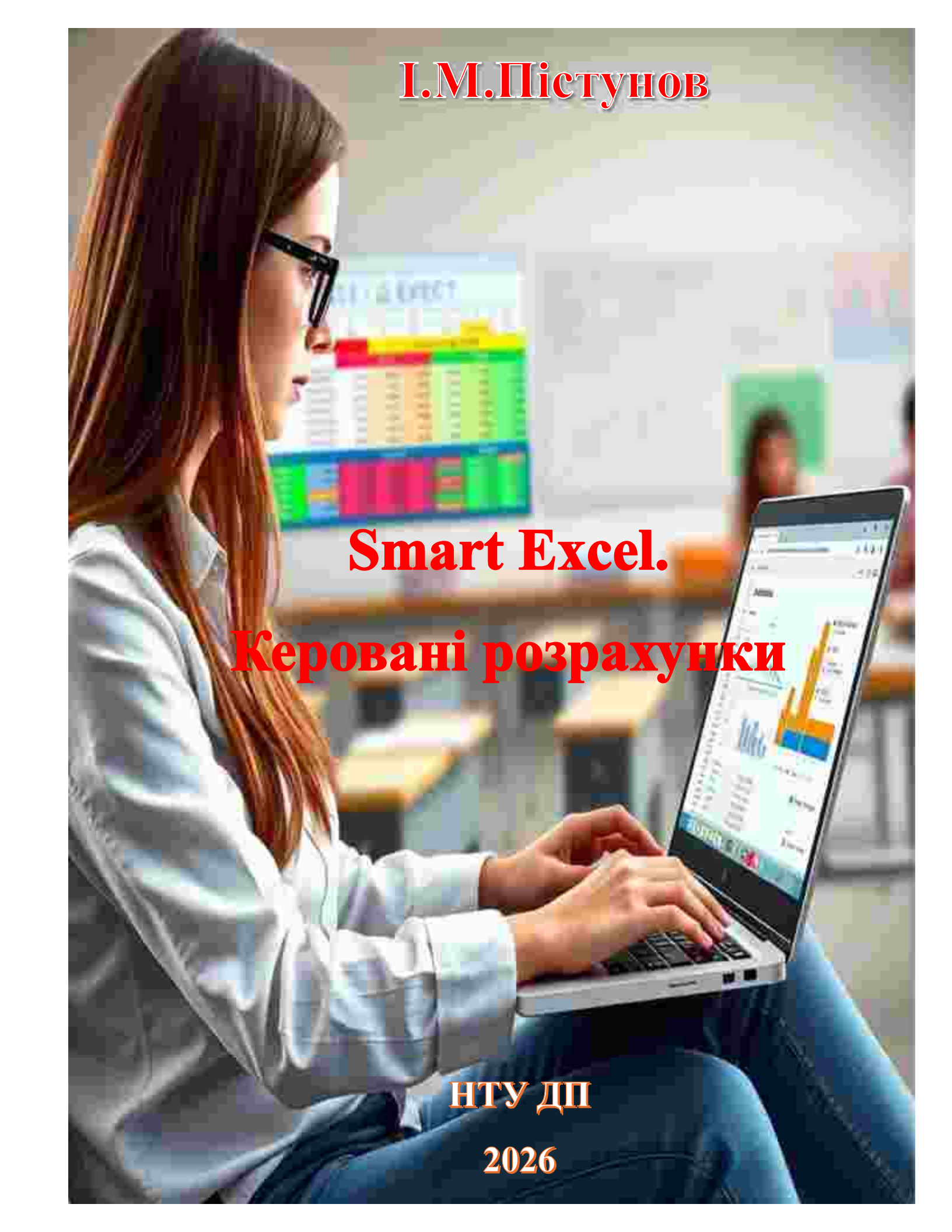




І.М.Пістунов

**Smart Excel.
Керовані розрахунки**

НТУ ДП

2026

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ДНІПРОВСЬКА ПОЛІТЕХНІКА»



І.М. Пістунов

Smart Excel. Керовані розрахунки

Навчальний посібник

Дніпро
НТУ ДП
2026

*Рекомендовано вченою радою НТУ «Дніпровська політехніка»
як навчальний посібник для здобувачів ступеня бакалавра всіх спеціальностей
(протокол № від . .2026)*

Рецензенти:

А.В. Бардась, докт. екон. наук, проф., декан факультету менеджменту
Національного гірничого університету;

Б.І.Мороз, док. техн. наук, проф., декан технічного факультету Університету
митної справи та фінансів.

