID) заснована на понятті класу, який визначається значеннями декількох перших біт адреси, тому, перший байт адреси 185.23.44.206 потрапляє в діапазон 128-191, ми можемо сказати, що ця адреса відноситься до класу B, а значить, номером мережі є перші два байти, доповнені двома нульовими байтами - 185.23.0.0, а номером вузла - 0.0.44.206.

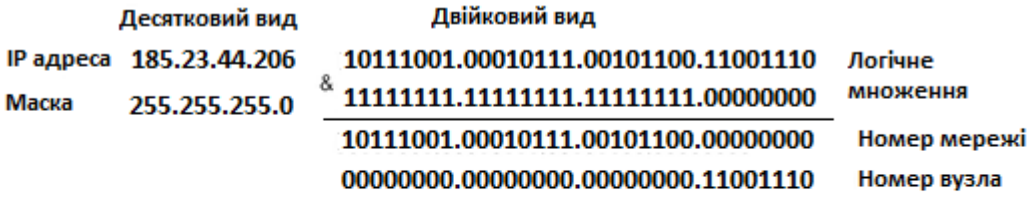
Для більш гнучкого встановлення межі між номером мережі і номером вузла широкого поширення набула так звана маска. **Маска** - це число, яке використовується в парі з IP-адресою; двійковий запис маски містить одиниці в тих розрядах, які повинні в IP-адресі інтерпретуватися як номер мережі. Оскільки номер мережі є цілісною частиною адреси, одиниці в масці також повинні представляти безперервну послідовність.

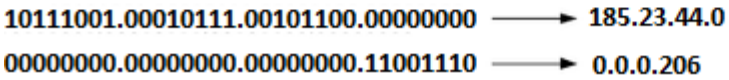
Для стандартних класів мереж маски мають таке значення:

- клас A - 11111111. 00000000. 00000000. 00000000 (255.0.0.0);
- клас B - 11111111. 11111111. 00000000. 00000000 (255.255.0.0);
- клас C - 11111111. 11111111.11111111. 00000000 (255.255.255.0).

Забезпечуючи кожен IP-адрес маскою, можна відмовитися від понять класів адрес і зробити більш гнучкою систему адресації. Наприклад, якщо розглянутий вище адресу 185.23.44.206 асоціювати з маскою 255.255.255.0, то номером мережі буде 185.23.44.0, а не 185.23.0.0, як це визначено системою класів.

Розрахунок номера мережі і номера вузла за допомогою маски:





Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

У масках кількість одиниць в послідовності, що визначає границю номера мережі, не обов'язково має бути кратним 8, щоб повторювати поділ адреси на байти. Нехай, наприклад, для IP-адреси 129.64.134.5 вказана маска 255.255.128.0, тобто в двійковому вигляді:

IP-адреса 129.64.134.5 - 10000001. 01000000.10000110. 00000101

Маска 255.255.128.0 - 11111111.11111111.10000000. 00000000

Якщо ігнорувати маску, то відповідно до системи класів адреса 129.64.134.5 відноситься до класу B, а значить, номером мережі є перші 2 байта - 129.64.0.0, а номером вузла - 0.0.134.5. Якщо ж використовувати для визначення границі номеру мережі маску, то 17 послідовних одиниць в масці, «накладені» (логічне множення) на IP-адресу, визначають як номер мережі в довічним вираженні число:

$$\begin{array}{l} 10000001.01000000.10000110.00000101 \\ \& \quad 11111111.11111111.10000000.00000000 \\ \hline 10000001.01000000.10000000.00000000 \end{array}$$

або в десяткового формі запису - номер мережі 129.64.128.0, а номер вузла 0.0.6.5.

Існує також короткий варіант запису маски, званий префіксом або короткою маскою. Зокрема мережу 80.255.147.32 з маскою 255.255.255.252, можна записати у вигляді 80.255.147.32/30, де «/ 30» вказує на кількість двійкових одиниць в масці, тобто тридцять бінарних одиниць (відлік ведеться зліва направо).

Механізм масок широко поширений в IP-маршрутизації, причому маски можуть використовуватися для самих різних цілей. З їх допомогою адміністратор може структурувати свою мережу, не вимагаючи від постачальника послуг додаткових номерів мереж. На основі цього ж механізму постачальники послуг можуть об'єднувати адресні простори декількох мереж шляхом введення так званих «префіксів» з метою зменшення обсягу таблиць маршрутизації і підвищення за рахунок цього продуктивності маршрутизаторів. Крім цього записувати маску у вигляді префікса значно коротше.

Одним з найголовніших недоліків інтернет протоколу IPv4 є відносно невелика кількість адрес – близько 4,23 мільярда, так як це число вже не здається такою великою в порівнянні з кількістю задіяних пристроїв підключених до мережі інтернет. До цього дня використання IPv4 проходить штатно, оскільки використовуються різні технології економії використання мережевих адрес, зокрема технологія NAT (NetworkAddressTranslation, перетворення мережевих адрес), але вже всім зрозуміло, що дні експлуатації IPv4 добігають кінця, оскільки в найближчому майбутньому передбачається наділяти можливістю доступу до інтернету всіх побутових приладів (холодильників, СВЧ-печей), для здійснення управління даними приладами віддалено, за допомогою мережі з будь-якої точки Землі.

У ситуації, що склалася перехід на новий формат мережевої адреси стає вкрай острів. Хоча багато фахівців передбачали проблему нестачі мережевих адрес ще на

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

початку 1990 року, в той же час почала працювати група проектування Інтернету над новою версією мережевого протоколу - IPv6.

Протокол IPv6 (Internet Protocol version 6) - це нова версія інтернет протоколу (IP), створена з метою вирішення проблем, з якими зіткнулася попередня версія (IPv4) при її використанні в інтернеті, одна з яких - це використання довжини адреси 128 біт замість 32 .

Довжина IPv6-адрес становить 128 біт, написаних у вигляді рядка шістнадцятирічних значень. Кожні 4 біта представляються однією шістнадцятковою цифрою, утворюючи 32 шістнадцяткових значення. Наприклад: 1FA9: FFFF: 2621: ACDA: 2245: BF98: 3412: 4167.

Унаслідок великої довжини адреса зазвичай містить багато нулів підряд. Для спрощення запису адрес використовується стисла форма, в якій суміжні послідовності нульових блоків замінюються парами символів двокрапок (: :). Однак такий символ може зустрічатися в адресі лише один раз.

Наприклад: адреса групової розсилки FFEA: 0: 0: 0: 0: CA28 1210: 4362 має стислу форму FFEA :: CA28 1210: 4362.

Адреса одноадресної розсилки 3FFE: FFFF: 0: 0: 8: 800: 02A1: 0 в стислій формі має вигляд: 3FFE: FFFF :: 8: 800: 02A1: 0.

Символьні доменні імена. Символьні імена в IP-мережах називаються доменними і будуються за ієрархічною ознакою. Складові повного символьного імені в IP-мережах розділяються крапкою і перераховуються в наступному порядку: спочатку просте ім'я кінцевого вузла, потім ім'я групи вузлів (наприклад, ім'я організації), потім ім'я більшої групи (поддомена) і так до імені домена найвищого рівня (наприклад, домену об'єднує організації за географічним принципом: RU - Росія, UK - Великобританія, SU - США). Прикладів доменного імені може служити ім'я nuczu.edu.ua. Між доменним ім'ям і IP-адресою вузла немає ніякого алгоритмічного відповідності, тому необхідно використовувати якісь додаткові таблиці або служби, щоб вузол мережі однозначно визначався як по доменному імені, так і по IP-адресою. У мережах TCP / IP використовується спеціальна розподілена служба Domain Name System (DNS), яка встановлює це відповідність на підставі створюваних адміністраторами мережі таблиць відповідності. Тому доменні імена називають також DNS-іменами.

4. Глобальна мережа Інтернет

Internet - всесвітня інформаційна комп'ютерна мережа, що є об'єднанням безлічі регіональних комп'ютерних мереж і безліч комп'ютерів (хостів), які обмінюють один з одним інформацією по каналах телекомунікацій. Хост - це будь-який комп'ютер, який є постійною частиною Internet, тобто з'єднаний по Internet - протоколу з іншим хостом, який в свою чергу, з'єднаний з іншим, і так далі. Комп'ютери, підключені до мережі Интернет, можуть мати будь-які апаратні і програмні платформи, але при цьому вони повинні підтримувати стек протоколів (сімейство протоколів) зв'язку TCP / IP. Єдиного власника і центру управління мережі Інтернет не існує.

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Інтернет почав своє існування з мережі ARPANet в 1969 році. Ця комп'ютерна мережа із застосуванням технології комутації пакетів була створена в США за завданням військового відомства США як високонадійна мережу передачі даних. У 1983 році ARPANet розділилася на дві мережі, одна - MILNET стала частиною оборонної мережі передачі даних США, інша - була використана для з'єднання академічних та дослідницьких центрів, яка поступово розвивалася і в 1990 році трансформувалася в Інтернет.

Інформація в Internet зберігається на серверах. Сервери мають свої адреси і управляються спеціалізованими програмами. Вони дозволяють пересилати пошту і файли, проводити пошук в базах даних і виконувати інші завдання.

Обмін інформацією між серверами мережі виконується по високошвидкісних каналах зв'язку (виділеним телефонним лініям, оптоволоконним і супутниковим каналам зв'язку). Доступ окремих користувачів до інформаційних ресурсів Internet зазвичай здійснюється через провайдера або корпоративну мережу.

Поняття провайдера

Провайдер (Internet Service Provider, ISP) - постачальник мережевих послуг - особа або організація надають послуги з підключення до комп'ютерних мереж. Різні провайдери обслуговують різні галузі і географічні регіони. Для порівняння досить уявити, як працюють постачальники електроенергії. По суті, це величезна мережа, яка обслуговується різними організаціями. Провайдери працюють за такою ж схемою.

Усіх провайдерів умовно можна поділити на компанії першого і другого рівня:

- Провайдери світового рівня володіють міжконтинентальними вузлами, які забезпечують зв'язок між материками за допомогою кабелів, прокладених по дну океанів. Це провайдери гіганти, на яких тримається вся мережа інтернет. Таких компаній в світі всього 11.
- Вторинні провайдери, до них відносяться всі, хто розподіляє інтернет по країнах і містах (регіональні постачальники).

Провайдер інтернету - це загальний термін, який позначає компанію або корпорацію, яка надає споживачам доступ до інтернету. З точки зору законодавчих актів Інтернет-провайдер - це оператор зв'язку, що має ліцензію на один з наступних видів послуг: Послуги зв'язку з надання каналів зв'язку, Послуги зв'язку в мережі передачі даних, за винятком передачі голосової інформації, Послуги зв'язку з передачі голосової інформації в мережі передачі даних, Телематичні послуги зв'язку.

Останнім часом провайдери все частіше об'єднуються з іншими телекомунікаційними компаніями і пропонують абонентам цілий пакет послуг: інтернет, кабельне телебачення, мобільний інтернет. Такі організації називаються операторами універсальних послуг зв'язку.

Провайдери не обмежуються однією лише постачанням інтернету. Адже для того, щоб без проблем виходити в інтернет, одного підключення кабелю недостатньо. Щоб комп'ютер зміг відкривати різні сайти, він повинен зв'язатися з сервером, де розміщена потрібна користувачеві інформація.

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Завдання ж інтернет-провайдера забезпечити надійне з'єднання комп'ютера користувача з будь-яким запитуваною їм сервером, де розміщується потрібний йому сайт. Це реалізується завдяки комутаційному обладнанню, в яке входять сервера, маршрутизатори, комутатори та інше обладнання.

Щоб не плутатися в поняттях, провайдерам було присвоєно класифікація в залежності від їх виду діяльності. Розглянемо коротко, чим вони відрізняються між собою:

- Постачальник послуг (Internet Service Provider, ISP) виконує виключно транспортну функцію, передає трафік кінцевого споживача.
- Постачальник інтернет-контенту (Internet Content provider, ICP) володіє своїми інформаційними ресурсами (доступ до веб-сайтів, IP адреси).
- Постачальник послуг хостингу (Hosting Service Provider, HSP) має в розпорядженні своє інтернет-сховище, власні канали зв'язку і сервери для розміщення контенту. Надає такі послуги як сховище даних, реєстрація доменних імен, віртуальний виділений сервер (VDS) та ін. Найчастіше носить назви: хостер, хостинг-провайдер, веб-хостер.
- Провайдери білінгових послуг (Billing Service Provider, BSP) забезпечують безпечну оплату банківських та інших рахунків через інтернет. Їм належать різні онлайн-сервіси та майданчики для електронних платежів.
- Провайдери послуг з доставки контенту (Content Delivery Provider, CDP) оптимізують доставку трафіку. Доставляють контент в ті точки доступу, які знаходяться найближче до споживачів. В результаті швидкість доступу абонентів до глобальної мережі підвищується.
- Постачальники послуг з підтримки онлайн-додатків (Application Service Provider, ASP). Відкривають доступ до масштабних програмних продуктів (таким як SAP R3), призначеним для управління великими підприємствами.

Способи доступу або підключення до Інтернет

В даний час існує безліч способів з'єднання з мережею Інтернет від підключення комп'ютера за допомогою аналогового модему до способів підключення з використанням високошвидкісних технологій.

Спосіб підключення комп'ютера до мережі Інтернет залежить від використовуваного користувачем рівня послуг, які він хоче отримати від провайдера (постачальника послуг), від швидкості і якості передачі даних. До послуг, які надаються Інтернет, відносяться: E-mail, WWW, FTP, Usenet, IP - телефонія, потокове відео і т.д.

Способи підключення до Інтернет можна класифікувати за такими видами:

- комутований доступ;
- доступ по виділених лініях;
- доступ по широкосмугової мережі (DSL - Digital Subscriber Line);
- доступ до Інтернет по локальній мережі;
- супутниковий доступ в Інтернет;

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

- доступ до Інтернет з використанням каналів кабельної телевізійної мережі;
- бездротові технології.

Для комутованого доступу, як правило, використовується аналоговий модем і аналогова телефонна лінія, але застосовується і комутований доступ по цифровій телефонній мережі ISDN (цифрова мережа зв'язку з інтеграцією послуг). Для підключення ПК до цифрової мережі з інтеграцією послуг ISDN використовується ISDN-адаптер. Крім того, комутований доступ до Інтернет може здійснюватися за допомогою бездротових технологій: мобільний GPRS - Інтернет і мобільний CDMA - Internet.

Доступ по виділених каналах зв'язку передбачає постійний канал зв'язку від приміщень з комп'ютером до комутатора, що належить ISP (провайдеру). Цей спосіб доступу забезпечує підключення комп'ютера всі 24 години на добу. Існує кілька варіантів підключення: через виділені лінії зі швидкостями 2400 біт / с - 1,544 Мбіт / с. і за постійними віртуальним каналам комутації кадрів зі швидкостями 56, Кбіт / с - 45 Мбіт / с. Для великих організацій цей метод підключення локальної мережі до Інтернет є найбільш ефективним.

Перспективним методом підключення до Інтернет, як для фізичних осіб, так і для компаній є широкосмугова мережа DSL. Digital Subscriber Line - сімейство цифрових абонентських ліній, призначених для організації доступу по аналоговій телефонній мережі, використовуючи DSL / кабельний модем. Цей спосіб забезпечує передачу даних до 50 Мбіт / с.

Доступ до Інтернет по локальній мережі з архітектурою Fast Ethernet забезпечує користувачеві доступ до ресурсів глобальної мережі Інтернет і ресурсів локальної мережі. Підключення здійснюється за допомогою мережевої карти (10/100 Мбіт / с) зі швидкістю передачі даних до 1 Гбіт / с на магістральних ділянках і 100 Мбіт / сек для кінцевого користувача.

Супутниковий доступ до Інтернет (DirecPC, Europe Online) є популярним для користувачів віддалених районів. Максимальна швидкість прийому даних до 52,5 Мбіт / с (реальна середня швидкість до 3 Мбіт / с).

Користувачі кабельного телебачення для підключення до Інтернет можуть використовувати канали кабельної телевізійної мережі, при цьому швидкість прийому даних від 2 до 56 Мб / сек. Для організації підключення до кабельної телевізійної мережі використовується кабельний модем.

Останнім часом все більш популярними стають бездротові методи підключення до Інтернет.

До бездротових технологій останньої милі відносяться:

- WiFi;
- WiMax;
- RadioEthernet;
- MMDS;
- LMDS;

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

- мобільний GPRS - Internet;
- мобільний CDMA - Internet.

Адресація в глобальній мережі Інтернет – сервіс DNS

Як було сказано раніше, основним протоколом мережі Інтернет є мережевий протокол TCP / IP. Кожен комп'ютер, в мережі TCP / IP (підключений до мережі Інтернет), має свою унікальну IP-адресу або IP - номер. Адреси в Інтернеті можуть бути представлені як послідовністю цифр, так і ім'ям, побудованим за певними правилами. Комп'ютери при пересилці інформації використовують цифрові адреси, а користувачі в роботі з Інтернетом використовують в основному імена.

Цифрові адреси в Інтернеті складаються з чотирьох чисел, кожне з яких не перевищує двохсот п'ятдесяти шести. При записи числа відокремлюються точками, наприклад: 195.63.77.21. Такий спосіб нумерації дозволяє мати в мережі більше чотирьох мільярдів комп'ютерів.

Спочатку в мережі Internet застосовувалися IP - номери, але коли кількість комп'ютерів в мережі стало більше ніж 1000, то був прийнятий метод зв'язку імен і IP - номерів, який називається сервер імені домена (Domain Name Server, DNS). Сервер DNS підтримує список імен локальних мереж і комп'ютерів і відповідних їм IP - номерів.

Наразі в Інтернеті застосовується так звана доменна система імен. Кожен рівень в такій системі називається доменом. Типове ім'я домена складається з декількох частин, розташованих в певному порядку і між якими ставлять крапку. Домени відокремлюються один від одного крапками, наприклад: www.asbit.nuczu.edu.ua.

Доменна система імен використовує принцип послідовних уточнень також як і в звичайних поштових адресах - країна, місто, вулиця і будинок, в який слід доставити лист. Домен верхнього рівня розташовується в імені правіше, а домен нижнього рівня - лівіше. У нашому прикладі домени верхнього рівня ua вказують на те, що мова йде про приналежність сайту до української (ua) частини Інтернету. Але в Україні безліч користувачів Інтернету, і наступний рівень визначає професійну діяльність організації, якій належить дана адреса (edu - освіта). Діючи за аналогією існує багато організацій, що надають освітні послуги, що приводить до конкретного університету (НУЦЗУ). Всі комп'ютери, підключені до Інтернету в університеті, об'єднуються в групу, що має таку адресу.

Ім'я окремого комп'ютера або мережі кожна компанія вибирає для себе самостійно, а потім реєструє його в тій організації Інтернету, яка забезпечує підключення. Це ім'я в межах домена верхнього рівня має бути унікальним. В імені може бути будь-яке число доменів, але найчастіше використовуються імена з кількістю доменів від трьох до п'яти.

Доменна система освіти адрес гарантує, що у всьому Інтернеті більше не знайдеться іншого комп'ютера з такою ж адресою. Для доменів нижніх рівнів можна використовувати будь-які адреси, але для доменів самого верхнього рівня існує угода.

В системі адрес Інтернету прийняті домени, представлені географічними регіонами. Вони мають ім'я, що складається з двох літер, наприклад:

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

- .ru — Росія
- .ua — Україна
- .se — Швеція
- .ro — Румунія
- .lt — Литва
- .it — Італія
- .az — Азербайджан

Існують і домени, розділені по тематичних ознаках, наприклад:

- .com - для комерційних сайтів;
- .edu - для освітніх сайтів;
- .gov - для сайтів державних організацій;
- .mil - для військових організацій;
- .net - для сайтів, чия діяльність пов'язана з Мережею;
- .org - для некомерційних організацій;
- .int - для міжнародних організацій.

Останнім часом додані нові зони, наприклад: biz, info, in, .cn і так далі

При роботі в Internet використовуються не доменні імена, а універсальні покажчики ресурсів, звані URL (Universal Resource Locator). URL - це адреса будь-якого ресурсу (документа, файлу) в Internet, він вказує, за допомогою якого протоколу слід до нього звертатися, яку програму слід запустити на сервері і до якого конкретного файлу слід звернутися на сервері.

URL- адреса складається з трьох частин:

1) назва протоколу, який визначає тип доступу. Після назви йдуть двокрапка ":" і два знаки "//" (дві коси риски) – **http://**
2) назва сервісу та DNS-ім'я комп'ютера: – **www.asbit.nuczu.edu.ua**
3) повне ім'я файла. Як роздільник у повному імені використовується знак "/" – **Files/lesson1.html**

Повна URL- адреса **http://www.asbit.nuczu.edu.ua/Files/lesson1.html**

Сервіси мережі інтернет

Як показує практика, при реалізації методів роботи з об'єктами того чи іншого мережевого інформаційного ресурсу може виявитися, що однієї пари програм (клієнта і сервера) недостатньо. Така ситуація пояснюється відмінністю алгоритмів обробки запитів різного типу. Тому може знадобитися розробка цілого комплексу програмного забезпечення і протоколів його взаємодії. Класичним прикладом є робота електронної пошти, коли відправка поштових повідомлень забезпечується за допомогою серверів і, відповідно, протоколу одного типу, а доставка повідомлень одержувачу - за допомогою серверів і протоколів інших типів. Комплекс програм і протоколів, що забезпечує користувачеві можливість роботи з інформаційними об'єктами будь-якого інформаційного ресурсу або декількох ресурсів, є однією зі складових такого поняття як **мережевий сервіс** або **служба**.

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Термін сервіс, який використовується по відношенню до Інтернет, походить від англійського service, що в перекладі означає служба. Але в даний час спостерігаються дві тенденції - використання перекладених термінів (в нашому випадку - це служба) і використання транслітерованих термінів (сервіс).

Кожен сервіс являє собою сукупність наступних компонентів: інформаційного ресурсу - сукупності інформаційних об'єктів, наприклад, документів, які використовуються людиною; програми-клієнта, яка встановлюється на комп'ютерах користувачів, і забезпечує можливість роботи з інформаційними об'єктами, розташованими на різних комп'ютерах мережі; набір програм-серверів, які здійснюють обробку запитів клієнтського програмного забезпечення і керуючих інформаційними об'єктами; набір протоколів взаємодії клієнтського програмного забезпечення з серверами.

Постійний розвиток інформаційних технологій призводить до появи різноманітних інформаційних ресурсів, що відрізняються один від одного формами уявлення і методами обробки складових їх інформаційних об'єктів. Тому в даний час в Інтернет існує досить велика кількість сервісів, що забезпечують роботу з усім спектром ресурсів. Найбільш відомими серед них.

Електронна пошта (E-mail) - технологія і надані нею послуги з пересилання та отримання електронних повідомлень (званих «листів» чи «електронніх листів») по розподіленої (у тому числі глобальної) комп'ютерної мережі.

В електронній пошті адреса складається з трьох частин: імені одержувача (поштової скриньки); знака “@” (эт); адреси поштового серверу, до якого приєднано одержувача. Наприклад: **Ikt@nuczu.edu.ua**.

Перевагами електронної пошти є:

- адреси виду username @ domainname, що легко запам'ятовувати;
- можливість передачі як простого тексту, так і форматованого, а також довільних файлів;
- незалежність серверів (в загальному випадку вони звертаються один до одного безпосередньо);
- досить висока надійність доставки повідомлення ;
- простота використання людиною та програмами.

Недоліки електронної пошти:

- наявність такого явища, як **спам** (масові рекламні та вірусні розсилки);
- теоретична неможливість гарантованої доставки конкретного листа;
- можливі затримки доставки повідомлення (до декількох діб);
- обмеження на розмір одного повідомлення і на загальний розмір повідомлень в поштовій скриньці (персональні для користувачів).

Електронна пошта працює за наступними протоколами

- SMTP (Simple Mail Transfer Protocol - простий протокол передачі пошти) - це мережевий протокол, призначений для **передачі** електронної пошти в мережах TCP / IP.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

- POP3 (Post Office Protocol Version 3 - протокол поштового відділення, версія 3) використовується поштовим клієнтом для **отримання** електронної пошти з сервера.
- IMAP (. Internet Message Access Protocol - Протокол доступу до електронної пошти Інтернету) - протокол прикладного рівня для доступу до електронної пошти.

Існує дві різні моделі роботи з поштою.