Пістунов І.М.

ПЗ4 Smart Excel. Керовані розрахунки. [Електронний ресурс] : навч. посіб. /
І.М. Пістунов ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська
політехніка». – Дніпро : НТУ «ДП», 2026. – 265 с.

В посібника навчає студентів використовувати всі можливості програми Excel, що входить до програмного пакету Microsoft Office.

Студенти навчаються створювати інтерактивні розрахунки, які заміняють комп'ютерні програми, використовуючи можливості інтеграції кнопочного управління, випадаючих вікон, списків, пересилань та логічних операторів.

В посібнику подано завдання для самостійного виконання, тому він може слугувати і як посібник для практичних чи лабораторних занять із застосуванням комп'ютерної техніки.

Призначений для студентів вищих навчальних закладів і може бути корисним для фінансистів, економістів, плановиків, менеджерів та маркетингологів.

Посібник базується на літературних джерелах вітчизняних, зарубіжних авторів, ресурсах Інтернету та на досвіді викладання дисципліни «Smart Excel. Керовані розрахунки» в НТУ ДП.

© І.М. Пістунов, 2026

© НТУ «Дніпровська політехніка», 2026

ЗМІСТ

ВСТУП.....	7
Мета та можливості інтерактивних розрахунків в Excel.....	8
Порядок засвоєння матеріалу.....	9
Набуті знання	12
 Розділ 1. НАЛАШТУВАННЯ EXCEL	16
1.1. Налаштування доступу до підпрограм в Excel.....	16
1.2. Підключення меню розробника	19
1.3. Налаштування інструментів Power Query та Power Pivot	22
1.4. Індивідуальне завдання №1. Налаштування Excel.....	24
 Розділ 2. ОСНОВИ АВТОМАТИЗАЦІЇ В EXCEL	26
2.1. Відносні, абсолютні та змішані посилання	27
2.2. Майстер функцій	29
2.3 Використання логічних функцій для динамічних розрахунків	31
2.3.1. Використання логічних функцій електронних таблиць Excel для опису висловлювань, представлених у вигляді формул алгебри логіки.....	40
2.4 Функції для роботи з текстовими даними.....	43
2.5. Робота з датами та часом	45
2.6. Масиви та нові функції	47
2.7. Нові функції	48
2.7.1. Функція LET	48
2.7.2. Функція LAMBDA	49
2.7.3. Функція SEQUENCE.....	51
2.7.4. Функція UNIQUE(масив)	54
2.7.5. Функція SORT(масив, стовпець, порядок)	57

2.7.6. Функція FILTER(масив, умова)	59
2.7.7. Функція XLOOKUP	63
2.7.8. SUMIF – додавання за умови	68
2.7.9. Функція INDIRECT – пошук за посиланням	72
2.7.10. Функція SUBTOTAL	74
2.8. Автоматизація розрахунків із Power Query та Power Pivot	77
2.9. Прогноз.....	85
2.9.1. Диспетчер сценаріїв.....	85
2.9.2. Підбір параметра.....	88
2.9.3. Таблиця даних.....	91
2.9.4. Аркуш прогнозу.....	94
2.10. Індивідуальне завдання № 2. Робота з новими можливостями Excel.....	98
 Розділ 3. ІНТЕРАКТИВНІ ЕЛЕМЕНТИ EXCEL	101
3.1. Використання списків, що фільтруються, та таких, що випадають	101
3.2. Використання кнопок, прапорців та перемикачів.....	104
3.2.1. Кнопка	104
3.2.2. Поле зі списком	106
3.2.3. Прапорець	107
3.2.4. Лічильник.....	108
3.2.5. Список	109
3.2.6. Кнопка-перемикач.....	111
3.2.7. Вікно групи	112
3.2.8. Надпис	114
3.2.9. Смуга прокручування	115
3.3. Графічні елементи	116
3.3.1. Картинки та фігури як елемент інтерактивного вікна.....	120
3.3.2. SmartArt.....	122