Концепція поштової скриньки. У концепції поштової скриньки пошта на сервері зберігається тимчасово, в обмеженому обсязі (аналогічно поштової скриньки для паперової пошти), а користувач періодично звертається до ящика і «забирає» листа (тобто поштовий клієнт завантажує копію листа до себе і видаляє оригінал з поштової скриньки) . На підставі цієї концепції діє протокол POP3.

Концепція постійного зберігання (сховища пошти). Концепція постійного зберігання має на увазі, що вся кореспонденція, пов'язана з поштовою скринькою, зберігається на сервері, а користувач звертається до сховища (іноді його за традицією також називають «поштовою скринькою») для перегляду кореспонденції і написання нових листів. На цьому принципі діє протокол IMAP і більшість веб-інтерфейсів безкоштовних поштових служб.

Система Usenet (Мережеві новини, Телеконференції) засіб спілкування груп людей зі схожими інтересами. Мережеві новини передають повідомлення "від одного до багатьох". Надіслане повідомлення поширюється по всій мережі, досягаючи за досить короткі терміни всіх учасників телеконференцій Usenet в усьому світі.

Загальна кількість повідомлень, що надходять у телеконференції щодня, складає близько мільйона.

Сервіс Списки розсилки (Maillists) - працюють виключно через електронну пошту. Ідея роботи списку розсилки полягає в тому, що існує якийсь адрес електронної пошти який, насправді, є загальною адресою багатьох передплатників цього списку розсилки. Цей вид служби дозволяє організувати обговорення окремих питань, не призначених для широкого кола осіб.

Сервіс FTP (File Transfer Protocol) - забезпечує прийом і передачу файлів у файлових архівах, до гігантських обсягів інформації в Інтернет. Сервер FTP можна налаштовується таким чином, що з'єднатися з ним можна не тільки під своїм ім'ям і паролем, але й під умовним ім'ям anonymous - анонім. Працює за протоколом ftp.

Система гіпермедіа WWW (World Wide Web - всесвітня павутина) - найбільш популярний і цікавий сервіс Інтернет сьогодні, найпопулярніше і зручний засіб роботи з інформацією. Більше половини потоку даних Інтернет припадає на частку WWW. Уся інформація у WWW розміщується на Web-вузлах - це група тематично об'єднаних Web-сторінок. Працює за протоколом http.

Сервіс IRC- (Internet Relay Chat, розмови через Інтернет) відноситься до інтерактивних сервісів, призначених для спілкування людей через Інтернет. Цей сервіс дозволяє різним користувачам в режимі реального часу спілкуватися між собою обміном текстовими повідомленнями

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Сервіс MUD (Multi User Dungeon) - багатокористувацькі ігри в Інтернет, зазвичай рольового або казкового характеру.

Сервіс MOO (Object-Oriented MUD) - об'єктно-орієнтований багатокористувацький світ. У віртуальному світі MOO відсутня гра, але зате створюються об'єкти і визначаються їх властивості та зв'язки, що може мати деяке застосування в освітніх цілях.

Сервіс ICQ ("I Seek You" - Я шукаю Вас) Цей сервіс дозволяє користувачами, присутнім в даний момент в Інтернеті спілкуватися в реальному часі. За допомогою ICQ можна посилати повідомлення і файли, грати, і т.д.

Сервіс Internet-телефонія (IT) Це один з наймолодших сервісів Інтернету. Інтернет аналог звичайного телефону. Низька якість зв'язку окупається найголовнішим - вартістю розмови, в порівнянні з міжнародними телефонними переговорами Internet-телефон - практично безкоштовне засіб зв'язку.

Сервіс Telnet - найстаріший сервіс Інтернету, за допомогою якого може здійснюватися віддалений доступ до іншої обчислювальної системи. При цьому комп'ютер, що підключається, перетворюється на термінал віддаленого комп'ютера.

Питання для самоконтролю

1. Що таке комп'ютерна мережа?
2. По яких ознаках класифікують комп'ютерні мережі?
3. Які параметри характеризують процес передачі інформації з мережі?
4. Що являє собою середовище передачі? Які існують кабелі, і які їх характеристики?
5. Що таке та для чого створюються КМ? Які можливості надає КМ?
6. Що таке сервер? робоча станція?
7. По якому принципі локальні мережі ділять на однорангові й мережі з виділеним сервером?
8. Які методи доступу від комп'ютера до комп'ютера використовуються в локальних мережах?
9. Які бувають топології КМ? Їхні достоїнства й недоліки.
10. Як об'єднати кілька локальних мереж?
11. Які конфігурації КМ використовуються в вашому комп'ютерному класі?
12. Що означають поняття «клієнт» і «сервер»?
13. Що таке протокол? Стек протоколів?
14. Яка мережна ОС використовується в КМ, у якій Ви працюєте?
15. Які ресурси можна надати в спільне використання? Як це зробити?
16. Як скасувати спільне використання своїх ресурсів?
17. Які апаратні пристрої називаються мережними ресурсами? Як по виду значка визначити, яким є ресурс?
18. Що являє собою поняття електрона пошта?
19. Яка система формування адреси у Інтернет є найбільш поширеною?
20. Назвіть складові електронної адреси.
21. Що означає поняття ДОМЕН, ПІДДОМЕН?
22. Як напрями читання доменного імені? Який піддомен є піддоменом верхнього рівня?
23. Які сервіси мережі інтернет Ви знаєте?

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Мова HTML.
Основні принципи створення WEB-сторінок.

Зміст

Мова HTML. Основні принципи створення WEB-сторінок. 114
1. *Загальне представлення про мову гіпертекстової розмітки HTML.* 114
2. *Структура HTML-документу.* 116
3. *Опис палітри кольорів в HTML-документі.* 118
4. *Основні теги форматування тексту* 119
Заголовки тексту в HTML 120
Установка абзацу 120
Форматування тексту 121
Додаткові теги з оформлення тексту 121
5. *Вставка зображень та гіперпосилань на Web-сторінку* 123
Зображення на Web-сторінках 123
Вставка посилань 124
6. *Спеціальні символи та крос-браузерність* 126
Питання для самоконтролю 127

1. *Загальне представлення про мову гіпертекстової розмітки HTML.*

HTML - це мова розмітки, який представляє прості правила оформлення і компактний набір структурних і семантичних елементів розмітки (тегів). HTML дозволяє описувати спосіб представлення логічних частин документа (заголовки, абзаци, списки і т.д.) і створювати Web-сторінки різної складності.

Розмітка – використання спеціальних кодів, які легко відокремлюються від значеннєвого змісту документа. Такі коди називаються **тегами** і вставляються безпосередньо в текст документа.

Спочатку мова **HTML** (HyperText Markup Language) була задумана та створена як засіб структурування та форматування документів без прив'язки до засобів відображення. В ідеалі, гіпертекстовий документ повинен однаково виглядати на різних пристроях (монітор ПЕОМ, екран ПДА або мобільного телефону, принтер, медіа-проектор тощо).

Розробкою специфікацій мови HTML і затвердженням їх в якості офіційних стандартів займається Консорціум всесвітньої павутини (The World Wide Web Consortium – W3C). Крім W3C, в розвитку мови беруть участь IT-компанії і спільнота розробників.

- Офіційної специфікації HTML 1.0 не існує. До 1995 року існувало безліч неофіційних вказівок для використання HTML, що з'явилися в ході браузерних війн.
- RFC 1866 - HTML 2.0, схвалений як офіційний стандарт 22 вересня 1995 року;
- HTML 3 (березень 1996) - не знайшла підтримки у розробників
- HTML 3.2 - 14 грудень 1997 року;
- HTML 4.0 - 18 січень 1997 роки (багато успадкованих елементів були відзначені як застарілі та нерекомендовані до використання (англ. Deprecated));

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

- HTML 4.01 - 24 січень 1999 року (версія включала малопомітні, але істотні зміни в порівнянні з попередньою);
- ISO / IEC 15445: 2000 (ISO HTML, заснований на HTML 4.01 Strict) - 15 травня 2000 року.
- HTML 5 - розроблений і прийнятий W3C спільно з робочою групою по вебу, гіпертексту, застосункам та технологіям, англ. Web Hypertext Application Technology Working Group) WHATWG.

HTML не є мовою програмування, але Web-сторінки можуть містити вбудовані елементи або файли програм на скриптових мовах (в першу чергу Javascript) і програми-аплети на мові Java.

Web-сторінка (HTML-документ) - **текстовий** документ, який являє собою сукупність інформаційного вмісту та спеціальних засобів мови HTML - **тегів розмітки**, які визначають структуру і зовнішній вигляд документа та показують браузеру, як слід відображати Web-сторінку при її перегляді. **Теги розмітки** дозволяють:

- задавати **оформлення** тексту: вирівнювати абзаци, встановлювати таблиці, змінювати розмір та накреслення шрифту;
- впроваджувати **графічні** і **мультімедійні** об'єкти: звук, відеокліпи тощо;
- вставляти **гіпертекстові** посилання.

Web-сторінка має розширення .**html**. Тому Web-сторінки ще називають HTML-документами.

Тег (Html-теги, теги розмітки) - керуюча символьна послідовність, яка задає спосіб відображення гіпертекстової інформації.

HTML-тег складається з імені, за яким може слідувати необов'язковий список атрибутів. Весь тег (разом з атрибутами) містяться у кутових дужках: **< … >**:

```
<ТЕГ [атрибути]>
```

Теги бувають

- **непарними** – впливають на весь документ або визначають разовий ефект у місці своєї появи. Структура: **<ТЕГ>** ;
- **парними** (контейнерними) – складаються з початкового та кінцевого тегів, між якими розміщується інформація. Ім'я кінцевого тегу збігається з ім'ям початкового, але перед ім'ям кінцевого тегу ставиться коса риска **/**. Структура: **<ТЕГ>**…**</ТЕГ>**

Теги можуть містити **атрибути**, що характеризують відображення інформації усередині тегу. **Атрибути** – додаткові ключові слова, що відокремлюються від назви тегу та інших атрибутів пробілами.

Атрибути вказують в початковому тегу, кілька атрибутів поділяють одним або декількома пропусками, табуляцією або символами кінця рядка. Деякі **атрибути** вимагають указівки їхнього значення. Значення **атрибута**, якщо таке є, вводять зі знаку **дорівнює**. Якщо значення атрибута - одне слово або число, то його можна просто вказати після знака рівності, не виділяючи додатково. Всі інші значення необхідно брати в **лапки**, особливо якщо вони містять кілька розділених пробілами слів. Порядок запису атрибутів у тегу не важливий. Структура:

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

```
<ТЕГ атрибут1 = ″значення1″ атрибут2 = ″значення2″
атрибут3 = ′значення3′ атрибут4 = значення4 … > … </ТЕГ>
```

Незважаючи на необов'язковість лапок, їх все ж варто завжди використовувати.

Атрибути можуть бути **обов'язковими** і **необов'язковими**. Необов'язкові атрибути можуть бути опущені, тоді для тегу застосовується значення цього атрибута за замовчуванням. Якщо не вказано обов'язковий атрибут, то вміст тегу швидше за все буде відображено неправильно.

Кінцеві теги **ніколи** не містять атрибутів. Деякі теги не мають кінцевого елемента. Регістр символів для тегів не має значення.

Частина HTML-документу між відкриваючим і закриваючим тегом називається **елементом мови HTML** або контейнером, що містить дані. Web-сторінка являє собою набір елементів. **Елементи мови HTML** поділяють на

- **блокові** – частина тексту рівня абзацу та усі елементи після яких текст починається з нового рядка;
- **текстові** – окремі фрази і більш дрібні частини тексту: слова та навіть окремі букви.

Елементи мови HTML (Теги) можуть бути вкладені друг в друга, при цьому форматування внутрішнього тегу має перевагу перед зовнішнім. При використанні вкладених тегів їх потрібно закривати, починаючи з самого останнього і рухаючись до першого та дотримуючись наступних правил:

- елементи не повинні перетинатися;
- блокові елементи можуть містити вкладені блокові і текстові елементи;
- текстові елементи можуть містити тільки вкладені текстові елементи.

Для того щоб створити HTML-документ вистачить простого додатка, яке вміє редагувати текстовий вміст файлу (наприклад **Блокнот**). Але так писати код складно, довго, і в процесі розробки допускається безліч помилок, які згодом важко знайти. Виправити ці недоліки допомагають сучасні редактори HTML–коду. Існують як платні редактори, так і безкоштовні. На сьогоднішній день найбільш популярними є:

- Notepad++;
- Sublime Text 3;
- Komodo Edit;
- Aptana;
- Araneae;
- jEdit;
- Visual Studio Code.

2. *Структура HTML-документу.*

Всі документи HTML мають одну і ту ж **структуру**, яка визначається фіксованим набором тегів структури. Структура HTML-документа (рис. 6.1) досить проста:

- Опис документа починається з вказівки його типу (секція **DOCTYPE**). Наявність секції **DOCTYPE** дозволяє вказати браузеру, який тип документа йому належить відображати, тобто, які вимоги потрібно виконувати при обробці гіпертексту.

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

- Текст документа розміщується в тегу **<HTML>**. Текст документа складається з заголовка і тіла, які виділяються відповідно тегами **<HEAD>** і **<BODY>**.

- У заголовку (**<HEAD>**) вказують назву HTML-документа і інші атрибути, які браузер буде використовувати при відображенні документа. Блок призначений для розміщення метаінформації, яка описує HTML-документ у цілому.

- Тіло документа (**<BODY>**) - це та частина, в яку й поміщається вміст HTML-документа. Тіло включає призначений для відображення текст і керуючу розмітку документа (**теги**), які використовуються браузером. Блок містить те, що потрібно показати користувачеві: текст, зображення, впроваджені об'єкти тощо.

Секція **DOCTYPE** вказує браузеру тип документа і версію мови розмітки, яка використовується. Тут також вказується назва і область видимості опису цієї мови і адреса файлу DTD (**Document Type Definition**).

Приклади DOCTYPE:

- <! DOCTYPE HTML PUBLIC "- // W3C // DTD HTML 4.01 Frameset // EN" "http://www.w3.org/TR/html4/frameset.dtd"> – Гіпертекстовий документ в форматі HTML 4.01, що містить фрейми.
- <! DOCTYPE HTML PUBLIC "- // W3C // DTD HTML 4.01 // EN" "http://www.w3.org/TR/html4/strict.dtd"> – Гіпертекстовий документ в форматі HTML 4.01 із суворим синтаксисом (тобто не використані застарілі і не рекомендовані теги).
- <! DOCTYPE HTML PUBLIC "- // W3C // DTD HTML 4.01 Transitional // EN" "http://www.w3.org/TR/html4/loose.dtd"> – Гіпертекстовий документ в форматі HTML 4.01 з нестрогим («перехідним») синтаксисом (тобто використані застарілі або які не рекомендовані теги і атрибути).
- <! DOCTYPE HTML> – Ще не стандартизоване оголошення для документів HTML5.

Рис. 6.1 Структура HTML-документа

Стандарт вимагає, щоб секція DOCTYPE була присутня в документі, тому що це дозволяє прискорити і поліпшити обробку гіпертексту. Це досягається за рахунок того, що браузер може не робити припущень про те, як інтерпретувати теги, а звіритися зі стандартним визначенням (файлом DTD). Детальний опис **DOCTYPE** – міститься на сайті Консорціуму W3C.

У розділі заголовків (**<HEAD>**) вказують так звану метаінформацію (**Мета-теги HTML**) – це елемент розмітки HTML, що описує властивості документа у цілому. Пошукові системи звертаються до метатегів для отримання опису сайту, ключових слів та інших даних. Дозволено використовувати необмежене число метатегів, всі вони повинні знаходитися між **<HEAD>** та **</HEAD>**. Призначення мета-тегів визначається набором його атрибутів, що є ключовими словами **content**, **name** або **http-equiv**, які задаються в тезі **<META>.** Метатеги не є обов'язковими елементами, але можуть бути дуже корисні.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Приклади опису метаданих:

<HEAD>

<META name = "author" content = "рядок"> - автор Web-документу

<META name = "date" content = "дата"> - дата останнього зміни Web-сторінки

<META name = "copyright" content = "рядок"> - авторські права

<META name = "keywords" content = "рядок"> - список ключових слів

<META name = "description" content = "рядок"> - короткий опис (реферат)

<META name = "ROBOTS" content = "NOINDEX, NOFOLLOW"> - заборона на індексування

<META http- equiv = "content-type" content = "text / html; charset = UTF-8"> - тип і кодування

<META http-equiv = "expires" content = "число"> - управління хешуванням

<META http-equiv = "refresh" content = "число; URL = адреса"> - перенаправлення сторінки на іншу адресу

</HEAD>

Також у розділі заголовків повинно міститися загальна назва документу (текстовий рядок), що буде відображатися в заголовку вікна браузера. Назву поміщають у парний тег **<TITLE>**...**</TITLE>**. Розділ **<TITLE>**...**</TITLE>** є єдиним обов'язковим структурним елементом. Проте, при створенні HTML-документа опускати інші структурні теги не рекомендується.

Елементи, що знаходяться всередині заголовка (крім **<TITLE>**) не відображаються на екрані під час перегляду документа.

3. Опис палітри кольорів в HTML-документі.

Таблиця (палітра) кольорів HTML надає можливість самостійно підібрати потрібний тон. Існує кілька основних способів представлення кольорів HTML-документу в Інтернеті:

- За допомогою **ключових слів**, наприклад **green** або **black**. Щоб уникнути випадків, коли вказане ключове слово «не розуміється» браузером, слід використовувати лише невеликий набір основних кольорів, які використовуються у всіх браузерах.

- **aqua** (морська хвиля, cyan)
- **black** (чорний)
- **blue** (синій)
- **fuchsia** (фуксія, magenta)
- **gray** (сірий)
- **green** (зелений)
- **lime** (лайм, lightgreen)
- **maroon** (темно-бордовий)
- **navy** (темно-синій)
- **olive** (оливковий)
- **purple** (пурпурний)
- **red** (червоний)
- **silver** (срібний)
- **teal** (синьо-зелений)
- **white** (білий)
- **yellow** (жовтий)

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

- У **форматі HEX**

  Подання кольору у **форматі HEX** здійснюється за допомогою трьох пар символьних значень в шістнадцятковій системі числення. Наприклад: **#ff00b3**, де перша пара чисел відповідає за червоний, друга за зелений, а третя за синій колір. Можливо також коротке представлення кольору у вигляді **#ABC**, що буде інтерпретовано як **#AABBCC**.

- У **форматі RGB**

  У **вигляді RGB** (*, *, *), де «*» - числа від 0 до 255, що позначає кількість відповідного тону (червоний, зелений, синій) в вихідному кольорі. Наприклад RGB (200,100,255). Можливий також варіант RGBA (*, *, *, *), де перші 3 «*» - компоненти кольору, задаються в діапазоні 0 до 255, а остання «*» - рівень непрозорості (альфа-канал), що задається дрібним числом від 0 до 1.

- У **форматі HSV**

  У **вигляді HSV** (**Hue**, **Saturation**, **Value** - тон, насиченість, значення) - колірна модель, в якій в якості координат є: **Hue** - колірний тон, може варіюватися в межах від 0 ° до 360 °. **Saturation** - насиченість, варіюється в межах 0-100 або 0-1. Чим більше цей параметр, тим «чистіше» колір, тому цей параметр іноді називають чистотою кольору. А чим ближче цей параметр до нуля, тим ближче колір до нейтрального сірого. **Value** (значення кольору) - задає яскравість, значення так само може варіюватися в межах 0-100 або 0-1.

> В епоху 256-кольорових відеокарт широко застосовувалася так звана «безпечна палітра Netscape» - однорідна палітра 6×6×6. У truecolor у безпечних кольорів все три компонента - R, G і B - дорівнюють 00, 33, 66, 99, CC і FF. Браузер Netscape гарантував, що ці кольори будуть відображені без наближення або растрування.
>
> Іншими словами, 216 кольорів стандартної 16-кольорової палітри і палітри 6×6×6 були найкращими для кольору зображення і фону.
>
> До кінця 90-х років, з повним зникненням 8-бітного кольору з робочих столів ОС, ця вимога зникло. Проте, Adobe Photoshop і понині оснащений опцією «Тільки Web-кольори» (англ. Only Web Colors).

4. Основні теги форматування тексту

Як було відмічено раніше, тег **<BODY>…</BODY>** містить тіло документа, що відображається браузером на екрані. Для того щоб створити найпростішу Web-сторінку, відкриємо блокнот та в ньому введено код найпростішої HTML-сторінки:

```
<HTML>

  <HEAD>
    <TITLE> Мій перший сайт </TITLE>
  </HEAD>
<BODY>
  Привіт усім! А це мій перший сайт!
</BODY>

</HTML>
```

Збережемо цей код (**Файл – Зберегти як**). При цьому у полі ім'я файлу вкажемо *назву_стрінки.html*, а в полі тип файлу вкажемо *Всі файли* та натиснемо **Зберегти**.

Якщо відкрити створений файл за допомогою браузера, то ми побачимо текст який міститься між тегами **<BODY>**...**</BODY>**. Окрім цього безпосередньо тег **<BODY>** має кілька атрибутів. Основні з них:

bgcolor – колір фону документа.

background – указує шлях до графічного файлу, що буде використаний як тіло документа.

Наприклад,

```
<BODY Bgcolor='yellow'> або <BODY Background='fon.jpg'>
```

Якщо параметри кольору для даного тегу не зазначені, то тіло документа буде білим.

Заголовки тексту в HTML

Заголовки мають важливе значення в структуруванні HTML-документа і є одними з найуживаніших у мові розмітки тексту. Їх призначення – показати структуру документа.

HTML має шість рівнями заголовків від **<H1>** до **<H6>**.

```
<H1> ... </ H1>
<H2> ... </ H2>
... ... ... ...
<H6> ... </ H6>
```

Самий верхній рівень **<H1>**, нижній - **<H6>**.

При відображенні на екрані ці елементи показуються за допомогою шрифтів різного розміру. Зазвичай дані теги мають атрибут **align** - вирівнювання тексту по краю вікна браузера (по лівому краю, по центру, по правому, по ширині)

```
Align = "left | center | right | justify"
<H1 Align = "center"> ... </H1>
```

За замовчуванням **left** (по лівому краю)

Установка абзацу

HTML припускає, що автор документа нічого не знає про комп'ютер користувача. Користувач має право встановити будь-який розмір вікна і користуватися будь-яким з наявних у нього шрифтів. Це означає, що місце перенесення в рядку визначається тільки програмою перегляду й установками кінцевого користувача.

Для розбиття тексту на абзаци використовується тег **<P> ... </P>** .

Перед першим рядком абзацу розміщують тег **<P>**, в кінці останньої - **</P>**.

Даний тег має велике значення при стильовому оформленні документа. Тег абзацу може мати атрибут **align**.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Для того, щоб перейти на новий рядок в будь-якому місці поточного рядка, в HTML існує тег розриву рядка **
** Цей тег змушує програму перегляду виводити символи, що стоять після нього, з нового рядка. На відміну від тегу абзацу, тег **
** не додає пустий рядок і не має закриваючого тегу.

Ще одним методом розділення документа є використання розділових ліній. Для проведення такої лінії в тілі документа використовується тег **<HR>**. Цей тег відноситься до блокових елементів, лінія завжди починається з нового рядка. Тег може використовуватися з наступними атрибутами:

Align = "left | center | right | justify" – вирівнювання, за замовчуванням left (по лівому краю),

Width = "число" – ширина елемента (в пікселях, відсотках тощо),

Size = "число" – розмір елемента,

Color = "Значення" - колір лінії.

Форматування тексту

Однією з важливих можливостей HTML є повний контроль вигляду створюваного документа, у тому числі керування шрифтами. На практиці тип виведення шрифтів сильно залежить від програмно-апаратного забезпечення комп'ютера користувача. Однак для HTML-документів проблема значно полегшується тим, що на всіх машинах мережі повинен бути встановлений стандартний набір шрифтів. Зазвичай це **Verdana**, **Arial**, **Times New Roman**.

Для роботи зі шрифтами використовується тег **<FONT> ... </FONT>.** Тег **<FONT>** має наступні атрибути:

Size = "число" - служить для вказівки розміру шрифту в умовних одиницях від 1 до 7. Прийнято вважати, що розмір «нормального» шрифту відповідає числу 3.

```
<FONT Size='4'> Розмір шрифту дорівнює 4 </FONT>
```

Color = "Значення" – встановлює колір шрифту, який може бути вказаний як у форматі **HEX**, так і заданий стандартним ім'ям.

```
<FONT Color='yellow'> Колір шрифту - жовтий </FONT>
```

Face = "Тип шрифту" – встановлює тип шрифту,

```
<FONT Face="arial"> Тип шрифту - Arial </FONT>
```

Хоча цей тег до сих пір підтримується всіма браузерами, він вважається застарілим і від його використання рекомендується відмовитися на користь стилів.