3.3.3. Рухомі графіки	124
3.4. Індивідуальне завдання №3. Вивчення меню Розробник	131
Розділ 4. АВТОМАТИЗАЦІЯ З VBA	134
4.1. Введення у VBA: основні принципи	136
4.2. Запис та редагування макросів.....	142
4.3. Використання кнопок для запуску макросів	145
4.4. Робота з діалоговими вікнами (MsgBox, InputBox, UserForm).....	148
4.5. Індивідуальне завдання №3. Вивчення елементів VBA.....	154
Розділ 5. ДИНАМІЧНА ВІЗУАЛІЗАЦІЯ ДАНИХ.....	157
5.1. Використання умовного форматування	158
5.2. Динамічні діаграми (включення/виключення серій даних).....	165
5.3. Графіки, що змінюються за допомогою кнопок.....	171
5.3.1. Вимикання сітки на діаграмах та на аркушах Excel.....	171
5.3.2. Інші методи для динамічних графіків	173
5.4. Індивідуальне завдання №4. Візуальні ефекти на графіках.....	181
Розділ 6. ПІДКЛЮЧЕННЯ ДО ЗОВНІШНІХ ДАНИХ.....	184
6.1. Використання Power Query для інтеграції даних.....	184
6.2. Мова M	191
6.3. Підключення до баз даних.....	194
6.4. Робота з веб-запитами.....	199
6.5. Індивідуальне завдання №5. Отримання даних із зовнішніх носіїв.....	205
Розділ 7. ПРАКТИЧНІ ПРИКЛАДИ ТА КЕЙСИ	208
7.1. Розрахунок кредитного платежу з регулюванням параметрів	208
7.2. Автоматизація обліку витрат	213
7.3. Динамічне планування ресурсів	221

7.4. Інтерактивний дашборд для аналізу продажів	228
ВИСНОВКИ	236
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	239
ДОДАТКИ.....	240
Додаток А. Гарячі клавіші в Excel	241
Додаток Б. Поширені помилки та їх усунення	243
Додаток В. Готові шаблони для завантаження	246
1. Перехід на інший лист книги Excel	247
2. Перехід на конкретну комірку у цьому ж листі	247
3. Звертання до Solver через VBA	247
4. Звертання до підпрограми Regression.....	251
5. Звертання до функції LINEST	252
6. Звертання до підпрограми Descriptive Statistics	254
7. Звертання до підпрограми Correlation	256
8. Звертання до будь-якої функції.....	258
9. Зміна налаштувань вікна Excel.....	260
10. Вимкнення вікна Excel	261
11. Використання UserForm для вставки тексту, картинки та кольорів.....	262
12. Типові іконки Excel.....	263

ВСТУП

Коли Microsoft Excel з'явився у 1985 році, він сприймався як зручний інструмент для роботи з таблицями та базовими обчисленнями. Однак із розвитком технологій ця програма стала справжньою «відмичкою» для аналізу даних, бізнес-процесів і фінансового моделювання. Сьогодні Excel – це не просто таблиці та формули, а потужне середовище для створення інтерактивних розрахунків, що дозволяють автоматизувати складні обчислення, візуалізувати результати та навіть реалізовувати логіку прийняття рішень.

Однією з головних переваг Excel завжди була його гнучкість. Програма дозволяє адаптуватися до найрізноманітніших завдань: від ведення сімейного бюджету до складного фінансового аналізу на рівні міжнародних корпорацій. Але зі зростанням обсягу даних та їхньої складності виникла потреба у нових підходах до роботи з інформацією.

На зміну статичним таблицям прийшли динамічні моделі, що дозволяють змінювати параметри у реальному часі. Це стало можливим завдяки інтеграції елементів управління (кнопки, повзунки, випадаючі списки), умовного форматування, макросів VBA та Power Query. Тепер користувачі можуть не просто вводити числа в клітинки, а створювати справжні інтерактивні інтерфейси, що реагують на зміни даних та автоматично виконують потрібні розрахунки.

Ця еволюція Excel не тільки розширила його можливості, а й зробила роботу з програмою більш інтуїтивною. Наприклад, замість того щоб кожного разу вводити нові коефіцієнти у формули, користувач може просто натискати кнопку або рухати повзунок – і Excel миттєво перерахує всі значення. Такий підхід значно зменшує ризик помилок, економить час і робить аналіз більш наочним.

З розвитком бізнесу, науки та освіти потреба в автоматизованих та інтерактивних рішеннях лише зростає. Саме тому сучасний Excel – це не просто

програма для роботи з числами, а повноцінне середовище для розробки аналітичних рішень та оптимізації бізнес-процесів.

Мета та можливості інтерактивних розрахунків в Excel

У цьому посібнику ми розглянемо основні принципи створення інтерактивних розрахунків у Excel та покажемо, як перетворити звичайну таблицю на динамічний інструмент для швидкого аналізу даних. Готові вдосконалити свої навички та відкрити нові можливості Excel? Тоді рухаємося далі!

Excel давно став невід'ємною частиною роботи з числовими даними, звітністю та аналітикою. Проте сучасні вимоги до швидкості та зручності обчислень виходять за межі простого введення формул і таблиць. Саме тут на допомогу приходять інтерактивні розрахунки – підхід, що дозволяє користувачам не тільки виконувати обчислення, а й керувати ними за допомогою кнопок, списків, повзунків та інших елементів управління.

Інтерактивність у розрахунках – це можливість динамічно змінювати параметри моделі без редагування формул вручну. Наприклад, фінансовий аналітик може легко переглядати вплив зміни відсоткової ставки на кредитний платіж, просто пересуваючи повзунок. Менеджер з продажу може швидко змінювати місяці або фільтрувати дані за регіонами, використовуючи випадуючі списки. Це не лише спрощує аналіз даних, але й робить його більш наочним і доступним.

Ключовими інструментами для створення інтерактивних розрахунків у Excel є умовне форматування, випадуючі списки, макроси VBA, кнопки та графічні елементи. Разом вони дозволяють створювати динамічні моделі, що змінюються в реальному часі, адаптуючись до введених параметрів.

Переваги інтерактивних розрахунків очевидні:

✓ Швидкість та зручність – зміна параметрів займає секунди, не вимагаючи внесення правок у формули;

✓ Візуалізація – користувачі можуть бачити результат одразу, що значно полегшує аналіз та прийняття рішень;

✓ Зменшення помилок – автоматизація зменшує ймовірність ручних помилок у розрахунках;

✓ Гнучкість – створені моделі легко адаптуються під різні сценарії та потреби.

Сьогодні інтерактивні розрахунки знаходять застосування у фінансовому аналізі, бюджетуванні, бізнес-аналізі, наукових дослідженнях та навіть навчанні. Використовуючи ці можливості, можна значно підвищити ефективність роботи з Excel, перетворивши його з інструмента введення чисел на потужний засіб моделювання та прогнозування.

Цей посібник допоможе вам освоїти всі необхідні навички для створення інтерактивних розрахунків, незалежно від рівня вашого досвіду. Готові розширити свої можливості в Excel? Тоді розпочнемо!

Порядок засвоєння матеріалу

Після кожного розділу подані індивідуальні завдання, які студенти мають виконати протягом часу на засвоєння предмету «Електронний менеджмент».

Індивідуальні завдання оформляються як документ, який подається в електронному вигляді, вміщеним на будь-який носій інформації. Формат електронного документу має відповідати електронному процесору Excel Microsoft Office. Окрім того, активні вікна мають бути скопійовані в документ Word з коментарями і описом роботи на кожному етапі.

Задачу спочатку треба розв'язати в загальному вигляді з представлення формули рішення, в яку потім підставлені конкретні числові дані для свого варіанта. В деяких завданнях числові значення потрібно визначити за простою

формулою. Наприклад, якщо в таблиці навпроти позначення C стоїть число 17, а числове значення в умові задачі подано як $0,01 \cdot C$, то це означає, що потрібно брати число $0,01 \cdot 17 = 0,17$.

Наприклад, якщо потрібно провести розрахунки за формулою $A = \frac{B-C}{D}$,

для наступних числових значень параметрів $B = 10$, $C = 5$, $D = 8$, то в підрозділі буде наведено малюнок, в якому видно фрагмент вікна електронної таблиці, де колонку A займають тестові визначення невідомих у формулі, колонку B – їх числові значення. Вікно f_x містить саму формулу розрахунку, де вказано адреси клітинок, які містять числові дані.

	B4		$f_x = (B1-B2)/B3$	
	A	B	C	D
1	B=	10		
2	C=	5		
3	D=	8		
4	A=	0,625		

Якщо будуть застосовані функції електронних таблиць, то буде показано їх вікно з уведеними туди параметрами.