Додаткові теги з оформлення тексту

<B> </B> - встановлює жирне накреслення шрифту.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

<I> </I> - встановлює курсивне накреслення шрифту.

<U> </U> - додає підкреслення до тексту.

<EM> </EM> - призначений для акцентування тексту. Браузери відображають такий текст курсивним шрифтом.

<S> </S> перекреслює текст.

<TT> </TT> - відображає текст моноширінним текстом. Виключений з HTML5.

**** - відображає шрифт у вигляді верхнього індексу. Шрифт при цьому відображається вище базової лінії тексту і зменшеного розміру.

**** - відображає шрифт у вигляді нижнього індексу. Текст при цьому розташовується нижче базової лінії інших символів рядка і зменшеного розміру.

<STRONG> </STRONG> - призначений для акцентування тексту. Браузери відображають такий текст жирним шрифтом.

<SMALL> </SMALL> - зменшує розмір шрифту на одиницю в порівнянні зі звичайним текстом. Допускається застосування вкладених тегів < small >, при цьому розмір шрифту буде менше на 1 з кожним вкладеним рівнем, але не може бути менше, ніж 1.

<BIG> </BIG> - збільшує розмір шрифту на одиницю в порівнянні зі звичайним текстом. Допускається застосування вкладених тегів <big>, при цьому розмір шрифту буде більше з кожним рівнем.

<Q> </Q> - використовується для виділення в тексті цитат. Вміст контейнера автоматично відображається в браузері в лапках.

<BLOCKQUOTE> </BLOCKQUOTE> - призначений для виділення довгих цитат всередині документа. Текст, позначений цим тегом, традиційно відображається як вирівняний блок з відступами зліва і справа (приблизно по 40 пікселів), а також з відбиття зверху і знизу.

<PRE> </PRE> - визначає блок попередньо форматованого тексту. Такий текст відображається зазвичай моноширінним шрифтом і з усіма пробілами між словами. За замовчуванням, будь-яку кількість прогалин йдуть в коді поспіль, на Web-сторінці показується як один. Тег <pre> дозволяє обійти цю особливість і відображати текст як потрібно розробнику.

<CITE> </CITE> - позначає текст як цитату або виноску на інший матеріал. Таке виділення зручно для поділу коду HTML на структурні елементи. Браузери зазвичай встановлюють текст всередині контейнера < cite > курсивом.

<ABBR> </ABBR> - вказує, що послідовність символів є абревіатурою. За допомогою атрибута title дається розшифровка скорочення, що дозволяє розуміти абревіатуру тим людям, які з нею не знайомі. Крім того, пошукові системи індексують повнотекстовий варіант скорочення, що може використовуватися для підвищення рейтингу документа. За замовчуванням, текст укладений в контейнері < abbr > підкреслюється пунктирною лінією.

<OL> </OL> - Тег встановлює нумерований список. Кожен елемент списку повинен починатися з тега **<LI>**.

<UL> </UL> - Тег встановлює маркований список. Кожен елемент списку повинен починатися з тега **<LI>**.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

<LI> </LI> - Тег визначає окремий елемент списку. Зовнішній тег **<UL>** або **<OL>** встановлює тип списку - маркований або нумерований.

5. *Вставка зображень та гіперпосилань на Web-сторінку*

Зображення на Web-сторінках

Існує досить багато форматів графічних документів: GIF, BMP, JPEG (JPG), PNG, WMF тощо. Однак при оформленні HTML-документа краще обмежитися трьома: GIF, JPEG або PNG, оскільки їх розуміє більшість сучасних програм перегляду.

Щоб вставити зображення на Web-сторінку використовується тег **<IMG>**. Цей тег одинарний, він не має закриває тегу. Основним атрибутом є **Src = "адреса"** - адреса (URL) джерела даних (наприклад картинки або скрипта).

Якщо малюнок знаходиться у тому ж каталозі, що і файл Web-сторінки, треба просто написати ім'я файлу. Можна вказати також відносну адресу файлу або URL. Наприклад:

```
<IMG Src = "image1.jpg">
<IMG Src = "images / image2.jpg">
<IMG Src = "http: //www.nuczu.edu.ua/img/image3.jpg">
```

Малюнок сприймається браузером як «велика буква» в тексті. Тому важливо вказати правильне вирівнювання в атрибуті **Align** тега **<IMG>**. Існує наступні варіанти вирівнювання:

```
Align = "left | bottom | right | middle | top "
```

- **top** - по верхній межі поточного рядка
- **bottom** - по нижній межі поточного рядка (такий тип вирівнювання встановлюється за умовчанням)
- **middle** - по середині поточного рядка
- **left** - по лівій межі абзацу з обтіканням текстом
- **right** - по правій межі абзацу з обтіканням текстом

Тег **<IMG>** має і інші атрибути, найбільш важливі з яких наступні:

Width = "число" – ширина малюнка в пікселях,
Height = "число" – висота малюнка в пікселях
Vspace = "число" – вертикальний відступ в пікселях
Hspace = "число" – горизонтальний відступ в пікселях.
Alt = "текст" – альтернативний текстовий опис
Border = "число" – ширина рамки навколо малюнка

Якщо розміри малюнка не вказані, вони визначаються з файлу (для цього, звичайно, треба завантажити файл з Інтернету). Якщо зазначені розміри не співпадають з розмірами малюнка в файлі, він буде стиснутий або розтягнутий.

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Рекомендується завжди вказувати розміри малюнка, оскільки це дозволяє браузеру відразу визначити розміри області, без завантаження рисунку.

Опис рисунку в атрибуті **Alt** з'являється у вигляді підказки при наведенні курсору миші на відповідний рисунок. Крім того, якщо у користувача відключений перегляд рисунків (для прискорення роботи), на місці рисунка виводиться текст, заданий в атрибуті **Alt**. Цей атрибут, так само, як і розміри рисунку, обов'язково вказується на всіх серйозних сайтах.

```
    <IMG Src = "photo.jpg" Alt = "Моя фотографія" Width =
"50" Height = "80" Border = "0">
```

Вставка посилань

Гіпертекстовий документ - це документ, що містить **гіперпосилання** на інші документи, що дозволяють за допомогою натискання кнопки миші швидко переміщатися від одного документа до іншого.

Гіперпосилання - це особливим чином позначений фрагмент Web-сторінки (текст, зображення та ін.), Який пов'язаний з іншим документом. Для вказівки гіперпосилань використовується тег **<A>** з яким можуть бути пов’язані наступні атрибути:

Accesskey - активує посилання за допомогою комбінації клавіш.

Href - задає адресу документа, на який слід перейти.

Hreflang - ідентифікує мову тексту за посилання.

Name - встановлює ім’я якоря Всередині документа (в HTML5 слід використовуват **Id**).

Tabindex - візначає послідовність переходу між посиланнями при натіканні на кнопку **Tab**.

Target - тип вікна, в якому браузер буде завантажувати документ. Може приймати значення **_Blank** - завантажує сторінку в новому вікні браузера. **_Self** - завантажує сторінку в поточному вікні (значення за замовчуванню). **_Parent** - завантажує сторінку у фреймі-батьку, якщо фреймів немає, то цей параметр працює як **_Self**. **_Top** - скасовує всі фрейми и завантажує сторінку в повному вікні браузера, якщо фреймів немає, то цей параметр працює як **_Self**.

Будь-яке посилання складається з двох частин: **покажчик** - це те, що відображається на Web-сторінці та **адресна частина** - інструкція браузеру, тобто адреса документа, який буде завантажуватися при натисканні на покажчик посилання.

Покажчики можуть бути текстові (слово, група слів) або графічні (зображення). При відображенні на Web-сторінці текстові покажчики звичайно виділяються підкресленням і кольором. Графічні покажчики виділяються рамкою.

Посилання можна розділити на:

- **зовнішні посилання**-які вказують на інші html-документи, що зберігаються на жорсткому диску;
- **внутрішні посилання** - вказують на якісь елементи всередині даного документа;

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

- **посилання на Інтернет-ресурси** - вказують на документи, що знаходяться в мережі Інтернет;
- **посилання на електронну пошту** - дозволяє швидко звернутися до адресата електронної пошти.

Досить часто виникає необхідність включити посилання на різні частини поточного документа. Прикладом може служити зміст з посиланнями на поточні розділи. Для побудови внутрішнього посилання необхідно спочатку створити покажчик, що показує місце призначення за допомогою тегу, наприклад

```
<A Name='якір'> Контрольні питання </A>
```

Після того як покажчик придбав ім'я, можна створити посилання на нього

```
<A Href='#якір'> Контрольні питання </A>
```

Для створення зовнішнього посилання (на інший локальний html-документ) використовують тільки одну частину - покажчик, на той документ, який повинен бути відкритий. Наприклад, якщо необхідно з файлу first.html перейти за посиланням до файлу second.html (і обидва ці файлу знаходяться про одному каталозі), досить включити наступний рядок в html-код:

```
<A Href='second.html'> second.html </A>
```

Для створення посилання на Інтернет-джерело використовується точно такий же механізм, як і при створенні зовнішнього посилання. Відмінність полягає лише в тому, що в адресній частині вказується не шлях до іншого html-файлу, електронна адреса (повна DNS-адреса) Інтернет-джерела.

Наприклад

```
<A Href='http://www.nuczu.edu.ua'> Сайт НУЦЗУ </A>
```

Для створення посилання на електронну пошту використовується зарезервоване слово **mailto**, яке використовується в адресній частини посилання в наступному форматі:

```
<A Href='mailto:ikt@nuczu.edu.ua'> Моя електронна
адреса </A>
```

Самі значення посилань, що міститься у атрибуті **Href** можуть бути **абсолютними** і **відносними**. **Абсолютні** посилання вказують, як правило, на зовнішній ресурс. Для них потрібно вказувати повний шлях:

```
<A Href ="http://www.nuczu.edu.ua/page.html">
Абсолютне посилання </A>

<A Href
="http://www.nuczu.edu.ua/images/figure1.gif"> Посилання
на зображення на сторінці</A>
```

Відносні посилання, навпаки, використовують для переходу на внутрішні сторінки сайту. Для них потрібно вказувати шлях щодо сторінки, з якої робиться послання:

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

```
    <A Href ="/index.html"> Посилання на сторінку в кореневому каталозі </A>

    <A Href ="page.html#seg1"> Посилання на фрагмент сторінки в поточному каталозі </A>

    <A Href =" images /figure1.gif"> Посилання на сторінку в підкаталозі поточного каталогу </A>

    <A Href ="/ docs /manual.html"> Посилання на сторінку в підкаталозі кореневого каталогу </A>

    <A Href ="../ files /index.html"> Посилання на сторінку в вищележачому каталозі </A>
```

6. *Спеціальні символи та крос-браузерність*

Крім тегів, в HTML-документах можуть бути присутніми і спеціальні символи.

Наприклад, © - знак авторського права. Для відображення спеціальних символів використовується мнемонічний або числовий код виду **&ім'я;** або **&#NNNN;** де NNNN - код символу в Юникоде в десяткової системі числення. Наприклад: & (числовий код ©) - амперсанд (&), < - символ «менше» (<) і > - символ «більше» (>), « - ліва друкарська лапки («) тощо.

Гіпертекстові документи обробляються спеціальними застосунками, які читають код розмітки і виводять документ в отформатованому вигляді. Такі застосунки, що мають назву «браузери» (в термінах специфікації HTML - «призначеними для користувача агентами», USER-AGENT), зазвичай надають користувачеві зручний інтерфейс для запиту Web-сторінок, їх перегляду (і виведення на інші зовнішні пристрої) і, при необхідності, відправки введених користувачем даних на сервер. Найбільш популярними на сьогоднішній день браузерами є Internet Explorer, Mozilla Firefox, Apple Safari, Google Chrome і Opera. Поряд з цими існує маса інших браузерів, які використовують їх системні бібліотеки (так званий «движок») або працюють на власному коді.

Різноманітність браузерів і відмінності в їх функціональності, а також початкова орієнтація HTML на підтримку різних пристроїв виводу, призводить розробників Web-сайтів до необхідності вирішення питання про **крос-браузерність**.

Крос-браузерність - властивість сайту відображатися і працювати у всіх популярних браузерах ідентично. Під ідентичністю розуміється відсутність розвалів верстки і здатність відображати матеріал з однаковим ступенем читабельності.

Термін «**крос-браузерність**» почали використовувати під час браузерних війн, розпочатих з середині 90-х років XX ст. У цьому контексті термін стосувався до сайтів, які однаково працюють як в Internet Explorer, так і в Netscape Navigator. У той час виробники стали впроваджувати власні функції для браузерів, що призвело до суттєвих відмінностей відображення Web-вмісту та концептуальним відмінностям в розробці Web-сайтів. В даний час ситуація пом'якшилася (частково через відхід з ринку Netscape), але не настільки, щоб можна було говорити про близьке закінчення браузерних війн.

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Питання для самоконтролю

1. Що таке тег? які бувають теги?
2. Дайте визначення контейнер?
3. Дайте визначення відкриваючого і закриваючого тегу.
4. Для чого призначені атрибути тегу?
5. Який документ називають HTML-документом?
6. Які теги відносяться до заголовних?
7. Як можна створити коментарі до HTML-документу?
8. Назвіть основні теги та їх параметри, які використовуються для оформлення тексту.
9. За допомогою яких тегів створюються нумеровані та марковані списки?
10. Яким є колір і вирівнювання фрагмента тексту за умовчанням?
11. Для чого призначені гіпертекстові посилання? як вони зазвичай виглядають?
12. Який тег дозволяє створити посилання?
13. Що таке URL?
14. Які бувають типи адрес посилання?
15. Чим відрізняється абсолютне посилання від відносного?
16. Що означає поняття якір?
17. Як додати зображення на Web-сторінку?
18. Як вирівняти зображення по центру документа?

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Мова HTML.
Створення таблиць та форм на WEB-сторінках.

Зміст

Мова HTML. Створення таблиць та форм на WEB-сторінках. 128
1. *Засоби опису таблиць в мові HTMLL.* 128
Теги заголовку <TH>… </TH> 129
Тег <CAPTION> 130
Атрибут BORDER 130
Атрибут COLSPAN та ROWSPAN 131
Атрибут WIDTH 131
Атрибут CELLPADDING 132
Атрибути ALIGN та VALIGN 133
Атрибут CELLSPACING 135
Атрибут BGCOLOR та BACKGROUND 136
Використання таблиць в дизайні сторінки 137
2. *Створення HTML-форм на Web-сторінках.* 138
Завдання форми - тег <FORM> 138
Визначення елементів управління форми - тег <INPUT> 138
Створення багаторядкових областей введення тексту - тег <TEXTAREA> 141
Використання списків у формі — тег <SELECT> 141
Питання для самоконтролю 142

1. *Засоби опису таблиць в мові HTMLL.*

З часом розвитку WWW стало ясно, що інструментів, що закладені в мові HTML, недостатньо для якісного відображення різного типу документів. Недоліком HTML була відсутність в його складі засобів відображення таблиць. Для цієї мети зазвичай використовувався предоформатований текст (тег **<PRE>**), В якому таблиця вимальовувалася символами ASCII. Але така форма подання таблиць була недостатньо високої якості і вибивалася із загального стилю документа. після введення таблиць в HTML у розробників Web-сторінок з'явився не просто інструмент для розміщення текстових і числових даних, а потужний засіб дизайну для розміщення в потрібному місці екрану графічних образів і тексту.

Для опису таблиць використовується тег **<TABLE>** . тег **<TABLE>**, являє собою блоковий елемент та автоматично переводить рядок до і після таблиці.

Для створення рядка таблиці використовується тег **<TR>** (**Table Row** – рядок таблиці). Весь текст, інші теги і атрибути, які потрібно помістити в один рядок, повинні розміщуватися між тегами **<TR>**…**</TR>**.

Визначення елементів таблиці відбувається за допомогою тегу **<TD>**. Усередині рядка таблиці зазвичай розміщуються комірки з даними. Кожна клітинка, яка містить текст або зображення, повинна бути оточена тегами **<TD>**…**</TD>**. Число тегів **<TD>**…**</TD>** в рядку визначає число комірок.

```
<HTML>

<BODY>
<H1 ALIGN = center> Таблиця </ H1>
```

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

```
<CENTER>
<table border>
   <tr>
       <td>Текст або дані</td>
       <td>Текст або дані</td>
   </tr>
   <tr>
        <td>Текст або дані</td>
       <td>Текст або дані</td>
    </tr>
</table>
</ CENTER> </ BODY>
</ HTML>
```

Таблиця

Текст або дані	Текст або дані
Текст або дані	Текст або дані

Теги заголовку <TH>… </TH>

Заголовки для стовпців і рядків таблиці задаються за допомогою тега заголовка **<TH>…</TH>** (**Table Header**, заголовок таблиці). Ці теги подібні **<TD>…</TD>**. Відмінність полягає в тому, що текст, укладений між тегами **<TH>…</TH>**, автоматично записується жирним шрифтом і за замовчуванням розташовується посередині комірки. Центрування можна скасувати і вирівняти текст по лівому або правому краю. якщо скористатися тегами **<TD>…</TD>** з тегом **<B>** і атрибутом **align =”center”**, текст теж буде виглядати як заголовок. Однак слід мати на увазі, що не всі браузери підтримують в таблицях жирний шрифт, тому краще ставити заголовки таблиць за допомогою **<TH>**.

```
<HTML>

<BODY>
<TABLE BORDER>
<TR>
<TH> Заголовок знаходиться по центру</TH>
<TH COLSPAN = 2> Заголовок може об'єднувати стовпці </TH>
</ TR>
<TR>
<TH> Заголовок може бути розташований перед стовпцями
</TH>
<TD> Текст або дані </ TD>
<TD> Текст або дані </ TD>
</ TR> <TR>
<TH ROWSPAN = 3> Заголовок може об'єднувати рядки </TH>
<TD> Текст або дані </TD>
<TD> Текст або дані </TD>
</ TR>
<TR>
<TD> Текст або дані </TD>
<TD> Текст або дані </TD>
</ TR>
```

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

```
<TR>
<TD> Текст або дані </TD>
<TD> Текст або дані </TD>
</ TR> </ TABLE>
</ BODY>

</ HTML>
```

Заголовок знаходиться по центру		Заголовок може об'єднувати стовпці
Заголовок може бути розташований перед стовпцями	Текст або дані	Текст або дані
Заголовок може об'єднувати рядки	Текст або дані	Текст або дані
	Текст або дані	Текст або дані
	Текст або дані	Текст або дані

Тег <CAPTION>

Тег **<CAPTION>** дозволяє створювати заголовки таблиці. За замовчуванням заголовки центруються і розміщуються або над (**<CAPTION align = top>**), або під таблицею (**<CAPTION align = bottom>**). Заголовок може складатися з будь-якого тексту та зображень. Текст буде розбитий на рядки, що відповідають ширині таблиці. Іноді тег **<CAPTION>** використовується для підпису під малюнком. Для цього досить описати таблицю без кордонів.

```
<HTML>

<BODY>
<TABLE BORDER>
<CAPTION ALIGN = top> Заголовок над таблицею  </CAPTION>
<TR>
<TD> Текст або дані </ TD> <TD> Текст або дані </ TD>
<TD> Текст або дані </ TD> <TD> Текст або дані </ TD>
 </ TR>
</ TABLE> <BR>
<TABLE BORDER>
<CAPTION  ALIGN  =  bottom>  Заголовок  під  таблицею
</CAPTION>
<TR>
<TD> Текст або дані </ TD> <TD> Текст або дані </ TD>
<TD> Текст або дані <
 </ TR> </ TABLE>
</ BODY>

</ HTML>
```

Заголовок над таблицею

Текст або дані	Текст або дані	Текст або дані	Текст або дані

Текст або дані	Текст або дані	Текст або дані

Заголовок під таблицею

Зазвичай будь-який текст, що не вміщується в один рядок в комірці таблиці, переходить на наступний рядок. Однак при використанні атрибута **Nowrap** з тегами **<TH>** або **<TD>** ширина комірки розширюється настільки, щоб укладений в ній текст помістився в один рядок.

Атрибут BORDER

У тезі **<TABLE>** часто визначають, як будуть виглядати рамки, тобто лінії, що оточують осередки таблиці і саму таблицю. Якщо ви не поставите рамку, то отримаєте

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

таблицю без ліній, але простір під них буде відведено. Того ж результату можна домогтися, задавши <TABLE Border = 0>. Іноді хочеться зробити границю товстішою, щоб вона краще виділялася. Можна для залучення уваги до малюнка або тексту задати виключно жирні кордону. При створенні вкладених таблиць доводиться робити для різних таблиць кордону різної товщини, щоб їх легше було розрізняти.

Атрибут COLSPAN та ROWSPAN

Теги <TD> та <TH> можуть бути модифіковані за допомогою атрибута **Colspan** (**Column Span –**з'єднання стовпців). Якщо ви хочете зробити якусь комірку ширше, ніж верхня або нижня, можна скористатися атрибутом **Colspan**, щоб розтягнути її над будь-якою кількістю звичайних комірок.

```
<HTML>

<BODY>
<CENTER>
<TABLE BORDER = 4>
 <TR>
<TD> Якщо Ви хочете зробити якусь комірку ширше,</ TD>
<TD> ніж верхня </ TD>
<TD> або нижня, </ TD>
<TD> можна скористатися </ TD>
 </ TR>
<TR>
<TD COLSPAN = 4> атрибутом  COLSPAN, щоб
 розтягнути її над будь-якою
 кількістю звичайних Комірок. </ TD>
 </ TR>
</ TABLE>
</ CENTER>
</ BODY></ HTML>
```

Якщо Ви хочете зробити якусь комірку ширше,	ніж верхня	або нижня,	можна скористатися
атрибутом COLSPAN, щоб розтягнути її над будь-якою кількістю звичайних Комірок.			

Атрибут **Rowspan**, який використовується в тегах <TD> і <TH>, подібний до атрибуту **Colspan**, тільки він задає число рядків, на які розтягується комірка. Якщо вказати в атрибуті **Rowspan** число, більше одиниці, то відповідна кількість рядків повинна знаходитися у комірці, що розтягується. Внизу таблиці її помістити не можна.

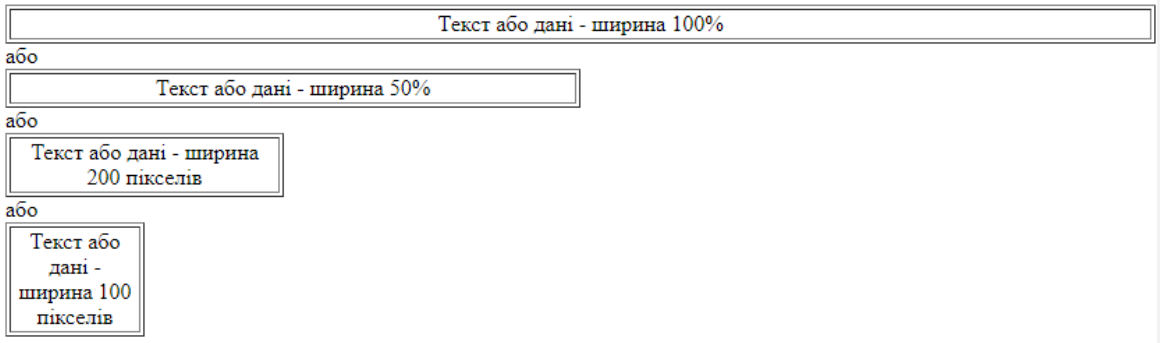
Атрибут WIDTH

Атрибут **Width** застосовується в двох випадках. Можна помістити його в тег <TABLE>, щоб задати ширину всієї таблиці, а можна використовувати в тегах <TD> або <TH>, щоб задати ширину комірки або групи комірок. Ширину можна вказувати в пікселях або у відсотках. Наприклад, якщо задати в тезі <TABLE WIDTH = 250>, ви

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

отримаєте таблицю шириною 250 пікселів незалежно від розміру сторінки на моніторі. При завданні **Width = 50%** в тезі **<TABLE>** таблиця буде займати половину ширини сторінки при будь-якому розмірі зображення на екрані. Так що, вказуючи ширину таблиці у відсотках, треба мати на увазі, що якщо у користувача вузька область перегляду, ваша сторінка може виглядати дещо дивно. Якщо ви користуєтеся пікселями, і таблиця виявляється ширше області перегляду, внизу з'явиться смуга прокрутки для переміщення вправо і вліво по сторінці. Залежно від поставлених завдань і той, і інший спосіб завдання ширини таблиці може виявитися корисним.

```
<HTML>

<BODY>
<TABLE BORDER WIDTH = 100%> <TR>
<TD ALIGN = center> Текст або дані - ширина 100% </ TD>
</TR> </TABLE> або <BR>
<TABLE BORDER WIDTH = 50%>  <TR>
<TD ALIGN = center> Текст або дані - ширина 50% </ TD>
</ TR></TABLE> або <BR>
<TABLE BORDER WIDTH = 200>  <TR>
<TD ALIGN = center> Текст або дані - ширина 200 пікселів
</ TD> </ TR></TABLE>
або <BR>
<TABLE BORDER WIDTH = 100>  <TR>
<TD ALIGN = center> Текст або дані - ширина 100 пікселів
</ TD>
</ TR> </TABLE>
</ BODY>

</ HTML>
```



Якщо комірка не містить даних, вона не матиме границь. Якщо потрібно, щоб у комірці були границі, але не було вмісту, необхідно помістити в неї щось, що не буде видно при перегляді. Можна скористатися тегом переходу на новий рядок **
**. Можна поставити порожні стовпці, визначивши їх ширину в пікселях або відносних одиницях і не ввівши в отримані комірці ніяких даних. Цей засіб може виявитися корисним при розміщенні на сторінці тексту і графіки.