Кожне виконане завдання повинно містити опис задачі, початкові значення змінних, які обираються за номером по списку студентської групи, вирішення та висновки щодо отриманих результатів.

Титульний лист оформлюється згідно прикладу, поданому нижче.

Міністерство освіти і науки, молоді та спорту України
Державний вищий навчальний заклад
«Національний гірничий університет»
Кафедра електронної економіки та економічної кібернетики

ІНДИВІДУАЛЬНА РОБОТА №1
з дисципліни «Інтерактивний Excel»
Варіант № 5

Розроби(ла)в ст. гр. С1-49-1 Косач-Квітка Л.П.

Прийняв проф., д.т.н. Пістунов І.М.

Набуті знання

Щоб по-справжньому ефективно працювати в Smart Excel, недостатньо знати лише базові функції. Необхідно освоїти розширені можливості програми, навчитися створювати інтерактивні елементи та автоматизувати робочі процеси. Завдяки правильному підходу Excel перестає бути просто інструментом для обчислень і перетворюється на повноцінну платформу для аналітики та прийняття рішень.

Цей посібник допоможе вам крок за кроком розібратися з усіма необхідними навичками та зробити роботу в Excel не лише ефективною, а й захопливою

Основні знання для роботи в Smart Excel:

1. Формули та функції. Основою будь-яких розрахунків у Excel є формули. Від найпростіших (SUM, AVERAGE, IF) до складних масивних функцій (INDEX, MATCH, XLOOKUP) – знання цих інструментів дає можливість швидко аналізувати та обробляти великі обсяги інформації.

2. Умове форматування. Візуалізація даних допомагає швидше сприймати результати розрахунків. Умове форматування дозволяє автоматично змінювати колір комірок залежно від значень, створювати динамічні діаграми та виділяти критично важливу інформацію.

3. Елементи керування. Кнопки, випадаючі списки, повзунки та чекбокси – це саме ті інструменти, які роблять розрахунки в Excel керованими. Вони дозволяють створювати інтуїтивно зрозумілі інтерфейси, які спрощують взаємодію користувача з таблицею.

4. Масиви та динамічні діапазони. Сучасні версії Excel підтримують динамічні масиви, що значно розширює можливості автоматизації та аналітики.

Функції FILTER, UNIQUE, SORT дозволяють працювати з даними гнучкіше та швидше.

5. Автоматизація з VBA. Visual Basic for Applications (VBA) – це мова програмування, яка дає змогу створювати макроси, автоматизувати повторювані завдання та реалізовувати складні алгоритми, які недоступні через стандартний функціонал Excel.

6. Power Query та Power Pivot. Ці інструменти дозволяють працювати з великими обсягами даних, витягувати їх із різних джерел та виконувати складну обробку без потреби вручну оновлювати таблиці.

Набуті знання

Вивчення можливостей інтерактивних розрахунків у Excel надає студентам вміння:

1. Будувати динамічні моделі для аналізу даних.
2. Проводити аналіз чутливості та оптимізації на основі різних сценаріїв.
3. Використовувати потужні інструменти для прогнозування та планування.
4. Створювати інтерактивні звіти, що дозволяють аналізувати дані та приймати обґрунтовані рішення.

Ці навички не тільки допомагають студентам у навчанні, а й є безцінними в професійній діяльності в багатьох галузях, таких як фінанси, маркетинг, управління проектами, аналітика та наука.

А більш детально треба визначити наступне.

1. Основи роботи з таблицями та даними
 - Структурування даних: студент навчиться організовувати дані в таблицях, правильно форматовувати їх і застосовувати фільтри для легкого аналізу.
 - Обчислення за допомогою формул: студент освоїть основи введення та використання формул для розрахунків, таких як сума, середнє, стандартне відхилення, мінімум і максимум.

- Використання функцій: студент навчиться використовувати стандартні функції Excel, такі як SUM, AVERAGE, IF, VLOOKUP, INDEX, MATCH та інші для розв'язування складніших задач.

2. Створення та налаштування інтерактивних елементів

- Слайдери, випадаючі списки, кнопки: студент навчиться створювати і використовувати інтерактивні елементи, які дозволяють змінювати дані безпосередньо в таблиці. Це може бути корисним для побудови моделей "що-якщо" або для створення звітів, де користувач може змінювати вхідні дані.