Атрибут CELLPADDING

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Даний атрибут визначає ширину порожнього простору між вмістом комірки і її межами, тобто задає поля всередині комірки.

```
<HTML>

<BODY>
<CENTER>
<TABLE BORDER CELLPADDING = 20>
 <TR>
<TD> Текст або дані </ TD> <TD> Текст або дані </ TD>
<TD> Текст або дані </ TD>
</ TR>
<TR>
<TD> Текст або дані </ TD> <TD> Текст або дані </ TD>
<TD> Текст або дані </ TD>
 </ TR>
</ TABLE>
<BR>
<TABLE BORDER CELLPADDING = 0>
 <TR>
<TD> Текст або дані </ TD> <TD> Текст або дані </ TD>
<TD> Текст або дані </ TD>
</ TR>
<TR>
<TD> Текст або дані </ TD> <TD> Текст або дані </ TD>
<TD> Текст
або дані </ T
 </ TR>
</ TABLE>
</ CENTER>
</ BODY>

</ HTML>
```

Текст або дані	Текст або дані	Текст або дані
Текст або дані	Текст або дані	Текст або дані

Текст або дані	Текст або дані	Текст або дані
Текст або дані	Текст або дані	Текст або дані

Атрибути ALIGN та VALIGN

Теги <TR>, <TD> і <TH> можна модифікувати за допомогою атрибутів **Align** і **Valign**. Атрибут **Align** визначає вирівнювання тексту і графіки по горизонталі, тобто по лівому або правому краю, або по центру. Горизонтальне вирівнювання може бути задано декількома способами:

- **Align = bleedleft** – притискає вміст комірки впритул до лівого краю.
- **Align = left** – вирівнює вміст комірки по лівому краю з урахуванням відступу, заданого атрибутом **Cellpadding**.
- **Align = center** – розташовує вміст комірки по центру.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

- **Align = right** – вирівнює вміст комірки по правому краю з урахуванням відступу, заданого атрибутом **Cellpadding**.

```
<HTML>

<BODY>
<TABLE BORDER WIDTH = 100%>
 <TR>
<TD ALIGN = left> Текст або дані </ TD>
<TD ALIGN = center> Текст або дані </ TD>
<TD ALIGN = right> Текст або дані </ TD>
</ TR>
<TR>
<TD ALIGN = right> Текст або дані </ TD>
<TD ALIGN = center> Текст або дані </ TD>
<TD ALIGN = left> Текст або дані </ TD>
</ TR>
<TR>
<TD ALIGN = right> Текст або дані </ TD>
<TD ALIGN = right> Текст або дані </ TD>
<TD ALIGN = right> Текст або дані </ TD>
</ TR>
<TR>
<TD ALIGN = center> Текст або дані </ TD>
<TD ALIGN = center> Текст або дані </ TD>
<TD ALIGN = center> Текст або дані </ TD>
</ TR>
<TR>
<TD ALIGN = left> Текст або дані </ TD>
<TD ALIGN = left> Текст або дані </ TD>
<TD ALIGN = left> Текст або дані </ TD>
</ TR>
</ TABLE>
</ BODY>

</ HTML>
```

Текст або дані	Текст або дані	Текст або дані
Текст або дані	Текст або дані	Текст або дані
Текст або дані	Текст або дані	Текст або дані
Текст або дані	Текст або дані	Текст або дані
Текст або дані	Текст або дані	Текст або дані

Атрибут VALIGN здійснює вирівнювання тексту і графіки всередині комірки по вертикалі. Вертикальне вирівнювання може бути задано декількома способами:

- **Valign = top** – вирівнює вміст комірки по її верхній межі.
- **Valign = middle** – центрує вміст комірки по вертикалі.
- **Valign = bottom** – вирівнює вміст комірки по її нижній межі.

```
<HTML>
```

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

```
<BODY>
<CENTER>
<TABLE BORDER WIDTH = 90%>
<TR>
<TD WIDTH = 100> Атрибут VALIGN здійснює вирівнювання
тексту і графіки всередині комірки по вертикалі. </ TD>
<TD VALIGN = top> верх, </ TD>
<TD VALIGN = middle> середина, </ TD>
<TD VALIGN = bottom> низ. </ TD>
</ TR>
</ TABLE>
 </ CENTER>
</ BODY>

</ HTML>
```

Атрибут VALIGN здійснює вирівнювання тексту і графіки всередині осередки по вертикалі.	верх,	середина,	низ.

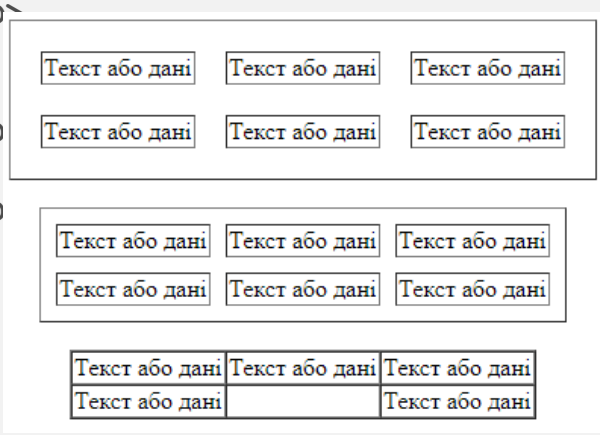
Атрибут CELLSPACING

Атрибут **Cellspacing** визначає ширину проміжків між комірками в пікселах. Якщо цей атрибут не вказаний, за замовчуванням задається величина, рівна двом пікселям. За допомогою атрибута **Cellspacing** можна керувати розміщенням тексту і графіки там, де потрібно. Якщо, потрібно залишити порожнє місце, можна вписати в комірку пробіл.

```
<HTML>

<BODY>
<CENTER>
<TABLE BORDER CELLSPACING = 20>
 <TR>
<TD> Текст або дані </ TD>
<TD> Текст або дані </ TD>
<TD> Текст або дані </ TD>
</ TR>
<TR>
<TD> Текст або дані </ TD>
<TD> Текст або дані </ TD>
<TD> Текст або дані </ TD>
 </ TR>
</ TABLE>
<TABLE BORDER CELLSPACING = 10>
 <TR>
```


©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

```
<TD> Текст або дані </ TD>
<TD> Текст або дані </ TD>
<TD> Текст або дані </ TD>
</ TR>
<TR>
<TD> Текст або дані </ TD>
<TD> Текст або дані </ TD>
<TD> Текст або дані </ TD>
 </ TR>
</ TABLE>
<TABLE BORDER CELLSPACING = 0>
 <TR>
<TD> Текст або дані </ TD>
<TD> Текст або дані </ TD>
<TD> Текст або дані </ TD>
</ TR>
<TR>
<TD> Текст або дані </ TD
<TD> </ TD>
<TD> Текст або дані </ TD
 </ TR>
</ TABLE>
</ CENTER>
</ BODY>

</ HTML>
```



Атрибут BGCOLOR та BACKGROUND

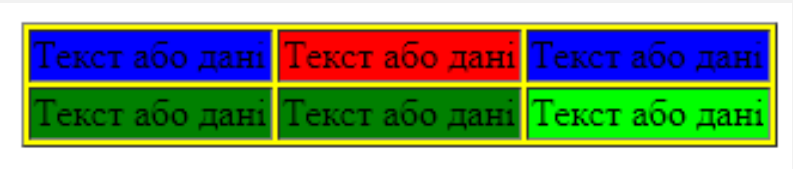
Атрибут **Bgcolor** дозволяє встановити колір фону. Залежно від того, з яким тегом (<TABLE>, <TR>, <TD>) він застосовується, колір фону може бути встановлений для всієї таблиці, для рядка чи для окремої комірки. Значенням даного атрибута є RGB-код або стандартна назву кольору.

```
<HTML>

<BODY>
<CENTER>
<TABLE BORDER BGCOLOR = yellow>
 <TR BGCOLOR = blue>
<TD> Текст або дані </ TD>
<TD BGCOLOR = red> Текст або дані
</ TD>
<TD> Текст або дані </ TD>
</ TR>
<TR BGCOLOR = green>
<TD> Текст або дані </ TD>
<TD> Текст або дані </ TD>
<TD BGCOLOR = lime> Текст або дані
 </ TD>
```

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

```
</ TR>
</ TABLE>
</ CENTER>
</ BODY>

</ HTML>
```

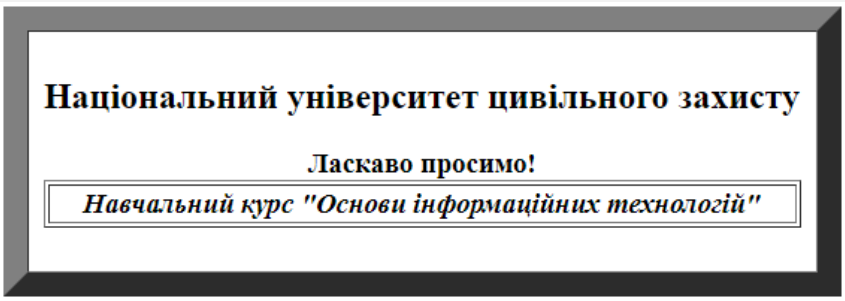


Атрибут **Background** задає фонове зображення для таблиць. Застосовується до тегів <TABLE> та <TD>. Його значенням є URL файлу з фоновим зображенням.

Використання таблиць в дизайні сторінки

Таблиці гарні тим, що при бажанні можна зробити їх межі невидимими. Це дозволяє за допомогою тега <TABLE> красиво розміщувати на сторінці текст і графіку. Тег <TABLE> навіть дотепер залишається потужним засобом форматування в HTML. Якщо браузер підтримує таблиці, він зазвичай правильно відображає найбільш цікаві ефекти, отримані з їх допомогою

```
<HTML>

<BODY>
<CENTER>
<TABLE CELLPADDING = "10" CELLSPACING = "0"
BORDER = "16">
<TR>
<TD ALIGN = "center">
 <H2> Національний університет цивільного захисту </ H2>
 <H3> Ласкаво просимо! </ H3>
<TABLE BORDER WIDTH = "100%">
<TR>
<TD ALIGN = "center"> <I> Навчальний курс
 "Основи інформаційних технологій" </ I> </ TD>
</ TR>
</ TABLE>
</ TD>
</ TR>
</ TABLE>
</ CENTER>
</ BODY>

<HTML>
```



Є кілька способів розфарбувати таблицю, в основному вони залежать від використовуваного браузера.

Ви не тільки можете оточити таблицю красивою рамкою, але ще і задати для неї колір, відмінний від кольорів тексту та фону. Створіть простий сірий GIF (або будь-який GIF, який ви хотіли б мати в якості фону) і визначте його в тезі <BODY> як фон сторінки. Потім задайте колір фону сторінки. В результаті ваш тег <BODY> буде виглядати приблизно так:

```
<BODY Background = "image.gif" Bgcolor = "#FF0000">
```

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Ви створили подвійний фон - GIF і заданий колір. В результаті фоновий колір буде видно на всіх кордонах таблиць і горизонтальних лініях (**<HR>**). Незалежно від того, є ваш фоновий GIF сірим чи ні, кольорові лінії і кордони таблиць будуть помітно виділятися. Якщо фоновий GIF влаштований не дуже складно, час завантаження сторінки зросте лише трохи.

2. *Створення HTML-форм на Web-сторінках.*

Форми були створені і використовуються в WWW для отримання відгуку користувача на надану інформацію та збору даних про користувача. Після заповнення користувачем форми і запуску процесу її обробки, інформація з неї потрапляє до програми, яка працює на сервері. Простота використання тегу **<MAILTO:>** в формах дозволяє навіть власникам невеликих сторінок отримувати відгуки від своїх читачів. Для обробки великої кількості відгуків використовуються програми, що підтримують **Common Gateway Interface** (CGI) і розташовані на сервері, в адреса якого надходять відгуки. Таким чином користувач може інтерактивно взаємодіяти з Web-сервер через *Internet*.

Завдання форми - тег <FORM>

Завдання форми відбувається через тег **<FORM>**, він позначає документ як форму і визначає межі використання інших тегів, що розміщуються в формі. Тег **<FORM>** визначається послідовністю тегів **<INPUT>**, розміщених всередині контейнеру **<FORM>...</FORM>**. Для опису обробки даних, що вводяться користувачем в форму виступають два атрибути: метод (**Method**) та дія (**Action**). Метод (приймає значення **GET** або **POST**) визначає, як повинні оброблятися вхідні дані з форми, а дія вказує на URI (**Uniform Resource Identifier**) програми, відповідальної за обробку цих даних.

```
<FORM METHOD = post
 ACTION = mailto: yourname@your.email.address>
```

Визначення елементів управління форми - тег <INPUT>

Для визначення елементів управління всередині форми, куди вводяться дані використовується тег **<INPUT>**. Він формує поле для введення інформації користувачем. Це може бути **текстове поле**, **опція**, **зображення** або **кнопка**. Вид поля введення визначається значенням атрибута **Type**.

Атрибут Type = TEXT

Коли користувачеві необхідно ввести невелику кількість тексту (одну або кілька рядків), використовується тег **<INPUT>** та атрибут **Type** встановлюється в значення **text**. Це значення прийнято за замовчуванням і вказувати його необов'язково. Крім того, задається атрибут **Name** для визначення найменування змінної поля.

```
Ваше ім'я <BR>
<INPUT NAME = Name SIZE =
```

Ваше ім'я

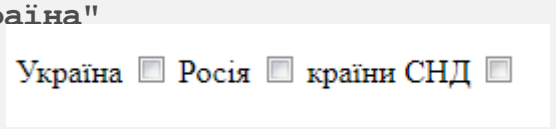
Перелік лекцій

Є ще три додаткових атрибута, які можна використовувати. перший називається **Maxlength**, він обмежує число символів, що вводяться користувачем в даний поле. За замовчуванням дане число не обмежена. Другим атрибутом є **Size**, що визначає розмір видимої на екрані області, займаної поточним полем. Значення за замовчуванням визначається типом браузера. якщо значення **Maxlength** більше ніж **Size**, браузер буде прокручувати дані у вікні. Останнім з додаткових атрибутів є атрибут **Value**, що забезпечує початкове значення поля введення.

Атрибут Type = CHECKBOX

Для створення незалежних прапорців в формах HTML використовується тег **<INPUT>** зі значенням атрибута **Type = Checkbox**. Залежно від змісту форми користувач може відзначити кілька прапорців. Коли форма використовує тег **<INPUT>** зі значенням атрибута **Checkbox**, в ньому мають бути присутні і атрибути **Name**, і **Value**. Атрибут **Name** вказує на найменування даного поля-прапорця (для ідентифікації та розрізнення з іншими полями введення). В атрибуті **Value** міститиметься значення поля.

```
<BR> Україна <INPUT NAME = "Країна" TYPE = checkbox
VALUE = "Ukraine">
Росія <INPUT NAME = "Країна" TYPE = checkbox
VALUE = "Russia">
країни СНД <INPUT NAME = "Країна"
TYPE = checkbox
VALUE = "SND">
```

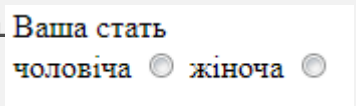


У деяких випадках необхідно ініціалізувати якись прапор, як уже відзначений. У таких випадках тег **<INPUT>** повинен містити атрибут **Checked**.

Атрибут Type = RADIO

У деяких випадках потрібно організувати вибір одного з декількох можливих значень. Для створення форми введення при виборі користувачем одного значення з декількох можливих необхідно використовувати тег **<INPUT>** з атрибутом **Type = Radio**. Коли в формі застосовується даний атрибут, в тезі **<INPUT>** повинні бути вказані атрибути **Name** і **Value**. Атрибут **Name** вказує найменування відповідного поля (кнопки). Атрибут **Value** містить значення поля.

```
Ваша стать<BR>
чоловіча
<INPUT NAME = "Пол" TYPE = radio VALUE = "Чоловічий">
жіноча <INPUT NAME = "Пол" TYPE = radi
VALUE = "Жіночий">
```



Атрибут Type = IMAGE

Залежно від вмісту форми може трапитися так, що користувачеві буде потрібно клацнути мишею на зображенні, щоб завершити роботу з формою. Для цього програмісти використовують тег **<INPUT>** з атрибутом **Type = Image**. Коли користувач клацає мишею по зображенню, браузер зберігає відповідне місце на екрану. Далі він "обробляє" введену в форму інформацію. Коли форма використовує атрибут **Image**,

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

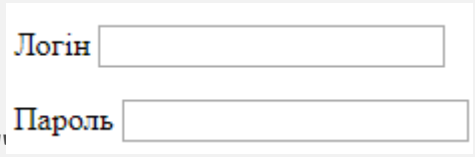
тег **<INPUT>** повинен містити також атрибути **Name** і **Src**. **Name** вказує найменування поля введення форми. Атрибут **Src** містить URI файлу - джерела зображення. Атрибут **Align** є додатковим і використовується аналогічно тому ж атрибуту тегу **<IMG>**.

```
<BR> Виберіть точку <INPUT TYPE = image
NAME = point SRC = image.gif>
```

Атрибут Type = PASSWORD

Якщо в формі потрібно організувати введення пароля, то атрибут **Type** можна встановити в значення **Password**. Використовуючи даний тип, можна організувати введення пароля без виведення на екран складових його символів. При цьому слід пам'ятати, що введені дані передаються по незахищених каналах зв'язку і можуть бути перехоплені.

```
<p>Логін
<INPUT NAME = login>
<p>Пароль
<INPUT TYPE=password NAME="pass"
```



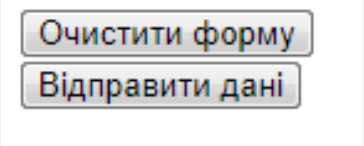
Атрибут Type= RESET

Коли користувач заповнює форму, йому може знадобитися почати все спочатку. На такий випадок існує кнопка **Reset**, по якій користувач може клацнути мишею, щоб повернутися до первинних значень полів. Коли користувач вибирає дану кнопку, форма відновлює початкові значення всіх елементів. Для створення кнопки **Reset** використовується тег **<INPUT>** з атрибутом **Type = Reset**. Браузер в свою чергу буде виводити зображення цієї кнопки. Якщо в формі використовується атрибут **Reset**, тег **<INPUT>** може додатково містити атрибут **Value**. Даний атрибут визначає напис на зображенні кнопки.

Атрибут Type= SUBMIT

Використовуючи форму HTML для введення інформації від користувача, необхідно забезпечити користувачеві можливість завершити введення даних. Для цього використовується тег **<INPUT>** з атрибутом **Type = Submit**. Браузер, в свою чергу, виводить даний елемент, як кнопку, по якій користувач може клацнути, щоб завершити процес редагування. Коли у формі використовується даний елемент, користувач може додати ще два додаткових атрибута: **Name** и **Value**. Атрибут **Name** зберігає назву змінної поля у вашій формі. Атрибут **Value** - Даний атрибут визначає напис на зображенні кнопки.

```
<BR>
<INPUT TYPE = reset
VALUE = "Очистити форму">
<BR><INPUT TYPE= submit
VALUE="Відправити дані">
```



Атрибут Type= HIDDEN

Приховані поля. Додавання в тег **<INPUT>** атрибута **Type = Hidden** дозволить включити в форму, яка відправляється значення атрибутів **Name** і **Value**, які

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

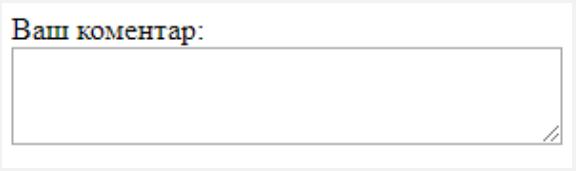
користувач змінити не може. Такі мітки корисні при наявності декількох форм для подальшої обробки даних.

Створення багаторядкових областей введення тексту - тег <TEXTAREA>

Залежно від типу форми може знадобитися організувати введення великої кількості тексту. У таких випадках використовується тег **<TEXTAREA>** для створення текстового поля з кількох рядків. Даний тег використовує три атрибути: **Cols**, **Name** и **Rows**.

- Атрибут **Cols** – вказує число колонок (число символів), що містяться в текстовій області.
- Атрибут **Name** – визначає найменування поля.
- Атрибут **Rows** – задає кількість видимих рядків текстової області.

```
Ваш коментар: <BR>
<TEXTAREA NAME = "тема"
COLS = "38" ROWS = "3">
</TEXTAREA>
```



Використання списків у формі — тег <SELECT>

Коли форми HTML стають більш складними, в них часто включають списки з прокруткою і меню, що випадають. Для цього використовують тег **<SELECT>**, для визначення кожного з списку пунктів використовують тег **<OPTION>**. Тег **<SELECT>** підтримує три необов'язкових атрибути: **Multiple**, **Name** и **Size**.

- Атрибут **Multiple** – дозволяє вибрати більш ніж одне найменування.
- Атрибут **Name** – визначає найменування об'єкта.
- Атрибут **Size** – визначає число видимих користувачеві пунктів списку. Якщо в форме встановлено значення атрибута **Size = 1**, то браузер виводить на екран список у вигляді меню, що випадає. У разі **Size > 1** браузер представляє на екрані звичайний список.

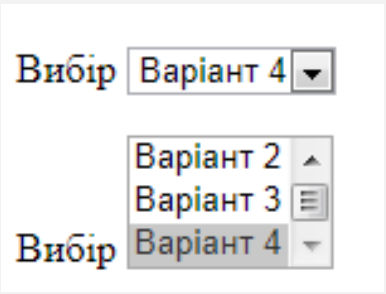
Тег **<OPTION>** у формі може використовуватися тільки всередині тега **<SELECT>**. Ці теги підтримують два додаткових атрибута: **Selected** и **Value**.

- Атрибут **Selected** – використовується для початкового вибору значення елемента за замовчуванням.
- Атрибут **Value** – Вказує на значення, яке буде оброблятися формою після вибору користувачем даного пункту. За замовчуванню значення поля дорівнює значенню тегу **<OPTION>**.

```
<BR> Вибір
<SELECT NAME = "Вибір">
<OPTION> Варіант 1
```

Перелік лекцій

```
<OPTION> Варіант 2
<OPTION VALUE = "Варіант 3"> Варіант 3
<OPTION SELECTED> Варіант 4
</SELECT>
<P> Вибір
<SELECT NAME = "Вибір" SIZE=3>
<OPTION> Варіант 1
<OPTION> Варіант 2
<OPTION VALUE = "Варіант 3"> Варіант
<OPTION SELECTED> Варіант 4
</SELECT>
```



Питання для самоконтролю

1. Якими основними тегами здійснюється формування таблиці?
2. Які теги можуть бути розміщені усередині контейнерного тегу <**TABLE**> </**TABLE**>?
3. За допомогою яких параметрів можна задати мінімальну ширину і висоту таблиці? У яких одиницях можна указувати розміри таблиці?
4. Який тег дозволяє створити заголовок таблиці і описати його положення?
5. Який параметр задає горизонтальне вирівнювання заголовка?
6. За допомогою якого тегу можна задати заголовні комірки?
7. У чому полягає відмінність тегів <**TD**> і <**TH**>?
8. Що задає конструкція <**Caption Align=Top**> Текст </**Caption**>, <**Table Align=Center**>?
9. Який параметр визначає відстань між межею комірки і її вмістом та відстань між зовнішніми межами комірок?
10. Що задає параметр **Border** та **Bordercolor** тегу <**TABLE**>?
11. Чи можна задати колір фону і фоновий рисунок для таблиці, рядка або комірки?
12. Які параметри використовуються для об'єднання комірок?
13. Для чого призначена Форма на веб-сторінках?
14. Яке призначення програм, підтримуючих **CGI**?
15. Який тег визначає документ як Форму?
16. Перерахуєте елементи управління Форми.
17. Що дозволяє створити у Формі список?
18. Як зробити список що випадає?
19. Як створити область для багаторядкового введення тексту?

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Основи CSS.
Загальні принципи верстки та оформлення Web-сторінки

Зміст

Основи CSS. Загальні принципи верстки та оформлення Web-сторінки 143
1. *Загальні відомості про Cascading Style Sheets.* 143
2. *Підключення таблиці стилів* 144
Підключення зовнішньої таблиці стилів 144
Підключення внутрішньої таблиці стилів 145
Вбудовані стилі 145
Пріоритет використання стилів 145
3. *Синтаксис CSS та селектори* 146
Селектор класу (class) 146
Селектор ідентифікатора (id) 147
Групової селектор 147
Селектор нащадків 148
Псевдокласи селекторів 149
Коментарі CSS та позначення опису синтаксису 151
4. *Параметри тексту в CSS* 152
5. *Блокова модель* 155
Характеристика фону та кольору тексту 155
Блокова модель 156
Позиціонування блоків 158
Питання для самоконтролю 160

1. *Загальні відомості про Cascading Style Sheets.*

Назва "**Каскадні таблиці стилів**" походить від англійського *Cascading Style Sheets*, абревіатурою якого є **CSS**. Вперше стилі з'являються в HTML 4.0 для визначення представлення елементів HTML і рішення проблем відображення документів. Стилі зазвичай зберігаються в таблицях стилів, які можуть бути визначені як всередині HTML-документа, так і в спеціальному файлі з розширенням **.css**.

Використовуючи окремі файли для зберігання таблиць стилів, можна істотно скоротити обсяг роботи по оформлення Web-сторінки. Також можна визначити кілька стилів, які, підкоряючись існуючим правилам, будуть каскадно задавати один певний стиль.

Як відомо, **мова HTML** була створена для опису вмісту документа. Її теги були призначені для визначення "заголовка", "параграфа", "таблиці" (**<H1>**, **<P>**, **<TABLE>** тощо). Спочатку для представлення документів не було передбачено жодних тегів форматування, тобто передбачалося, що про подання документа подбає браузер. Але це породило процес створення браузерами власних тегів і атрибутів до вихідної специфікації HTML-браузерами Netscape і Internet Explorer (такі, наприклад, як тег **<FONT>** і атрибут **color**), що ускладнило створення Web-сайтів, на яких вміст документів HTML було чітко відокремлено від рівня представлення документа.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

У цій ситуації **консорціум W3C** (*World Wide Web Consortium*) - некомерційна організація, відповідальна за стандартизацію HTML - створила при розробці стандарту HTML 4.0 каскадні таблиці стилів. Таким чином, необхідність розділу вмісту документа і його відображення привела до створення технології CSS.

Таблиці стилів визначають **правило відображення** елементів HTML (як, наприклад, тег **<FONT>** та атрибут **color** в HTML 3.2). Як вже відмічалося, каскадні таблиці стилів зазвичай зберігаються в зовнішніх файлах з розширенням **.css**. Цей спосіб зручний тим, що дозволяє змінити зовнішній вигляд і компонування всіх необхідних Web-сторінок, редагуючи тільки один, єдиний документ **CSS**.

Ще раз загостримо: **CSS** стали проривом в **Web-дизайні**. Основою для нього послужило те, що технологія дозволяє розробникам одночасно керувати стилем і компонуванням безлічі Web-сторінок, визначати стиль для кожного елемента HTML і застосовувати його до будь-якої кількості Web-сторінок. Також **CSS** дає можливість з одного документа посилатися на кілька зовнішніх таблиць стилів. Таким чином, щоб зробити глобальну зміну, треба просто змінити стиль, і всі елементи в Web-сторінки будуть автоматично змінені.

2. Підключення Таблиці стилів

Коли браузер прочитує таблицю стилів, він форматує документ згідно з цією таблицею. Існує три способи підключення таблиці стилів.

Підключення зовнішньої таблиці стилів

Цей спосіб зазвичай застосовується в ситуаціях, коли один стиль визначається для безлічі сторінок. Якщо необхідно швидко поміняти дизайн Web-сайту цей спосіб ідеально підходить - редагується тільки один файл з таблицею стилів. Для підключення зовнішньої таблиці стилів призначений тег **<link>**, який розміщується в розділі заголовка:

```
    <HEAD>
    <LINK Rel = "stylesheet" Type = "text / css" Href =
"first.css" />
    </HEAD>
```

Даний код вказує браузеру на те, що визначення стилю зберігається в файлі **first.css**. Останній, в свою чергу, виконає форматування документа згідно з цим визначенням стилю.

> Деякі зауваження: зовнішню таблицю стилів можна створити в будь-якому текстовому редакторі; файл із зовнішньою таблицею стилів не повинен містити ніяких тегів **html**; файл із зовнішньою таблицею стилів необхідно зберегти з розширенням **.css**.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Підключення внутрішньої таблиці стилів

Даний спосіб застосовується тоді, коли один документ має унікальний стиль. Для визначення внутрішніх стилів використовують тег **<STYLE>**, який розміщується в розділі заголовка:

```
<HEAD>
    <STYLE Type = "text / css">
        Li {Font-family: arial}
        P {Font-size: 25%}
        H1 {Margin-right: 10pt}
    </ STYLE>
</ HEAD>
```

Відповідно до визначень стилю браузер форматує документ. Однак якщо він зустріне незнайомі теги, то проігнорує їх. Наприклад, старий браузер, який не підтримує стилі, проігнорує тег **<STYLE>**, але вміст останнього буде виведено на сторінці. Щоб цього не сталося, необхідно приховати опис стилю в елементі коментаря HTML.

Вбудовані стилі

Даний спосіб використовується в тому випадку, якщо необхідно застосувати стиль лише до одного елемента. Метод має істотний недолік, тому що він змішує вміст документа з його поданням і втрачає, відповідно, багато переваг таблиць стилів.

Для визначення вбудованих стилів у відповідному тезі застосовують атрибут **style**, який може містити будь-яку властивість CSS. У прикладі показано, як задати колір фону документа (**blue**) і колір заголовка **<H1>** (**yellow**).

```
<BODY Style = "Background-color: blue">
    <H1 Style = "Color: yellow"> Тема H1 </ H1>
</BODY>
```

Пріоритет використання стилів

У разі об'єднання або використання декількох таблиць стилів можуть виникнути ситуації, коли деякі властивості задані для одного і того ж елементу HTML в різних таблицях стилів. Тому для каскадних таблиць стилів визначено пріоритет використання. Якщо для елемента HTML визначено більше одного стилю, то всі стилі будуть послідовно "каскадуватися" в нову "віртуальну" таблицю стилів, згідно з такими правилами:

- стилі, використовувані за замовчуванням браузером;
- стилі, що зберігаються в **зовнішній** таблиці;
- стилі, що зберігаються у **внутрішній** таблиці стилів (всередині тега **<head>**);

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

- вбудований стиль (всередині елемента HTML).

Ці правила визначають порядок збільшення пріоритету стилів.

Таким чином, У разі об'єднання або використання декількох таблиць стилів коли деякі властивості задані для одного і того ж елементу HTML в різних таблицях стилів, вбудований стиль (всередині елемента HTML) має найвищий пріоритет, тобто буде перевизначати стиль, який оголошений в тезі **<HEAD>**, А стиль заданий у тегу **<HEAD>** буде перевизначати стиль, заданий у зовнішній таблиці стилів або в браузері (значення за замовчуванням).

3. Синтаксис CSS та селектори

Синтаксис CSS складається з трьох частин: **селектора**, **властивості** і **значення**:

```
Селектор {Властивість: значення}
```

Селектор - **тег** (елемент мови HTML), який необхідно визначити. **Властивість** - атрибут, який бажано змінити. Кожна властивість може приймати значення. Існує ряд правил і рекомендацій, які необхідно запам'ятати.

- Властивість і значення розділяються двокрапкою і поміщаються всередині фігурних дужок:

```
P {font-size: 75%}
```

- Якщо значення складається з декількох слів, то необхідно помістити значення в лапки:

```
H1 {font-family: "lucida calligraphy"}
```

- Якщо потрібно визначити більш одного властивості, то необхідно розділити властивості крапкою з комою:

```
TABLE {font-family: arial, "sans serif"; border-style: dotted}
```

- Щоб визначення стилів було зручно читати, можна кожну властивість писати на окремому рядку:

```
H2
{
  Font-family: arial;
  Margin-right: 20pt;
  Color: #ffffff
}
```

Селектор класу (class)

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

При Web-розробці часто виникає необхідність в завданні різних стилів для одного і того ж типу елемента HTML. Для вирішення цього завдання використовують селектор класу.

```
.left {Margin-left: 40pt}
.right {Margin-right: 80pt}
```

У наступному прикладі всі елементи HTML, мають **class = "left"**, будуть мати зовнішній відступ зліва, рівний 40 пунктам.

```
<H1 Class = "left"> Заголовок з відступом 40 пунктів зліва. </H1>
```

У наступному коді елементи **<TABLE>** і **<P>** мають **Class = "right"**. Це означає, що обидва елементи будуть слідувати правилам в селекторі **.right** (мати відступ 80 пунктів праворуч):

```
<TABLE Class = " right "> Ця таблиця буде мати зовнішній відступ праворуч, рівний 80 пунктам. </ TABLE>
<P Class = " right "> Цей параграф також матиме зовнішній відступ праворуч, рівний 80 пунктам. </ P>
```

Для коректності обробки коду браузерами не рекомендується починати ім'я класу з числа, тому що це не буде працювати в Mozilla / Firefox.

Селектор ідентифікатора (id)

Стилі елементів HTML можна визначити також за допомогою **селектора ідентифікатора**, який визначається символом #.

Наступне правило стилю буде застосовано до елементу, який має атрибут id зі значенням **"fontsz":**

```
#fontsz {font-size: 50%}
```

Наступне правило стилю буде застосовано до того елементу **<ul>**, який має атрибут **Id** зі значенням **"first"**:

```
Ul #first
{List-style: disc;
Color: #ffffff }
```

Групової селектор

Для скорочення кількості коду в CSS файлі використовується груповий селектор. Сенс групового селектора в тому, що коли ми хочемо задати якесь загальне властивість для деяких елементів, ми прописуємо селектори елементів через кому. У наступному прикладі в групу були об'єднані всі елементи абзаців, таблиць і списків. Всі ці елементи будуть виведені шрифтом sans serif:

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

```
P, TABLE, LI
{
font-family: "sans serif";
}
```

Селектор нащадків

Вся HTML-сторінка схожа на дерево. Головний елемент сторінки це тег **<HTML>**, а всі інші теги це його нащадки. Теги **<HEAD>** і **<BODY>** є нащадками тега **<HTML>**. Тобто тег **<HTML>** для них є для них батьком. Наочно можна побачити на рис. 8.1.

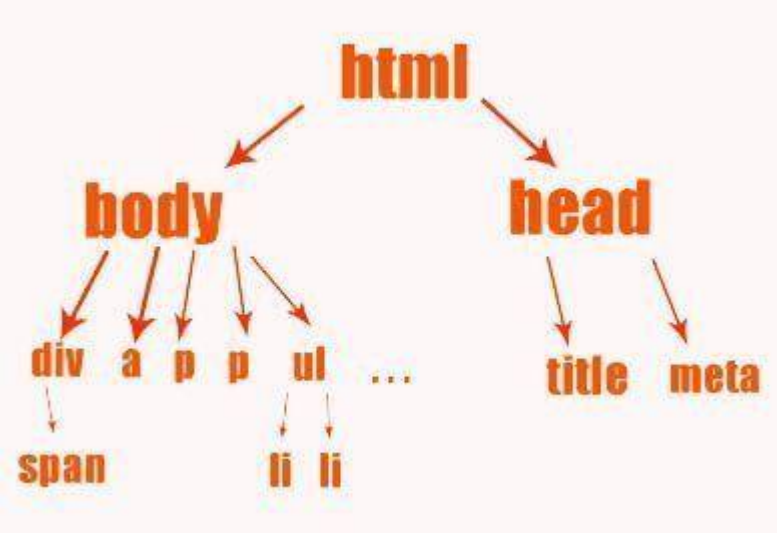


Рис. 8.1 Деревоподібна структура HTML-сторінки

А так як HTML-сторінка це ієрархія елементів, значить ми можемо звертатися до будь-якого елементу сторінки через **спадкування** властивостей

Спадкування - це перенесення правил форматування для елементів, що знаходяться всередині інших. Наприклад, якщо для тегу **<P>** заданий червоний колір тексту, а для курсиву **<I>**, який знаходиться всередині абзацу, ні, то в цьому випадку вкладений елемент успадковує властивості свого батька і курсивний текст також буде червоного кольору.

Спадкування корисно для завдання властивостей, застосовуваних до елементу за замовчуванням. Так, досить задати параметри форматування тегу **<TABLE>** і до елементів таблиці **<TD>** вони будуть застосовані автоматично. Точно так само можна визначити властивості тегу **<BODY>**, який породжує стиль всіх інших елементів Web-сторінки.

Наприклад, необхідно розфарбувати всі посилання на сторінці, які знаходяться всередині списку. Нижче представлена HTML код необхідної нам сторінки.

```
<html>
<head>
 <title>Головна</title>
 <meta http-equiv="Content-Type" content="text/html;
charset = utf-8">
 <link rel="stylesheet" type="text/css" href=" style.
css">
```

Перелік лекцій

```
    </head>
    <body>
        <p>Вивчення CSS</p>
        <p><a href="#">Перейти на головну</a></p>
        <p>Vivamus nec nulla a ipsum vulputate interdum. Nunc fermentum lacus velit, id dapibus sapien porttitor sit amet.</p>
        <ul>
         <li>Чудовий сайт <a href="#">SiteHere.RU</a></li>
         <li>Nunc fermentum lacus velit, id dapibus sapien porttitor sit amet. Перейти за посиланням <a href="#"> Перейти </a> </li>
        </ul>
    </body>

    </html>
```

З коду видно, що є посилання, які знаходиться в тезі **<HTML>**, а також посилання які знаходяться в списку в тегах **<LI>**. Для того щоб вирішити нашу задачу існує селектор нащадків. Рішення в CSS буде виглядати наступним чином:

Для того щоб вибрати тільки ті посилання, які знаходяться всередині тегів <li> необхідно в CSS написати наступне:

```
    Body{

    Background-color:#C1BAFF;
    }
    P{

Color:#4600BF;
    }
    UL LI A{

Color:#ffffff;

    }
```

Рис. 8.2 Селектори нащадків

Як бачите в прикладі нижче у нас вибралися тільки посилання які знаходяться всередині тега **<LI>**, а посилання в тезі **<P>** залишилася нефарбованими. Тобто ми можемо отримати доступ до будь-якого елементу через ієрархію HTML–елементів (рис. 8.2).

Псевдокласи селекторів

Псевдокласи селекторів визначають динамічний стан елементів, яке змінюється за допомогою дій користувача, а також положення в структурі документу. Прикладом такого стану служить текстове посилання, яка змінює свій колір при наведенні на неї курсора миші. При використанні псевдокласів браузер не

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

перевантажує поточний документ, тому за їх допомогою можна отримати різні динамічні ефекти на сторінці.

Синтаксис застосування псевдокласу селектору наступний:

```
Селектор: Псевдоклас { властивість: значення}
```

Спочатку вказується **селектор**, до якого додається псевдоклас, потім слідує двокрапка, після якого йде ім'я псевдокласу. Допускається застосовувати псевдокласи до імен ідентифікаторів або класів (`A.menu:hover {color: green}`), а також до контекстних селекторів (`.menu A: hover {background: # fc0}`). Якщо псевдоклас вказується без селектора попереду (`: hover`), то він буде застосовуватися до всіх елементів документа.

Умовно всі псевдокласи поділяються на три групи:

- Ті, що визначають стан елементів;
- Ті, що мають відношення до дерева елементів;
- Такі, що вказують мову тексту.

До групи **псевдокласів, що визначають стан елементів** належать псевдокласи, які розпізнають поточний стан елемента і застосовують стиль тільки для цього стану.

`:active` – Відбувається при активації користувачем елементу. Наприклад, посилання стає активною, якщо навести на неї курсор і клацнути мишкою. Незважаючи на те, що активним може стати практично будь-який елемент веб-сторінки, псевдоклас `:active` використовується переважно для посилань.

`:link` – Застосовується до посиланнями, що невідвідувалися, тобто таких, на які користувач ще не натискав. Браузер деякий час зберігає історію відвідувань, тому посилання може бути позначене як відвідане хоча б тому, що по ній був зафіксований перехід раніше. Запис `A {...}` і `A: link {...}` за своїм результатом рівноцінна, оскільки в браузері дає один і той же ефект, тому псевдоклас `:link` можна не вказувати. Винятком є якоря, на них дія `:link` не поширюється.

`:focus` – Застосовується до елемента при отриманні ним фокусу. Наприклад, для текстового поля форми отримання фокусу означає, що курсор встановлений в поле, і за допомогою клавіатури можна вводити в нього текст.

`:hover` – Псевдоклас активізується, коли курсор миші знаходиться в межах елемента, але клацання по ньому не відбувається. Псевдоклас `:hover` не обов'язково повинен застосовуватися до посилань, його можна додавати і до інших елементів документа.

`:visited` – Даний псевдоклас застосовується до відвіданих посилань. Зазвичай таке посилання змінює свій колір за замовчуванням на фіолетовий, але за допомогою стилів колір і інші параметри можна задати самостійно.

> Потрібно звернути увагу на те, що порядок проходження псевдокласів має значення. Спочатку вказується **:visited**, а потім йде **:hover**, в іншому випадку відвідані посилання не будуть змінювати свій колір при наведенні на них курсору.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Група **псевдокласів, що мають відношення до дерева документа**, визначає положення елемента в дереві документа і застосовують до нього стиль в залежності від його статусу.

`:First-child` – Застосовується до першого дочірнього елементу селектора, який розташований в дереві елементів документа. Тобто тільки до першого абзацу, першого рисунку, першого посилання тощо.

Група псевдокласів, що задають мову тексту. Для документів, які одночасно містять тексти на декількох мовах, має значення дотримання правил синтаксису, характерні для тієї чи іншої мови. За допомогою псевдокласів можна змінювати стиль оформлення закордонних текстів, а також деякі настройки.

`:lang` – Визначає мову, який використовується в документі або його фрагменті. У коді HTML мова встановлюється через атрибут **lang**, він зазвичай додається до тегу `<html>`. За допомогою псевдокласу `:lang` можна задавати певні налаштування, характерні для різних мов, наприклад, вид лапок в цитатах. Синтаксис наступний.

```
Елемент: lang (мова) {...}
```

В якості мови можуть виступати наступні значення: **ru** - російська; **en** - англійська; **de** - німецька; **fr** - французька; **it** - італійська.

Коментарі CSS та позначення опису синтаксису

Для пояснення коду і для подальшого його редагування використовуються **коментарі**, які ігноруються браузерами. Коментар CSS починається символом `/*` і закінчується символом `*/`, як в прикладі нижче:

```
/ * Перший коментар. * /
TABLE
{
     / * Другий коментар. * /
     margin-left: 10pt;
     font-family: "sans serif"
          / * Третій коментар. * /
          border-style: outset;
}
```

Зверніть увагу, що при описі синтаксису застосовуються наступні позначення описі синтаксису:

- вертикальна риса між значеннями (наприклад, `fixed | scroll`) вказує на логічне виключає АБО, це означає, що треба вибрати тільки одне значення із запропонованих;
- подвійна вертикальна риса (`border-style || color`) позначає об'єднує АБО, кожне значення може використовуватися самостійно або спільно з іншими через пробіл;

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

- квадратні дужки (`[left | center | right]`) позначають групу, з якої, як правило, вибирається одне значення, причому воно не є обов'язковим до використання;
- два числа в фігурних дужках (`{a, b}`) говорять, що значення, котре їм передує слід повторювати не менше **a**, але не більше **b** раз.

4. Параметри тексту в CSS

Шрифтам у комп'ютерній графіці завжди приділялося багато уваги, і *World Wide Web* не є винятком. Але все багатство і різноманітність існуючих шрифтів для кирилічних шрифтів обмежена фактично трьома шрифтами: **serif** (зазвичай Times або інший шрифт із зарубками), **sans-serif** (Arial, Helvetica або інший шрифт без зарубок) і **monospace** (Courier). Якщо бути точним, то тут перераховані сімейства шрифтів. Зазвичай кожне з цих сімейств представлено лише одним кириличним шрифтом.

Автор HTML-сторінки для управління відображенням букв може застосувати кілька атрибутів, які впливають на шрифт:

- `font-family` - сімейство накреслень шрифту (гарнітура);
- `font-style` - пряме накреслення або курсив;
- `font-weight` - насиченість шрифту, "жирність" букв;
- `font-size` - розмір шрифту (кегль). Здається в пікселах (px) і друкарських пунктах (pt).
- `font-variant` - варіант накреслення (звичайний або дрібними буквами - капітель)

Гарнітура (`font-family`)

Встановлює сімейство шрифту, яке буде використовуватися для оформлення тексту. Список шрифтів може включати одне або кілька назв, розділених комою. Якщо в імені шрифту містяться пробіли, воно повинно полягати в одинарні або подвійні лапки

Універсальні сімейства шрифтів:

- **serif** - шрифти із зарубками, типу Times;
- **sans-serif** - шрифти без зарубок або гротески, типовий представник - Arial;
- **cursive** - курсивні шрифти;
- **fantasy** - декоративні шрифти;
- **monospace** - моноширинних шрифти, ширина кожного символу в такому сімействі однакова (шрифт Courier).

```
H1 {
    font-family: Geneva, Arial, Helvetica, sans-serif;
```

Перелік лекцій

```
        }
        P {
         font-family: Georgia, 'Times New Roman', Times,
serif;
        }
```

Кегль (`font-size`)

Кегль - це, якщо говорити спрощено, розмір шрифту. Більш детальне пояснення слід шукати в спеціальній літературі. Для спрощення досить знати, що CSS через параметр `font-size` дозволяє управляти розміром літер.

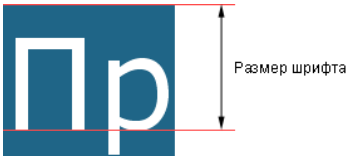
Синтаксис:

```
    font-size: абсолютний розмір | відносний розмір |
значення | проценти
```

Абсолютний розмір задається набором констант (**xx-small, x-small, small, medium, large, x-large, xx-large**). Інший набір констант (**larger, smaller**) встановлює відносні розміри шрифту.

Також дозволяється використовувати будь-які допустимі одиниці CSS: **em** (висота шрифту елемента), **ex** (висота символу x), пункти (**pt**), пікселі (**px**), відсотки (**%**) тощо. За 100% береться розмір шрифту батьківського елементу. Негативні значення не допускаються.

При установці розмір шрифту в друкарських пунктах (pt, 0,35 мм) або пікселах (px) слід пам'ятати, що `font-size` задається не висоту літери, а розмір "віконця" під букву, який трохи більше самої літери.




```
    H1 {
        font-family: 'Times New Roman', Times, serif; /*
Гарнитура тексту */
         font-size: 250%; /* Розмір шрифту в процентах */
        }
        P {
         font-family: Verdana, Arial, Helvetica, sans-serif;
         font-size: 11pt; /* Розмір шрифту в пунктах */
        }
```

Накреслення

У кожної гарнітури шрифту (`font-family`) є кілька накреслень. Кожне з них визначається в CSS трьома параметрами стилю: `font-style`, `font-stretch`, `font-weight`.

Атрибут стилю `font-style` визначає зображення шрифту – звичайне (**Normal**), курсивне (**Italic**) або похиле (**Oblique**).

```
    font-style: normal | italic | oblique
```

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Курсив і похилий шрифт при всій їх схожості не одне і те ж. Курсив це спеціальний шрифт імітує рукописний, похилий же утворюється шляхом нахилу звичайних знаків вправо Визначає.

```
     H1 {
         font-family: Verdana, Arial, Helvetica, sans-serif;
/* Рублений шрифт заголовку */
        }
        P {
         font-family: 'Times New Roman', Times, serif; /* Шрифт
з зарубками */
         font-style: italic; /* Курсивне накреслення */
        }
```

Атрибут стилю `font-weight` визначає насиченість шрифту.

```
     font-weight:
bold|bolder|lighter|normal|100|200|300|400|500|600|700|800|90
0
```

Насиченість шрифту задається за допомогою ключових слів: **bold** - напівжирний шрифт, **normal** - нормальне накреслення. Також допустимо використовувати умовні одиниці від 100 до 900. Значення **bolder** і **lighter** змінюють жирність щодо насиченості батьківського елементу, відповідно, в більшу і меншу сторону.

```
     H1 {
         font-weight: normal; /* Нормальне накреслення */
        }
        .select {
         color: maroon; /* Колір тексту */
         font-weight: 600; /* Напівжирне  накреслення */
        }
```

Атрибут стилю `font-stretch` встановлює вузьке, нормальне або широке накреслення шрифту, що дозволяє ущільнювати або розширювати текст.

```
     font-stretch:  ultra-condensed  |  extra-condensed  |
condensed | semi-condensed | normal | semi-expanded | expanded
| extra-expanded | ultra-expanded
```

Деякі інші атрибути стилю для управління оформленням тексту:

- підкреслення(перекреслення тексту).

```
     text-decoration:  [  line-through  ||  overline  ||
underline ] | none
```

- вирівнювання

```
     text-align: center | justify | left | right
```

- Відступ абзацу

```
     text-indent: <значення> | <проценти>
```

- між строковий інтервал

```
     line-height: множник | значення | проценти | normal
```

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

- інтервал між словами

```
word-spacing: <розмір> | normal
```

- інтервал між словами

```
letter-spacing: значення | normal
```

- перетворює текст елемента в заголовні або прописні символи

```
text-transform: capitalize | lowercase | uppercase | none
```

Для завдання усіх атрибутів одночасно існує скорочена запис:

```
font: [font-style||font-variant||font-weight] font-size [/line-height] font-family | inherit

P { font: bold 12pt/10pt sans-serif; }
```

5. Блокова модель

Характеристика фону та кольору тексту

У HTML для управління кольором тексту, що відображається використовується елемент **<font>**. Його аналогом в CSS є атрибут **color**. Цей атрибут можна застосовувати як для блокових, так і для строкових елементів розмітки. Каскадні таблиці стилів (CSS) у першу чергу описують властивості тексту. Це стосується як текстових блоків, так і строкових елементів розмітки змісту сторінки. Мова піде про управління відображенням кольору тексту (**color**) і кольору фону (**background-color**), на якому відображається текст.

Універсальна властивість **background** дозволяє встановити одночасно до п'яти характеристик фону. Значення можуть йти в будь-якому порядку, браузер сам визначить, яке з них відповідає потрібному властивості. Для детального ознайомлення дивіться інформацію про кожну властивість окремо.

background: [background-attachment || background-color || background-image || background-position || background-repeat]

Універсальна властивість **border** дозволяє одночасно встановити товщину, стиль і колір кордону навколо елемента. Значення можуть йти в будь-якому порядку, розділяючись пробілом, браузер сам визначить, яке з них відповідає потрібному властивості. Для установки кордону тільки на певних сторонах елементу, скористайтеся властивостями **border-top**, **border-bottom**, **border-left**, **border-right**.

```
border: [border-width || border-style || border-color]
```

```
border: 4px double black; /* Параметри границі */
```

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Блокова модель

В описі елементів розмітки мови HTML існує поняття строкового (**in-line**) елемента розмітки і блочного (**block**) елемента розмітки. Пояснити відмінність між блоковим і строковим елементами можна на прикладі:

- параграф - це блоковий елемент розмітки;
- виділення курсивом - це строковий елемент розмітки.

Блокові елементи можна вкладати один в одного, але вони не повинні перетинатися. Строкові елементи можна як вкладати, так і перетинати, але останнє робити не рекомендується. Очевидно, що за набором атрибутів управління відображенням строкові і блокові елементи відрізняються. Спрощено можна сказати, що атрибути опису стилю строкового елементу є підмножиною атрибутів опису стилю блокового елемента.

Узагальненнями блочного і строкового елементів, з точки зору стилів, є теги **<DIV>** та **<SPAN>**, відповідно.

Блокова модель дозволяє оперувати з текстом в термінах прямокутників, які цей текст займають та обробляти усі теги як невеликі блоки, для них будь-який тег - контейнер з вмістом: текстом, зображеннями, іншими тегами і т.д.

Існують наступні атрибути для блокової моделі (рис. 8.3):

Рис. 8.3 Атрибути блокової моделі

- **Margin** (зовнішній відступ) - порожній простір (поле), яке відокремлює один елемент від іншого. Найпростіший приклад зовнішнього відступу - проміжок між абзацами, що йдуть один за одним. Область зовнішнього відступу прозора і не може мати свій колір і інших ефектів оформлення.

- **Border** (рамка елемента) – границя з будь-якого боку елемента. Рамка може бути встановлена як з усіх боків елемента, так і з одного боку. Використовуючи рамку можна просто красиво оформити елемент або візуально відокремити вміст поточного елемента від інших елементів сторінки.

- **Padding** (внутрішній відступ) - порожній простір між вмістом елемента і його рамкою. Область внутрішнього відступу прозора і не може мати свій колір або інші ефекти оформлення.

У більшості елементів є область вмісту, в якій розташовується весь вміст елемента (текст, вставляються зображення або інші елементи).

За замовчуванням для блокових елементів використовується автоматична ширина. Це означає, що елемент буде розтягнутий рівно на стільки, скільки є вільного

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

місця. Висота блокових елементів за замовчуванням встановлюється автоматично, тобто браузер розтягує область вмісту в вертикальному напрямку так, щоб відобразилося весь вміст. Щоб встановити власні розміри для області вмісту елемента, ви можете використовувати атрибути **`width`** (дозволяє задати ширину області вмісту елемента) та **`height`** (висота області вмісту) (рис. 8.4).

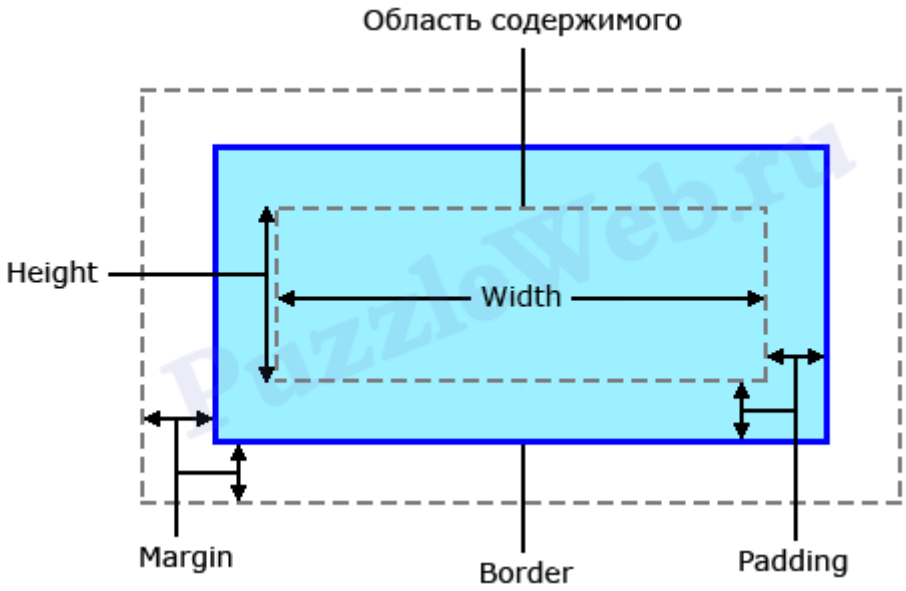


Рис. 8.4 Атрибути `width` та `height` блокової моделі

Треба зверніть увагу, що властивості **`width`** і **`height`** визначають розмір тільки області вмісту, щоб обчислити загальну ширину блочного елемента (рис. 8.5), потрібно скласти ширину області вмісту, лівого і правого внутрішніх відступів і ширину лівої і правої рамки. Те ж саме стосується і загальної висоти елемента, тільки все значення обчислюються по вертикалі:

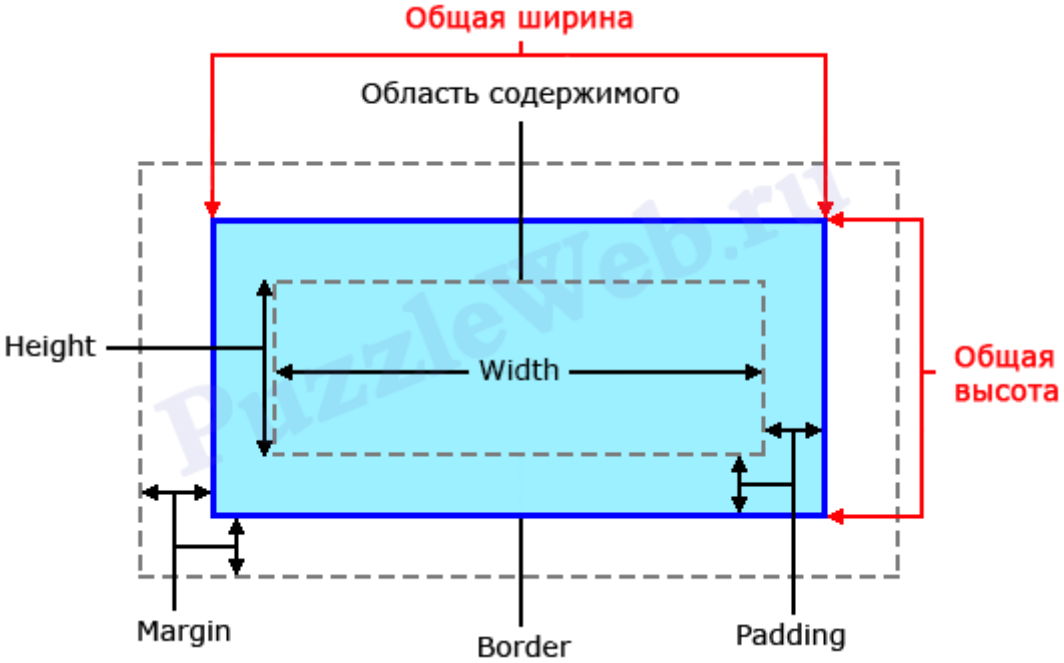


Рис. 8.5 Обчислення загальної ширини та висоти блочного елементу

Після того як ви визначили для елемента ширину і висоту, варто звернути увагу на один важливий момент - вміст, розташоване всередині елемента, може перевищити зазначений розмір блоку. У цьому випадку частина вмісту буде виходити

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

за межі елементу, щоб уникнути цього неприємного моменту, можна скористатися властивістю **overflow**. Властивість **overflow** повідомляє браузеру, як потрібно вчинити в тому випадку, якщо вміст блоку перевищує його розмір. Це властивість може приймати одне з чотирьох значень:

- **visible** - значення, яке використовується браузером за замовчуванням. Вказівка цього значення буде мати той же ефект, що і відсутність установки властивості overflow.
- **scroll** - додає до елементу смуги прокрутки по вертикалі і горизонталі.
- **auto** - додає смуги прокрутки при необхідності.
- **hidden** - приховує частину вмісту, яке виходить за межі блочного елемента.

Обтікання блоку тексту

Під обтіканням блоку текстом розуміють той же ефект, який можна реалізувати для графіки, коли картинка не розриває блок тексту, а вбудовується в нього. Текст в цьому випадку "обтікає" картинку з одного боку - там, де є вільне поле між межею сторінки (елемента) і картинкою. Обтікання картинки текстом від звичайного вбудовування картинки в текст документа відрізняється тим, що уздовж вертикальної межі картинки розташовується кілька рядків тексту, а не одна.

Обтіканням блоку тексту іншим текстом керують два атрибути CSS: **float** і **clear.**

Атрибут **float** визначає "плаваючий" блок тексту. Він може приймати значення:

- **left** - блок притиснутий до лівої кордоні охоплює елемента;
- **right** - блок притиснутий до правої кордоні охоплює елемента;
- **both** - текст може обтікати блок з обох сторін.

Другий атрибут опису стилів **clear** дозволяє управляти власне обтіканням. Він не допускає наявності "плаваючих" блоків близько блоку тексту та може приймати значення: **right**, **left**, **none**, **both.**

Позиціонування блоків

До появи CSS єдиним засобом щодо точного позиціонування були таблиці. Вони дозволяли досить точно розташувати компоненти HTML-сторінки відносно один одного на площині. CSS дозволяє не тільки точно розмістити елемент розмітки щодо інших компонентів сторінки, але і щодо меж сторінки.

Стандарт CSS дозволяє з точністю до пікселя розмістити блоковий елемент розмітки в робочому полі вікна браузера. При такому підході виникає природне запитання: як влаштована система координат, в якій розробник сторінки розміщує її компоненти.

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

CSS визначає дві системи координат: **відносну** і **абсолютну**. Це дозволяє забезпечити гнучкість розміщення елементів як щодо кордонів робочого поля вікна браузера, так і відносно один одного.

При використанні **абсолютних** координат точка відліку поміщається у верхній лівий кут батьківського блоку (наприклад, вікна браузера), а осі X і Y спрямовані вправо по горизонталі (вісь X) і вниз по вертикалі (вісь Y), відповідно:

Якщо в цій системі координат деякий блоковий елемент повинен бути розміщений на 10 px нижче верхнього обріза робочої області браузера і на 20 px правіше лівого краю робочої області браузера, то його опис буде виглядати наступним чином:

```
.example
{ position: absolute; top: 10px; left: 20px; }
```

В даному записі тип системи координат задано атрибутом `position` (значення - **absolute**), координата X задана атрибутом `left` (значення - 20 px), координата Y - атрибутом `top` (значення - 10 px).

Атрибути top і left визначають координати верхнього лівого кута блоку в абсолютній системі координат. Значення координат можуть бути і негативними. Для того, щоб приховати з області блок з лінійними розмірами 100 px (висота) на 200 px (ширина), досить позиціонувати його наступним чином: (відкрити)

```
.example {position: absolute;
          top: -100px; left: -200px;
          width: 200px; height: 100px; }
```

При використанні **відносних** координат система дозволяє розмістити блоки на сторінці в координатах блоку, що їх охоплює. Переваги такої системи координат очевидні: вона дозволяє зберігати взаємне розташування елементів розмітки при будь-якому розмірі вікна браузера і його настройках за умовчанням.

В якості точки відліку в цій системі координат вибрана точка розміщення поточного блоку за замовчуванням. Ось X при цьому спрямована горизонтально вправо, а вісь Y - вертикально вниз.

Щоб задати координати блоку, в цій системі застосовують запис типу:

```
<DIV STYLE = "border-width: 1px; border-style: solid;
width: 100%; height: 100px; ">
<DIV STYLE = "position: relative; top: 0px;
left: 0px; border-width: 1px; ">
Цей блок знаходиться в точці відліку відносних координат
</ DIV>
<DIV STYLE = "position: relative; top: 0px; left: 50px;
border-width: 1px; ">
А цей блок зміщений вправо на 50px
</DIV> </DIV>
```

Для роботи з відносною системою координат краще користуватися універсальними блоками DIV. У відносній системі координат також можна користуватися негативним значенням позиції розміщення.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Абсолютне і відносне позиціонування блоків дозволяє надати сторінці третій вимір. Координати блоку, а точніше, лівого верхнього кута прямокутника блоку, і його лінійні розміри визначають місце розташування блоку на площині. Оскільки блок не є абстрактною точкою, прямокутники блоків перекриваються. Причому в певному порядку, що дозволяє говорити про їхнє розміщення в просторі. Порядок накладення (перекриття) блоків визначається атрибутом **z-index**.

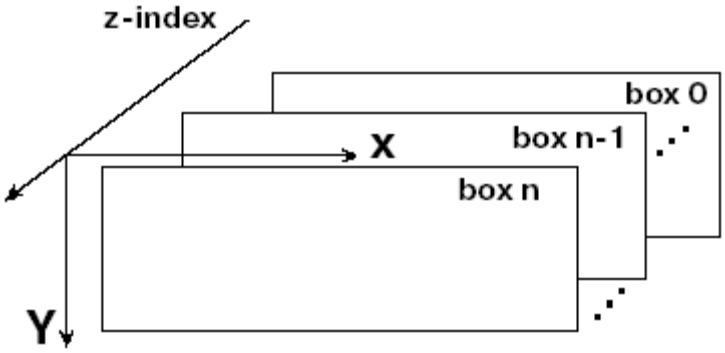


Рис. 8.6 Позиціонування блоків атрибутом z-index

Якщо бути більш точним, то блоки - це прямокутні картки, які лежать на поверхні, перекриваючи один одного. Відстань між ними поставити не можна, тобто повноцінного третього виміру немає Порядок перекриття блоків визначається атрибутом **z-index**, чим більше значення **z-index**, тим ближче до спостерігача знаходиться блок. Значення **z-index** може приймати і негативні значення.

Питання для самоконтролю

1. Що таке CSS? Для чого він призначений?
2. Які існують способи підключення таблиці стилів, їх переваги та області застосування?
3. Укажіть загальні правила пріоритету використання стилів.
4. Вкажіть основний синтаксис CSS. Що таке селектор?
5. Селектор класу, ідентифікатора, груповий селектор та псевдо селектор – їх відмінності та області застосування.
6. Що таке селектор нащадків?
7. Наведіть основні властивості для форматування параметрів тексту (шрифт та накреслення).
8. Які властивості для зміни відображення тексту (оступи вирівнювання, форматування)?
9. Властивість border – синтаксис та параметри застосування.
10. Властивість background синтаксис та параметри застосування.
11. Що таке блокова модель у CSS?
12. Наведіть основні властивості для зміни параметрів блоку.
13. Z-index, для чого він використовується?

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

VBA
Ведення в об'єктно-орієнтовне програмування. Мова програмування VBA

Зміст

VBA Ведення в об'єктно-орієнтовне програмування. Мова програмування VBA 161
1. *Загальні поняття про Visual Basic For Application* 161
Код VBA 162
Типи даних мови VBA 164
2. *Елементи мови VBA. Опис змінних і констант* 166
Опис змінних і констант 166
Операції VBA 168
Стандартні функції 169
Вирази, оператор присвоювання та коментаря 169
3. *Організація вводу та виводу інформації* 171
Функції перетворення типів Str() і Val() 171
Обмін інформацією за допомогою діалогових вікон 172
4. *Реалізація основних алгоритмічних конструкцій на мові VBA* 173
Програмування алгоритмів лінійної структури 173
Програмування розгалужуваний обчислювальних процесів 174
Програмування алгоритмів циклічної структури 177
5. *Процедури* 180
Питання для самоконтролю 183

1. *Загальні поняття про Visual Basic For Application*

Visual Basic For Application (VBA – Visual Basic для застосунків) реалізований у чотирьох найважливіших компонентах пакета Microsoft Office. VBA являє собою повністю уніфіковану мову програмування й майже повністю уніфіковане середовище розробки.

VBA містить у собі основні конструкції мови Visual Basic для Windows, вбудований в MS Office та у додатки Office. VBA є загальним для всіх додатків MS Office. В VBA застосовується об'єктно-орієнтований підхід до розробки додатків. Кожний додаток – Word, Excel, Access, PowerPoint та ін.- доповнює засоби VBA новими функціональними можливостями за допомогою об'єктів, властивостей і методів, властивих цьому додатку. Як об'єктна–орієнтована мова програмування VBA використовує наступні основні поняття: об'єкт, властивість, метод, подія, клас і сімейство об'єктів. Введемо визначення.

Об'єкт – це об'єднання даних з кодом, призначеним для їхньої обробки. Об'єкти застосовуються в програмуванні, тому що вони представляють предмети, про внутрішню організацію яких і про те, як вони зроблені, програмістові знати не обов'язково, але він може ними користуватися й управляти.

Властивості – це параметри об'єкта, які йому властиві. Вони характеризують об'єкт. Усі властивості об'єкта мають свої імена. Різні об'єкти характеризуються різними властивостями, наприклад, об'єкт «лінія» характеризується довжиною й

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

кольором, а об'єкт «коло» радіусом і кольором. Синтаксис завдання властивості об'єкту має вигляд

ОБ'ЄКТ. ВЛАСТИВІСТЬ = ЗначенняВластивості

```
'змінити ім'я поточного аркуша
Activesheets.Name = "Січень"
```

Метод – це програма, яка впливає на об'єкти і їх параметри. За допомогою методів можна, наприклад, закрити об'єкт, вилучити його, або показати на екрані. Синтаксис запису такий:

ОБ'ЄКТ.МЕТОД

```
'вилучити аркуш
Worksheets("Огляд").Delete
```

Подія – це дія, що розпізнається об'єктом, для якого можна запрограмувати відгук (наприклад, клацання мишею по рядкові меню або натискання клавіші). При настанні події відбувається активізація методу, тобто починають виконуватися пов'язані з об'єктом програми й функції.

Клас – це шаблон, на основі якого під час виконання програми створюється об'єкт. Клас визначає властивості й методи об'єкта. Якщо підібрати приклад з життя, то в якості класу об'єктів можна прийняти навчальний клас, де об'єкти – це учні, кожний зі своїми, властивими тільки йому властивостями. Також кожному об'єкту в класі властиві одне або кілька однакових властивостей, наприклад – вік, по якім і визначається приналежність об'єкта до класу. Об'єкт, який створюється на основі класу, називають екземпляром класу.

Отже, усі додатки Microsoft Office містять макромову й засіб записи макросів. Усі макрокоманди VBA включаються в проекти, які можуть належати різним документам. При створенні проекту використовується один із трьох методів:

- запис макросу за допомогою макрорекодера, з наступним його редагуванням,
- створення макросу в редакторі VBA,
- створення проекту в редакторі VBA.

Код VBA

Код VBA зберігається в проектах. Проекти містять модулі, а модулі включають процедури й функції. Процедури й функції містять опис й інструкції мовою VBA.

Кожний додаток MS Office зберігає проекти по у своїх файлах, при цьому при зберіганні документу необхідно вибрати тип з підтримкою макросів:

- Word зберігає проекти в шаблонах документа (DOTM) і в документах з підтримкою макросів (DOCM).
- Excel зберігає проекти в робочих книгах з підтримкою макросів(XLSM).
- Access зберігає проекти у файлах бази даних (ACCDB).
- PowerPoint зберігає проекти в презентаціях з підтримкою макросів (PPTM).

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Для програмування на VBA немає необхідності в обов'язковому порядку встановлювати яку-небудь версію транслятора VB (Visual Basic). Цілком достатньо, якщо на комп'ютері буде встановлений Microsoft Office. Для створення програми, яка називається макросом, необхідно завантажити Microsoft Word або Microsoft Excel та перейти у середовище розробки за допомогою комбінації клавіш **Alt+F11**.

> Для роботи з макросами Microsoft Office використовується спеціальна вкладка – **Розробник**, містить засоби створення макросів і форм, а також функції для роботи з .xml. Але за замовчуванням ця вкладка не відображається на стрічці. Для її відображення необхідно виконати команду Файл – Параметри – Налаштувати стрічку та встановити відповідний прапорець.

При відкритті редактора Visual Basic за замовчуванням разом з ним відкриваються два допоміжних вікна - **вікно проекту** (Project Explorer Windows) і **вікно властивостей** (Properties Windows), а також як мінімум одне **вікно модуля** (Code Window) (рис. 9.1).

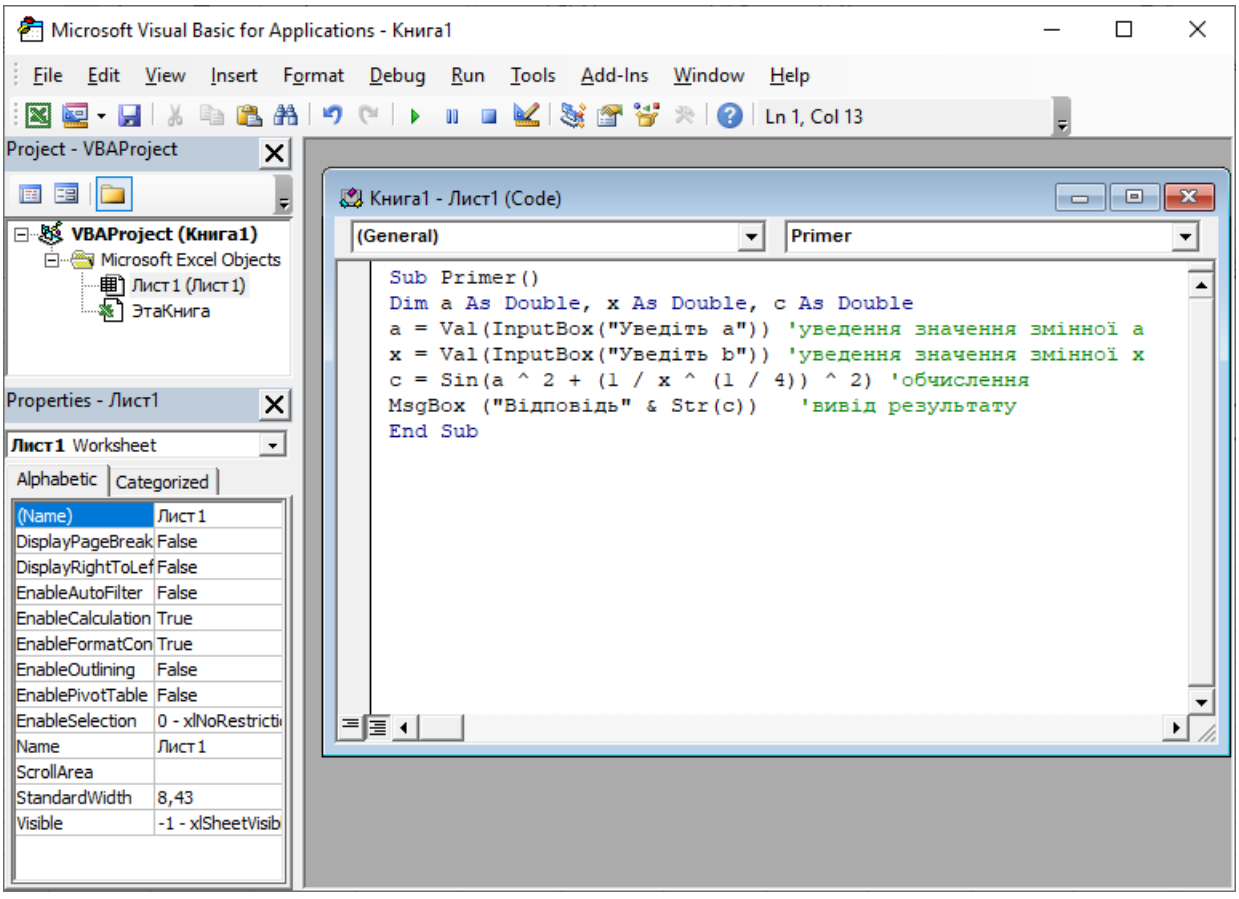


Рис. 9.1 Вікно редактора Visual Basic For Application

Як уже згадувалося вище, редактор Visual Basic є окремим додатком, яке має свою рядок меню і панель інструментів. Крім того, до його складу входять сім допоміжних вікон, аби допомогти курувати проектом, виконувати навігацію і пошук об'єктів, їх властивостей, методів, а також налагодження та оптимізацію програмних процедур. Серед цих вікон наступні: **перегляду об'єктів** (Object Browser Window), **безпосереднього налагодження** (Immediate Window), **локальних змінних** (Locals Window), **контрольних виразів** (Watch Window), **стека викликів** (Call Stack Window).

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Допоміжні вікна доступні через меню **Вид** (View), відповідні кнопки панелі інструментів і пункти контекстного меню.

Якщо програма складається тільки з однієї процедури, то в найпростішому випадку першим рядком записується оператор оголошення процедури **Sub ім'я()**. Це стандартний заголовок макросу. Завершальним рядком процедури є рядок **End Sub**.

```
Sub ім'я()
   Оператори
End Sub
```

Кожний оператор програми записують в окремому рядку і для введення рядка натискають клавішу **Enter**. Допускається запис декількох операторів в одному рядку (при цьому один оператор від іншого відділяється двома крапками), однак це не рекомендується, тому що програму важче налагоджувати та редагувати.

Якщо рядок програми занадто довгий, то використовують один з наступних варіантів:

- продовжують писати на тому ж рядку, допускаючи, щоб частина програми перебувала за межами екрану;
- вводять символи продовження рядка - пробіл і знак підкреслення.

Усередині програм можна поміщати коментарі. При цьому кожний рядок коментаря починається зі знаку апострофа.

Для запуску програми з вікна редактора використовується клавіша **F5**. Для запуску програми з документа — **Alt+F8**. Для контролю проміжних результатів програму можна виконати в покроковому режимі. Для цього курсор установлюють у межах тексту програми, яку потрібно протестувати, і натискають клавішу **F8**. Після кожного натискання клавіші **F8** програма переходить до чергового рядка тексту. Після тестування довідатися поточні значення змінних допоможуть спливаючі підказки. Для цього курсор миші варто помістити на ім'я змінної і на екрані відобразиться її значення.

Типи даних мови VBA

Visual Basic має 11 вбудованих типів даних:

Тип даних		Розмір (байт)	Розрядність цифр	Діапазон значень
Boolean	логічний	2	1	True або False (так - ні, включений- відключений)
Integer	ціле	2	5	Від (-32768) до 32768
Long	довге ціле	4	10	Від (-2147483648) до 2147483647
Single	дійсне одинарної точності	4	7	Від (–3.402823E+38) до (–1.401298E-45) і від 1.401298E-45 до 3.402823E+38
Double	дійсне подвійної точності	8	15	Від ± 1.79769313486232E+308 до ± 4.94065645841247E-324

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Currency	грошовий	8	19	Від –922337203685477.5808) до 922337203685477.5807
Date	для дат і часу	8	-	Від 01.01.100 до 31.12.9999
String	символьний	1	1	на будь-який символ - Від 0 до 65535 символів
Object		4	-	Будь-який об'єкт
Array	масиви	Визначається кількістю й розміром елементів	-	-
Variant		Визначається записаними даними	-	Кожної із вбудованих типів даних

Тип даних визначає, яким образом величина, що задається, зберігається в пам'яті комп'ютера; скільки розрядів може містити, що привласнюється їй значення; наскільки воно може бути велике й чи може мати дробову частину.

Boolean використовує для зберігання значення 2 байта пам'яті – як і тип Integer, хоча може приймати значення або 0 (неправда - False), або 1 (True – правда).

Integer призначений для зберігання цілого числа з вказівкою знаку. З 16 біт 1 використовується для зберігання знаку, а інші – для значущих цифр.

Long використовується для зберігання цілого числа з подвійним розміром у порівнянні з **Integer** і, відповідно, займає у два рази більше пам'яті. З 32, займаних у пам'яті біт, 1 призначає для зберігання знаку, 31 – для зберігання самого числа.

Single призначений для зберігання дійсного числа із плаваючою крапкою одинарної точності (до 7 значущих цифр)- числовий тип одинарної точності (до 7 значущих цифр), відповідно **Double** – подвійної точності (до 14 значущих цифр) стосовно **Single**.

Currency призначений для зберігання дійсного числа з фіксованою крапкою, яка завжди розташовується між 4 і 5 цифрами праворуч. Цей тип спеціально призначений для одержання мінімальних помилок округлення й найчастіше використовується в бухгалтерських розрахунках.

Date призначений для зберігання дат і часу з точністю до секунди. Дати записуються в такий спосіб: ціла частина – кількість днів, що пройшли від початку відліку 01.01.1900, а дрібна частина – час. VBA дозволяє оперувати датами в діапазоні 01.01.100 – 31.12.9999.

String використовується для зберігання тексту по одному байту на символ, додаючи до цієї кількості ще один – для позначення кінця рядка.

Object призначений для посилань на будь-який об'єкт VBA.

Array визначає індексний список величин одного й інших типів - масив.

Тип **Variant** використовується за замовчуванням. Він задається VBA, якщо користувач не вкажіть інший тип. При цьому проводиться перевірка значення, що вводиться і його ідентифікація, тому різко зростає й час виконання програми, і обсяг пам'яті.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

2. *Елементи мови VBA. Опис змінних і констант*

Опис змінних і констант

При написанні коду VBA слід знати деякі правила або угоди, які слід дотримуватися, щоб уникнути помилок виконання коду. Перша угода — це угода по іменам. Іменувати доведеться змінні, константи, процедури, функції, об'єкти тощо, тому необхідно дотримуватися наступних правил.

- Довжина імен не повинна перевищувати 255 символів та повинно бути унікальним усередині області дії.
- Імена повинні починатися з букви й можуть містити комбінацію букв, цифр і символу підкреслення (_)
- Імена не повинні збігатися зі стандартними іменами Visual Basic.
- Імена не повинні включати **крапок**, **пробілів** і наступних символів**: !** @ # **$ %.**

Символ пробілу можна замінити символом підкреслення _. Регістр букв в іменах не має значення, тобто наступні імена **PRICE**, **Price** й **price** — еквівалентні.

```
Приклади правильного запису імен:
     H123, Price, Sale
Неправильно записані імена:
     21days – починається із цифри;
     Price Sale –містить символ пробілу;
     Dim – збігається зі службовим словом мови.
```

Дамо також кілька визначень елементам мови VBA.

Ключові слова — це слово або символ, які є елементом мови програмування Visual Basic, це може бути інструкція, ім'я функції, оператор. Наприклад: **SUB**.

Вираз — це комбінація ключових слів, операцій, констант. Результатом вираження може бути рядок, логічне значення або число. Приклади виразів : X+3; Z<>0; 11.

Оператори використовуються для об'єднання простих виразів у більш складні. Оператори VBA виконують арифметичні, логічні операції, а також операції порівняння й конкатенації (зчеплення рядків).

Змінна — це область пам'яті, призначена для тимчасового зберігання даних, і якої привласнено ім'я. Вміст змінної може бути змінене в процесі виконання програми. Дані, збережені в змінних, можуть бути різного типу: числові дані, строкові, логічні тощо. Тип даних визначає тип змінної.

Синтаксис опису змінних:

Dim Им'я Змінної As тип, …, Им'я Змінної As тип

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

```
Наприклад:
   Dim  k As String 'змінна k описаний як символьна
   Dim  x As Single 'змінна x описана як дійсна одинарної
                    точності
   Dim  I As Integer змінна I описана як ціла
   Dim  n As Integer, s As Double 'змінна n описана як ціла і
                        змінна s як дійсна подвоєної точності
```

В VBA є три рівні визначення змінних. Рівень процедури. Змінні використовуються тільки в процедурі, у якій вони описані. Змінна описується за допомогою інструкції **DIM**, розміщеної в тілі процедури. Рівень модуля. Змінні використовуються тільки в процедурах модуля, у яких вони описані, але не в інших модулях проекту. Змінна описується за допомогою інструкції **DIM**, розміщеної в розділі описів модуля. Загальні змінні. Змінні використовуються у всіх модулях і процедурах проекту. Змінна описується за допомогою інструкції **PUBLIC**, розміщеної в розділі описів модуля.

Змінні рівнів процедури зберігають своє значення лише при виконанні процедури, при виході із процедури значення змінної губиться. Якщо процедура викликає інші процедури, змінна зберігає своє значення, поки не закінчиться виконання й цих процедур. Якщо змінна рівня процедури описана за допомогою ключового слова **Static**, вона зберігає своє значення доти, поки програма виконується в якому-небудь модулі. По завершенню роботи всієї програми змінна втрачає свою область визначення й своє значення. Її час життя збігається із часом життя змінної рівня модуля.

Строго кажучи, в VBA можна не описувати більшість змінних перед їхнім застосуванням. У цьому випадку вони називаються неявно описаними, і їм автоматично присвоюється тип **Variant**. Але слід пам'ятати, що на зберігання даних типу **Variant** затрачається більше пам'яті і швидкість їхньої обробки менша.

При явному описі змінних (тобто усі зміні, що використовуються повинні бути заздалегідь описані користувачем) на початку модуля повинна бути інструкція **Option Explicit** (вона з'являється автоматично при відкритті вікна модуля). Інструкція **Option Explicit** має наступні особливості - якщо неправильно вказати ім'я описаної змінної або взагалі її не описувати, то при компіляції видається повідомлення про помилку.

Константа — це іменована область пам'яті для зберігання даних, зміна яких під час роботи програми не допускається. Константа може мати числовий, логічний або строковий тип.

Константа описується ключовим словом **CONST**, і при описі константи потрібно привласнити їй значення. Спроба змінити в ході виконання програми значення константи викликає помилку.

```
Const Pi as Double = 3.1415926 'описана константа Pi
Const Name = "Visual Basic"    'описана константа строкового
                                типу.
```

Масиви – упорядкована сукупність даних одного типу. Масиви можуть мати одну, дві й більше розмірності (але не більше 60). Наприклад, $X=(x_i)$, i=0,1,2,…10 –

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

одновимірний масив з 11 елементів; A=(a_{ij}), i=0,1,2,…,5; j=0,1,2,…,6; - двовимірний масив з 6x7 елементів. Елемент масиву – це змінна з індексами. Індекси в VBA обмежуються круглими дужками й відділяються один від одного комами. Наприклад, $x_i \rightarrow x(i)$; $a_{ij} \rightarrow a(i,j)$.

У найпростішому випадку синтаксис опису масиву фіксованого розміру має вигляд:

Dim им'я Массиву (n1,n2,…,nk) As тип

де n1,n2,…,nk–максимальні значення відповідно 1-го, 2-го, ..., k-го індексів, мінімальне значення індексу вважається рівним 0.

```
Наприклад
  Dim X (19)   As String
  Dim Y (3,7)  As Single
```

Тут описаний строковий масив X, що складається з 20 елементів і числовий масив Y, що складається з 4 рядків і 8 стовпців. Щоб нумерація індексів елементів починалася з 1, необхідно на початку модуля ввести команду **Option Base 1**.

Операції VBA

Операції проводяться над виразами, змінними, константами, символами та умовно розділяються на три основні категорії: **арифметичні**, **порівняння** і **логічні**.

Арифметичні операції

Операція	Назва	Приклад	Результат
+	Додавання	10+5	15
-	Віднімання	10-5	5
*****	Множення	10*5	50
/	Ділення	10/4	2.5
****	Цілочисленне ділення	10\4	2
%	Відсотки	10%	0.1
^	Піднесення до ступеня	2 ^ 3	8
Mod	Залишок від ділення	10 mod 3	1

Операції порівняння

Операції порівняння використовуються в командах при порівнянні двох або більше числових, строкових величин. Якщо результат порівняння вірний, то результату присвоюється логічне значення **True** (Істина). Якщо результат порівняння невірний, то результату присвоюється логічне значення **False** (Неправда).

Оператор	Ім'я	Приклад	Результат
=	Дорівнює	10=5	Неправда
>	Більше	10>5	Істина
<	Менше	10<5	Неправда
>=	Більше або дорівнює	"a">="b"	Неправда
<=	Менше або дорівнює	"a"<="b"	Істина
<>	Не дорівнює	"a"<>"b"	Істина

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Логічні операції

Логічні операції використовуються для побудови логічних виразів.

Оператор	Загальний вид	Результат
And	Вираз1 **And** вираз2	Повертає значення "Істина", якщо вірні вираз1 і вираз2. У протилежному випадку повертає "Неправда".
Or	Вираз1 **Or** вираз2	Повертає значення "Істина", якщо хоча б один з виразів вірний. Значення "Неправда" буде тільки у випадку, якщо обидва вирази помилкові.
Xor	Вираз1 **Xor** вираз2	Повертає значення "Істина", якщо тільки один з виразів вірний. Якщо вірні обидва вирази або помилкові обидва вирази, то повертає значення "Неправда".
Not	**Not** вираз	Повертає значення "Неправда", якщо вираз вірний, і значення "Істина", якщо вираз невірний.

Стандартні функції

Для обчислення стандартних функцій необхідно вказувати ім'я функції і у дужках аргумент. Перелік основних стандартних функцій:

Функція	Призначення	Дія, що виконується (Аргумент)
Abs(x)	$\|x\|$	Повернення абсолютного значення числа (по модулю). (Будь-який числовий вира)з
Sqr(x)	$\sqrt{x}$	Повернення квадратного коріння числа (Будь-який числовий вираз >0)
Exp(x)	e^x	Повернення значення ex (Будь-який числовий вираз <=709.7827
Log(x)	Ln x	Повернення натурального логарифма числа (Будь-який числовий вираз >0)
Sin(x)	Sin x	Повернення синуса кута (Аргумент у радіанах)
Cos(x)	Cos x	Повернення косинуса кута (Аргумент у радіанах)
Tan(x)	Tg x	Повернення тангенса кута (Аргумент у радіанах)
Atn(x)	Arctg x	Повернення арктангенса кута (Будь-який числовий вираз. Повертає значення в радіанах)
Int(x)		Відкидання дрібної частини (Будь-який числовий вираз)

Вирази, оператор присвоювання та коментаря

В VBA можна виділити арифметичні, логічні й строкові вирази.

Арифметичні вирази - це сукупність числових констант, змінних, функцій, зв'язаних арифметичними операціями: +, -, *, /, ^, \, mod.

Наприклад:

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

```
sin(x)+log(y)/a^3*2.7
```

Загальні правила запису арифметичних виразів на мові VBA.

- Усі арифметичні операції записуються в один рядок. Якщо в чисельнику або знаменнику дроби присутній багаточлен, його слід взяти в дужки.
- При записі десяткових дробі для відділення цілої частини від дробової використовується десяткова крапка.
- Знаки арифметичних операцій опускати не можна.
- Якщо в десятковій дробі немає цілої частини, то нуль перед десятковою крапкою можна вилучити.

Арифметичні операції виконуються зліва направо у порядку пріоритету. Для зміни природнього пріоритету операцій застосовуються дужки (при цьому кількість дужок, що відкриваються й закриваються у виразі повинна бути однаковою).

Логічні вирази - це сукупність логічних констант, логічних змінних, порівнянь, функцій логічного типу, з'єднаних логічними операціями. Результатом обчислення може бути **True** або **False**.

```
Наприклад:
   x < a+c
   x*y < 0  and  a+b>0
```

Строкові (текстові) вирази - сукупність строкових констант, строкових змінних, строкових функцій, з'єднаних операцією об'єднання **&**. Строкова константа (рядок) - це послідовність символів, розташованих в лапках. Строкові змінні (мають тип **String**) приймають значення строкових констант.

```
Наприклад:
    "Мені  " & "двадцять  " &  "років"
    Dim a As String, b As String, c As String
      a="Мені  "
      b="двадцять років"
      c=a & b
```

Якщо вираз містить операції різних типів, то першими виконуються арифметичні операції, потім - операції порівняння, останніми - логічні операції.

Оператор присвоювання

Синтаксис оператора: **ім'я змінної = вираз.** Вираз може бути арифметичним, логічним, строковим. Оператор обчислює вираз, потім отримане значення присвоюється змінній.

```
Наприклад:
   c = a+b,
   x = y,
   a = 2.5,
   s="студент"
```

В операторі присвоювання змінна в лівій частині і вираз в правій частині повинні бути одного типу. Однак, якщо змінна в лівій частині описана як змінна типу **Variant**, то вона прийме тип виразу правої частини.

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Оператор коментарів

Оператор коментарів не виконує в програмі ніяких дій і може містити будь-який текст. Інструкція коментарів має два види: це інструкція, що починається із ключового слова **REM**, за яким іде довільний текст, або знак апострофа ('), який ставиться на початку рядка або після будь-якого оператора, і за ним також можна писати будь-який текст.

```
Наприклад:
REM довільний текст коментаря
A1=5      'коментар до рядка оператора
```

3. *Організація вводу та виводу інформації*

Функції перетворення типів Str() і Val()

Функція **Str** (**Str**ing) використовується для приведення числового виразу типу **Long** в рядок (тип **String**).

Функція **Str** повертає значення **Число**, перетворене в текстовий тип даних **String**. При перетворенні на початку рядка повертається резервується місце для знаку числа. Якщо число позитивно, то в цьому місці буде пробіл, якщо число негативно, то виводиться знак мінус.

У Якості десяткового роздільника функція **Str** сприймає тільки точку. При використанні інших десяткових роздільників (наприклад, коми) слід використовувати функцію **CStr().**

```
Наприклад
Dim retval As String
retval = Str (123)          'отримуємо "123"
retval = Str (-12.3)        'отримуємо "-12.3"
```

Функція **Val** (**Val**ue) служить для перетворення аргументу в числовий тип даних. Функція **Val** припиняє читання рядка на першому символі, який вона не може розпізнати як частини числа. Символи, які часто розглядаються як елементи числових значень, типу знаке долара і коми, не розпізнаються. Однак, ця функція розпізнає префікс підстави &O (для вісімкових значень) і &H (для шістнадцятирічних значень). Пробілі, символи табуляції і символи розриву рядків видаляються з значення параметра. Функція **Val** розпізнає як роздільник цілої та дробової частини тільки точку. Якщо використовуються інші роздільники цілої та дробової частин (наприклад в національних версіях додатків), слід застосовувати для перетворення рядка в число функцію **CDbl**

```
Наприклад
'В наступному прикладі повертається
'Числове значення 1615198
Print Val ( "1615 198-я вул.  Свободи.")
```


©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Обмін інформацією за допомогою діалогових вікон

При створенні макрокоманд потрібно організувати обмін інформацією між макросом і користувачем. Це завдання вирішується за допомогою функцій **Inputbox** та **Msgbox**.

Функція Inputbox. Виводить на екран діалогове вікно, що містить повідомлення й поле введення, установлює режим очікування введення тексту користувачем або натискання кнопки, а потім повертає значення типу String, що містить текст, уведений у поле. Синтаксис функції:

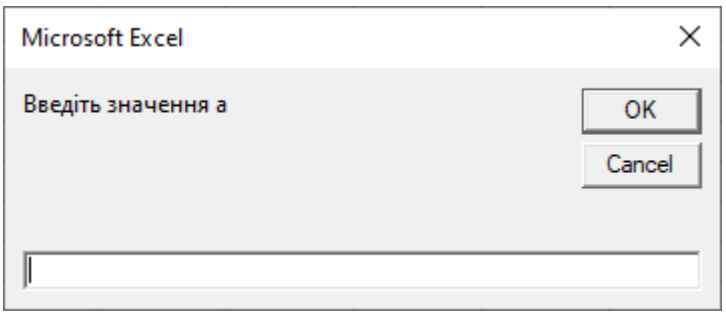
InputBox (запрошення[, заголовок][, по умовчанню][, гор][, верт][, файл допомоги[, контекст])

У квадратних дужках позначені ті елементи інструкції, які не являються обов'язковими для запису.

де **запрошення** - повідомлення, що виводиться в діалоговому вікні; **заголовок** - текст, що з'являється в панелі заголовка діалогового вікна; **по умовчанню** - значення, що вводиться по умовчанню, у текстовому вікні; **гор** - горизонтальне розташування та **верт** - вертикальне розташування діалогового вікна стосовно лівої межі екрана. Якщо цей аргумент опущений, то діалогове вікно розміщається в центрі; **файл допомоги** - вказівка на файл довідки; **контекст** - число, що вказує на розділ довідки.

Усі параметри, крім першого, є необов'язковими. Функція **InputBox** повертає рядок символів, введених користувачем. Для перетворення цього рядка в число використовується зовнішня функція **Val**, яка виконується після завершення роботи **InputBox**.

```
Наприклад, потрібно ввести значення змінної a=3,5.
 Dim a As single
 a = Val(InputBox ("Введіть значення a"))
```



При виконанні фрагмента програми на екрані з'явиться діалогове вікно, представлене на рисунку. У текстове поле вікна потрібно ввести 3.5 і натиснути кнопку ОК.

Функція MsgBox використовується для виведення даних або повідомлень на екран. Синтаксис функції:

MsgBox (запрошення[, кнопки][, заголовок][, файл допомоги][, контекст]),

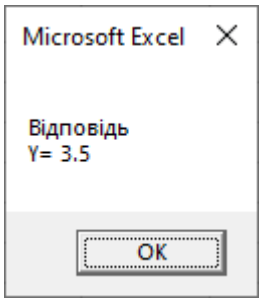
де **кнопки** - константа, що визначає командні кнопки, які з'являються в діалоговому вікні. За замовчуванням установлюється значення 0. Інші параметри мають той же зміст, що і у функції **InputBox**.

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Деякі значення констант кнопок функції MsgBox. 0 – Виводиться тільки кнопка ОК, 1 – Виводяться кнопки ОК і Отмена, 3 – Виводяться кнопки Да, Нет Отмена, 4 – Виводяться кнопки Да, Нет.

Як і у всіх функціях VBA, вказівка аргументів функції **MsgBox** у круглих дужках необхідна тільки тоді, коли планується використання значень, що повертаються, цієї функції. Якщо виводиться велике повідомлення, то **VBA** робить перенос тексту в рамках діалогового вікна. Для вказівки власних меж переносу в кожному рядку використовується функція **Chr** і код повернення каретки (ASCII 13).

Зазвичай у якості аргументу функції використовується повідомлення, що представляє собою конкатенацію (зчеплення) двох рядків: коментаря до результату, узятого в лапки, і рядка, отриманої із числового вираження за допомогою функції **Str**.




```
Вивести результат
на екран монітора.
Msgbox("Відповідь=" & Chr(13) & "Y=" & Str(y))
```

Використання вікна налагодження. Вікно налагодження можна вивести на екран за допомогою команди **Вид(View) - (Вікно налагодження (Immediate Window)** або натиснувши клавіші **Ctr+G** .

Для виводу поточних значень у це вікно безпосередньо з макросу необхідно ввести в потрібному місці коду команду **Debug** з єдиним методом **Print**. Синтаксис

Debug.Print a1 & a2 & a3 …

де a1, a2, a3 - константи, змінні, вирази. У місці, де встановлена ця команда, відбудеться переривання виконання макросу й у вікні налагодження з'явиться поточне значення змінної або виразу.

4. Реалізація основних алгоритмічних конструкцій на мові VBA

Виділяють наступні алгоритмічні структури:

- **лінійна**, коли всі дії виконуються послідовно, без умов, від початку (**Sub**) до кінця (**End Sub**);
- **розгалужена**, коли на деякому кроці зустрічається умова, залежно від якого процес розв'язку може піти по одному з декількох напрямків;
- **циклічна**, коли деякі фрагменти програми повинні виконуватися кілька раз.

Програмування алгоритмів лінійної структури

Програма лінійного обчислювального процесу являє собою послідовність рядків операторів, що містять оператори присвоювання, а також засобу введення й виводу.

Звичайно будь-який текст програми мовою VBA починається з опцій, які управляють описом змінних, способом завдання індексів масивів тощо. Потім слідують оголошення глобальних змінних або констант для даного модуля, тобто таких змінних, які можуть використовуватися у всіх процедурах модуля. Далі необхідно організувати введення початкових даних, потім безпосередньо розташовується текст функцій і

Перелік лекцій

процедур, що становлять саму послідовність обчислень, у кінці необхідно передбачити вивід отриманих результатів.

```
Початок модуля
   Опції VBA (Option Base 1, Option Explicit)
   Опис глобальних констант
   Опис глобальних змінних
   Sub ім'я_процедури ([аргументи])
     Опис констант
     Опис змінних
     Введення вхідних значень
     Інструкції процедури
     Виведення отриманих результатів
   End Sub
Кінець модуля
```

Бажано додавати у програму коментарі для зручності її читання.

```
Приклад.
Скласти програму для обчислення площі S трикутника зі
сторонами a, b, c по формулі Герона

Sub Geron()
  ' опис змінних
Dim a As Single, b As Single, c As Single
Dim p As Single, s As Single
a = Val(Inputbox("Уведіть a")) 'вводимо значення сторони a
b = Val(Inputbox("Уведіть b")) 'вводимо значення сторони b
c = Val(Inputbox("Уведіть c")) 'вводимо значення сторони c
' обчислюємо півпериметр
p = (a + b + c)/2
' обчислюємо площу трикутника
s = Sqr(p * (p - a) * (p - b) * (p - c))
' виводимо результат
MsgBox ("Площа=" & Str(s))
End Sub
```

Програмування розгалужуваний обчислювальних процесів

Звичайне виконання макросу починається з виконання інструкцій **SUB** і триває в порядку проходження рядків з операторами доти, поки не зустрінеться інструкція **END SUB**.

Але часто, при проектуванні макросів, необхідно виконувати різні варіанти програми, залежно від значення яких-небудь змінних або параметрів (умов), що задаються користувачем під час виконання програми. Кожний окремий напрямок обробки інформації в такому процесі називається галузями. Вибір галузей визначається перевіркою виконання логічної умови, що визначає властивості вихідних даних або проміжних результатів. У кожному конкретному випадку процес обробки даних виконується лише по одній з галузей.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

На цей випадок в VBA передбачений ряд операторів, призначених для зміни порядку виконання операторів у програмі.

Оператор безумовного переходу **GoTo**. Дозволяє змінити порядок виконання команд. Задає безумовний перехід на зазначений рядок усередині тієї процедури, у якій вона перебуває. Синтаксис інструкції

GoTo Мітка

Мітка (ціле число або ім'я змінної) з двома крапками, яка ставиться перед рядком, до якого здійснюється перехід. Кількість міток не обмежена, але кожна повинна бути унікальної.

```
Приклад: тут міткою служить NN:
         MsgBox ("Даний оператор виконується")
         GoTo NN
         Msgbox ("Даний оператор пропускається")
         NN:
         Msgbox ("Тепер можна продовжувати")
```

Занадто велике число інструкцій **Goto** утрудняє читання й налагодження програм. По можливості слід використовувати структурні керуючі інструкції

Інструкція **GoTo** в VBA застосовується, головним чином, для обробки помилок.

Оператор **If... Then... Else**

Умовний оператор **If... Then...Else** виконує певні інструкції або набори інструкцій залежно від значення заданих умов. Синтаксис наступний:

If умова Then [оператор1] [Else оператор2]

Розрізняють умовні оператори **рядковий** і **блоковий**. **Рядковий** оператор використовується в тому випадку, коли при розгалуженні програми необхідно на кожній вітці виконати тільки по одному операторові, а блоковий необхідний тоді, коли операторів декілька.

Обов'язковий параметр **умова** — являє собою логічних вираз, яке повертає значення **TRUE** (істина) або **FALSE** (неправда). Якщо умова рівна **TRUE**, то виконується **оператор1**, якщо умова рівна **FALSE**, те виконується **оператор2**. Далі виконується, якщо інше не передбачене програмою, інструкція, що стоїть одразу за рядком з інструкцією **If... Then...Else**.

Частина оператора **If**, а саме ключове слово **Else** і наступні за ним інструкції можна вилучити, тоді за умови рівному **FALSE** виконується оператор, що стоїть одразу за рядком з інструкцією **If**. Таким чином інструкція виглядає так:

If умова Then [оператор]

У реальних випадках часто потрібна обробка декількох інструкцій при виконанні умови. У цих випадках використовується блоковий вид інструкції **If**, синтаксис якої виглядає наступним чином

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

If умова Then

[Блок операторів]

[Else

[Блок операторів]]

End If

Блокова структура **If** зручніше, тому що кожний блок операторів може містити в собі довільне число команд. Крім того, допускається вкладення структур **If... Then...Else** одна в одну (до 7 вкладень).

Оператор **Select Case**

Це оператор вибору з багатьох можливих варіантів. Оператор **Select Case** виконує одну з декількох груп операторів залежно від значення вираження. У мовах високого рівня такий оператор називається перемикачем. Синтаксис інструкції

Select Case вираз

[Case список умов-1

[оператори-1]]

[Case список умов-2

[оператори-2]]

…

[Case список умов-n

[оператори-n]]

[Case Else

[оператори_else]]

End Select

Вираз — обов'язковий параметр, може бути будь-яким виразом будь-якого типу даних VBA, включаючи числовий, строковий, або логічний.

Список умов — обов'язковий при наявності інструкції **CASE**. Містить результати обчислення виразу, що записані в одному з видів:

Case КОНСТАНТА1, КОНСТАНТА2, КОНСТАНТА3, ...

Case Is ЗНАК_ ВІДНОСИНИ КОНСТАНТА

Case КОНСТАНТА1 to КОНСТАНТА2

Виконується оператор у такий спосіб — спочатку обчислюється значення виразу, що розташовується після ключових слів **Select Case**, потім проводиться перевірка на збіг значення виразу й одного з **Case** умов, у випадку збігу виконуються оператори, поміщені за цією **Case** умовою. У випадку, якщо збігу не виявлено у жодній

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Case умов, то виконуються оператори, що розташовані після ключових слів **Case Else**.

Якщо список виразів у **Case** умові записаний у першій формі, то необхідно, щоб результат обчислень виразу дорівнював однієї константі зі списку, якщо **Case** умова записана в другій формі, то необхідно, щоб виконувалася умова відносини між значенням вираження й константою. Якщо ж у третій формі, то необхідне виконання співвідношення **КОНСТАНТА1 ≤ вираз ≤ КОНСТАНТА2**.

```
Приклад.
Дано дійсне число x. Обчислити y = sin(x), якщо x > 0,
інакше y = x^2.

Sub Prim2()
Dim x As Double, y As Double
x = Val(Inputbox("Уведіть x")) 'уведення значення змінної x
'перевірка умови, оператор рядковий
If x > 0 Then y = Sin(x) Else y = x^2
Msgbox ("Значення y=" + Str(y))
End Sub
```

```
Приклад.
Дано три числа. З'ясувати, чи існує трикутник з такими
сторонами.

Sub Treug()
Dim a As Double, b As Double, c As Double
a= Val(Inputbox("Уведіть сторону a"))
b= Val(Inputbox("Уведіть сторону b"))
c= Val(Inputbox("Уведіть сторону c"))
'трикутник існує, якщо сума двох сторін більша за третю
'перевірка умови, оператор блоковий
If (a + b) > c And (b + c) > a And (a + c) > b Then
   Msgbox ("Трикутник існує")
 Else
   Msgbox ("Трикутник не існує")
End If
End Sub
```

Програмування алгоритмів циклічної структури

Розв'язок багатьох практичних завдань зводиться до виконання обчислень по тим самим формулам, але при різних значеннях вхідних величин до них. Такий обчислювальний процес називається циклічним, а багаторазово повторювані ділянки цього процесу називаються циклами.

Розрізняють **регулярні** цикли з керуючим параметром (з відомим числом повторень), умовою закінчення якого є досягнення параметром циклу свого кінцевого

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

значення та цикли **ітераційні**, у яких умова повторення або закінчення циклу задається по деякому результату, наприклад, поки не буде досягнута точність обчислень. Реалізуються цикли за допомогою спеціальних операторів циклу.

Оператор циклу **While... Wend.** Оператор має синтаксис

While умова

[оператори]

Wend

умова — обов'язковий елемент. Числовий або строковий вираз, який має значення **True** або **False**. **Оператори** — необов'язковий елемент. Один або декілька операторів, що виконують, дотиди поки умова має значення **True**.

Цей оператор називають оператором циклу із **передумовами**. Виконується оператор **While... Wend** у такий спосіб. Якщо умова має значення **True**, виконуються всі оператори до інструкції **Wend**. Потім керування вертається до інструкції **While** і знову перевіряється умова. Якщо умова як і раніше має значення **True**, процес повторюється. Якщо вона не має значення **True**, виконання відновляється з інструкції, що є наступною за інструкцією **Wend**. У зв'язку із цим елемент **умова** тут є умовою виконання циклу. Цикли **While... Wend** можуть мати будь-яку глибину вкладеності.

Оператор циклу **Do...Loop**. Оператор **Do...Loop** також використовується для виконання наборів операторів невизначене число раз. Оператор має синтаксис:

Do

[оператори]

Loop Until [умова]

умова — необов'язковий елемент. Числовий або строковий вираз, який має значення **True** або **False**. **Оператори** — один або кілька операторів, виконання яких повторюється, доки умова не прийме значення **True**.

Цей оператор називають оператором циклу з **післяумовою**. Виконується оператор **Do...Loop** у такий спосіб. Якщо умова має значення **False**, виконуються всі оператори після інструкції **Do**. Потім керування передається інструкції **Until** і знову перевіряється умова. Якщо умова як і раніше має значення **False**, процес повторюється. Якщо воно має значення **True**, керування передається наступному за інструкцією **Loop Until** операторові. У зв'язку із цим елемент умова тут є умовою виходу із циклу.

Оператор циклу **For...Next**. Часто при складанні макросу заздалегідь відома кількість повторень групи операторів, у таких випадках можна використовувати інструкцію **For...Next**.

Оператор **For...Next** використовується для виконання наборів операторів зазначене число раз. Цикли **For** використовують у якості лічильника змінну, значення

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

якої збільшується або зменшується при кожнім виконанні циклу на зазначене значення. Синтаксис інструкції **For...Next**

For лічильник = початок To кінець [step крок]

[оператори]

Next [лічильник]

лічильник — обов'язковий елемент. Це повинна бути числова змінна. Вона не може мати тип **Boolean** або бути елементом масиву. **початок** — обов'язковий елемент, містить початкове значення змінної лічильник. **кінець** — обов'язковий елемент, містить кінцеве значення змінної лічильник. **крок** — елемент необов'язковий, це значення, на яке змінюється лічильник при кожнім виконанні тіла циклу. Якщо це значення не задане, за замовчуванням крок дорівнює одиниці. Крок може бути як позитивним, так і негативним. **Оператори** — необов'язковий елемент. Один або декілька операторів між **For** і **Next**, які виконуються зазначене число раз.

Інструкція **For...Nex**t працює в такий спосіб: початкове значення елемента лічильник порівнюється з кінцевим значенням. Якщо крок позитивний і початкове значення менше кінцевого або якщо крок негативний і початкове значення більше кінцевого, то керування передається усередину тіла циклу. Після виконання всіх операторів у тілі циклу значення крок додається до поточного значення змінної лічильник. Після цього оператори тіла циклу або виконуються ще раз (на основі тієї ж умови, яка привела до початкового виконання циклу), або цикл завершується й виконання продовжується з оператору, що знаходиться за **Next**.

Допускається вкладення циклів **For...Next** (один цикл **For...Next** розташовується усередині іншого). Лічильник кожного циклу повинен мати унікальне ім'я.

```
Приклад.
Знайти суму квадратів усіх натуральних чисел 1,2,3, … n
Число n задавати довільно
Sub sum()
Dim n, i As Integer, s As Double
n = Val(Inputbox("Уведіть кількість чисел до підрахунку"))
' перед накопиченням суми треба обнулити змінну
s = 0
' регулярний цикл від 1 до n
For i = 1 To n
' накопичення суми
s = s + i^2
Next i
' вивід результату
Msgbox ("Сума s=" + Str(s))
End Sub
```

Усередині одного циклу можуть знаходитися один або декілька інших циклів. У цьому випадку цикл, що охоплює, називається зовнішнім, а вкладені в нього цикли –

Перелік лекцій

внутрішніми. Правила організації як зовнішнього, так і внутрішніх циклів аналогічні правилам організації простого циклу. Параметри зовнішнього й внутрішнього циклів змінюються не одночасно, тобто при одному значенні параметра зовнішнього циклу параметр внутрішнього послідовно приймає всі можливі значення. При організації вкладених циклів необхідно стежити за тим, щоб область дії внутрішнього циклу не виходила за область дії зовнішнього циклу.

Оператор **For Each...Next**. Цей оператор дозволяє виконувати обробку набору об'єктів (наприклад, елементів масиву даних). У цьому випадку немає необхідності використовувати лічильник циклу, тому що циклічний перегляд і обробка виробляється для певних елементів набору. Синтаксис оператора:

For Each елемент In група

Блок операторів

Next [елемент]

Параметр **елемент** - змінна типу **Variant** (ця обов'язкова умова), що представляє елемент сімейства або масиву. **Група** - ім'я масиву або сімейства. Блок операторів виконується тільки один раз для кожного члена групи.

Найважливіше в операторі **For Each...Next** те, що не потрібно заздалегідь знати, скільки елементів утримується в масиві або сімействі. Таким чином, не потрібно підраховувати число елементів або перевіряти досягнення кінця групи. Це дозволяє виключити одну з найбільш загальних помилок при програмуванні: неправильну індексацію елементів масивів.

5. *Процедури*

Програма може являти собою сукупність окремих блоків-підпрограм (процедур). Використання процедур дозволяє реалізувати один із самих прогресивних методів програмування - структурне програмування.

Процедури доцільно застосовувати у випадках, коли однакову послідовність дій необхідно виконувати в різних частинах програми й при різних вхідних даних. Таку послідовність дій можна виділити в самостійну програмну одиницю - **процедуру**, що може бути використана багаторазово в міру необхідності.

В VBA є 3 види процедур: процедура **Function**, процедура **Sub** і процедура **Property**.

Процедура **Function.** VBA містить безліч вбудованих функцій, наприклад, **cos**, **sqr**, **mid**, але користувач може створювати власні функції, використовуючи для цього процедуру **Function**. Синтаксис:

[Private | Public | Static] Function Ім'я(Список аргументів) [As тип]

Блок операторів

End Function

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Тут і далі │ означає, що використовується один з варіантів: **Private**, **Public** або **Static**:

- **Private** - означає, що процедура **Function** доступна для інших процедур тільки того модуля, у якому вона описана.
- **Public** - означає, що процедура **Function** доступна для інших процедур у всіх модулях.
- **Static** – означає, що локальні змінні процедури **Function** зберігаються в проміжках часу між викликами цієї процедури.

Ім'я – ідентифікатор функції, що задовольняє стандартним правилам побудови імен; **Тип** – тип значення, що повертається **Function**; **Список аргументів** - список змінних, що представляють аргументи, які передаються в процедуру **Function** при її виклику. Аргументи в списку відділяються комами.

Структура аргументу:

[Optional] [ByVal │ByRef] [ParamArray] Ім'я [As тип]

Optional - указує, що даний аргумент є необов'язковим. Аргументи, позначені кваліфікатором **Optional**, повинні перебувати наприкінці списку аргументів.

ByVal - означає, що аргумент передається за значенням. При цьому виключається можливість зміни значення аргументу процедури після її завершення.

ByRef – аргумент передається по посиланню. Передача аргументу по посиланню дає процедурі доступ до вмісту змінної за її адресою в пам'яті. Значення змінної може бути змінене процедурою, що викликається.

За замовчуванням установлюється **ByRef**.

ParamArray - позначає необов'язковий аргумент, що є масивом довільного числа змінних типу **Variant**. Цей аргумент може бути тільки останнім у списку аргументів. З ним не можуть використовуватися ключові слова **ByVal**, **ByRef**, **Optional**.

Тип - тип аргументу, може використовуватися будь-який стандартний тип даних. За замовчуванням використовується **Variant**.

```
Приклади заголовка процедури-функції:
1) Function Sum(n As Integer, a As Single) As Integer

Ця функція – Public (за замовчуванням), доступна для всіх
модулів. Ім'я функції – Sum, аргументи: n – цілого типу, a –
числового, обидва аргументи мають кваліфікатор  ByRef (за
замовчуванням). Функція повертає значення типу Integer.

2)Private Function F1(a As Single, ParamArray x) As Single

Функція доступна тільки для модуля, у якому вона записана.
Аргументи: a - числова змінна типу Single, x - масив
довільного числа елементів типу Variant.Функція повертає
значення типу Single.
```

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Процедура **Function** може перебувати в будь-якому місці модуля, за винятком тіла іншої процедури. Викликається процедура по імені із вказівкою фактичних аргументів, у ролі яких можуть виступати константи, змінні, вирази. Виклик може перебувати усередині будь-якого виразу.

Виконання **Function** починається з першого рядка тіла процедури й триває доти, поки не буде досягнута інструкція **End Function** або **Exit Function** (вихід з **Function**). Далі управління передається в основну процедуру.

Процедура **Function** повертає в основну процедуру значення, присвоює ім'я **Function** оператором: **ім'я = вираз**. Цей оператор може бути записаний у тілі **Function** кілька разів, при цьому **Function** повертає останнє присвоєне значення. Якщо тип функції не заданий явно, то функція повертає значення типу **Variant**.

Формально **Function** повертає тільки одне значення, однак після виконання **Function** можна використовувати й модифікації аргументів, переданих в **Function** по посиланню (**ByRef**).

Процедура **Sub**. Синтаксис процедури **Sub**:

[Private | Public | Static] Sub Ім'я([Список аргументів])

Блок операторів

End Sub

Загаловок **Sub** аналогічний заголовку **Function**, тільки в процедурі **Sub** не вказується тип процедури.

Процедура **Sub** не повертає значення, що присвоюється її імені, як це робить процедура **Function**, однак як і **Function**, може змінювати значення аргументів, переданих до неї. Тому, якщо процедура **Sub** призначена для розрахунку якихось величин, то в списку аргументів процедури **Sub** повинні бути присутнім як вхідні аргументи, так і вихідні.

Процедура **Sub**, на відміну від **Function**, не може бути викликана по імені функції усередині виразу. Для її виклику використовується окремий оператор. Застосовуються 2 модифікації виклику:

Call Ім'я (Список аргументів)

Ім'я Список аргументів

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Питання для самоконтролю

1. Що таке ООП (об'єктно-орієнтоване програмування)?
2. Як розшифровується абревіатура VBA?
3. Що таке в термінах ООП об'єкт, властивість, метод, подія?
4. За що відповідає властивість (Name), Caption, Font, Visible, Picture?
5. Які події Ви знаєте?
6. За допомогою якої команди відкривається середовище редактора VBA?
7. Які типи даних використовуються в VBA?
8. Як описуються константи й змінні в тілі програми?
9. Приведіть синтаксис функції Inputbox і процедури Msgbox.
10. Які функції перетворюють строкову змінну в числову та навпаки?
11. Які математичні функції VBA ви знаєте?
12. У яких випадках використовуються математичні функції VBA?
13. Що таке алгоритми, що розгалужуються?
14. Яким елементом на блок-схемі позначається умовний оператор?
15. Який синтаксис має умовний оператор If... Then...Else і Select Case?
16. У чому відмінність рядкового й блокового операторів If... Then...Else? У яких випадках потрібно використовувати блоковий оператор?
17. У якому виді можна записати список умов в операторі Case?
18. Що таке циклічні алгоритми?
19. Яким елементом на блок-схемі позначається оператор циклу?
20. Що таке регулярні й ітераційні цикли? У чому їх відмінність?
21. Приведіть синтаксис оператора циклу Do…Loop і While…Wend.
22. У чому відмінність циклу з післяумовою від циклу із передумовою? У яких випадках який цикл потрібно використовувати?
23. Приведіть синтаксис оператора циклу For…Next. При розв'язку яких завдань використовується даний оператор?
24. Яких правил слід дотримуватися при побудові вкладених циклів?
25. Що таке вкладений цикл і як він реалізується?
26. Які правил слід дотримуватися при побудові вкладених циклів?
27. Що таке масив і для чого він створюється? Як організувати доступ до даних у масиві?
28. Що таке нумерація з нульовою базою і як її можна змінити?
29. Для чого призначений тип String
30. Що таке конкатенація?

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua

Перелік використаних джерел

1. Нелюбов В. О., Куруца О. С. Основи інформатики. Microsoft Excel 2016: навчальний посібник. Ужгород: ДВНЗ «УжНУ», 2018. - 58 с.: іл.
2. Конспект лекцій з дисципліни «Інформатика» для студентів першого курсу заочної форми навчання спеціальностей 051 «Економіка», 075 «Маркетинг» // Укл. Нємцова А.А., Харків: НФУ, 2016. - 48 с.
3. Комп'ютерні мережі та телекомунікації : навч. посібник / В. А. Ткаченко, О. В. Касілов, В. А. Рябик. – Харків: НТУ "ХПІ", 2011. – 224 с.
4. Денисова А., Вихарев И., Белов А., Наумов Г. Интернет. Самоучитель. 2-е изд. – СПб. Питер. 2004.– 368 с.
5. Навчальний курс НВУ ІНТУІТ :Введення до HTML. [електронний ресурс]. Режим доступу: https://www.intuit.ru/studies/courses/33/33/info
6. Навчальний курс НВУ ІНТУІТ : Застосування каскадних таблиць стилів (CSS). [електронний ресурс]. Режим доступу: https://www.intuit.ru/studies/courses/34/34/info
7. Інформатика та інформаційні технології у цивільній безпеці: Практикум / [Маляров М.В, Гусева Л.В., Паніна О.О. та ін.]; Під заг. ред. М.В. Малярова. - Харків: НУЦЗ України, 2015. - 330 с.
8. Хорев В.Д. Самоучитель программирования на VBA в Microsoft Office. — К.: Юниор, 2001. —320 с., ил.
9. Конспект лекцій з курсу «Інформатика» для бакалаврів 6.051701 денної та заочної форм навчання // Укл. О.Д. Гонтар, Н.О. Макоєд, Одеса: ОНАХТ, 2009, 49 с.

Перелік лекцій

©НУЦЗ України, 2019
Http://www.nuczu.edu.ua