- Фільтри та сортування: студент освоїть використання різних інструментів для швидкого сортування та фільтрації даних для зручності аналізу.

3. Аналіз чутливості та сценаріїв

- Сценарії (Scenario Manager): студент дізнається, як створювати різні сценарії для аналізу впливу різних умов на результат (наприклад, варіанти вартості або зміни вхідних даних).

- Таблиці даних (Data Tables): навчиться використовувати таблиці даних для виконання аналізу "що-якщо", коли змінюється один або два параметри і потрібно побачити їхній вплив на результат.

- Підбір параметрів (Goal Seek): студент зможе знаходити значення вхідної змінної, яке дасть бажаний результат на виході, що є корисним для оптимізації бізнес-процесів чи фінансових моделей.

4. Побудова графіків та діаграм

- Створення діаграм: студент навчиться будувати різні типи діаграм (лінійні, стовпчикові, кругові, комбінації діаграм і т.д.), що дозволяють наочно візуалізувати дані.

- Інтерактивні графіки: студент дізнається, як додавати фільтри до графіків, щоб користувач міг інтерактивно змінювати виведені дані.

- Динамічні графіки за допомогою формул: навчиться створювати графіки, де відображення даних змінюється в залежності від значень у клітинках або при використанні формул.

5. Прогнозування та аналіз тенденцій

- Прогнозування (Forecasting): студент освоїть інструменти для прогнозування майбутніх значень на основі історичних даних, використовуючи такі методи, як експоненціальне згладжування (ETS).

- Прогнозні аркуші: навчитися створювати прогнозні аркуші, які автоматично генерують майбутні значення на основі наданих даних, з візуалізацією та довірчими інтервалами.

6. Моделювання та оптимізація

- Побудова фінансових і бізнес-моделей: студент зможе створювати інтерактивні моделі для бізнес-аналізу, управлінських рішень, фінансового планування, бюджету тощо.

- Моделювання "що-якщо": навчитися створювати моделі, які дозволяють змінювати вхідні параметри і дивитися, як це впливає на результат, що є корисним для бізнес-аналітики та фінансових прорахунків.

7. Автоматизація та розширення можливостей

- Використання макросів: студент може ознайомитися з основами програмування в Excel за допомогою макросів (VBA), щоб автоматизувати повторювані процеси або створити інтерактивні додатки.

- Робота з даними з різних джерел: навчитися імпортувати дані з різних джерел (CSV, текстові файли, бази даних, інтернет-джерела) і здійснювати їх подальший аналіз.

8. Оцінка та візуалізація результатів

- Створення звітів і презентацій: студент навчитися підготовлювати звіти за допомогою інтерактивних таблиць і графіків, що можуть бути використані для презентацій чи прийняття рішень.

- Динамічні таблиці: створення та використання зведених таблиць, які дозволяють аналізувати великі обсяги даних і легко змінювати їх виведення за допомогою різних фільтрів та налаштувань.

9. Прогнозування за допомогою інструментів для аналізу даних

- Пункт меню "Прогнозування" (Forecast Sheet): студент зможе створювати прогнози на основі історичних даних, автоматично генеруючи графіки та оцінки майбутніх значень.
- Аналіз чутливості та оптимізація: використання інструментів для оптимізації рішень і аналізу чутливості (наприклад, аналіз сценаріїв, підбір параметрів), що дозволяє оцінити різні варіанти виходячи з різних умов.

Розділ 1.

НАЛАШТУВАННЯ EXCEL

Вивчивши матеріали цього розділу студенти набувають знань щодо підключення додаткових можливостей Excel.

Окрім знань функцій Excel, які студенти набувають в курсі Інформатики, для ефективної роботи з інтерактивними розрахунками потрібні певні інструменти та налаштування:

- Актуальна версія Excel (краще за все Microsoft 365 або Excel 2019/2021, які підтримують нові функції);
- Доступ до макросів та редактора VBA для створення автоматизованих рішень;
- Розширення Power Query для роботи з великими обсягами даних;
- Готові шаблони та макроси, які можна адаптувати під власні потреби;
- Знання гарячих клавіш, які значно пришвидшують роботу з Excel (наприклад, Ctrl + T для створення таблиць, Alt + H + O + I для автоширини комірок).

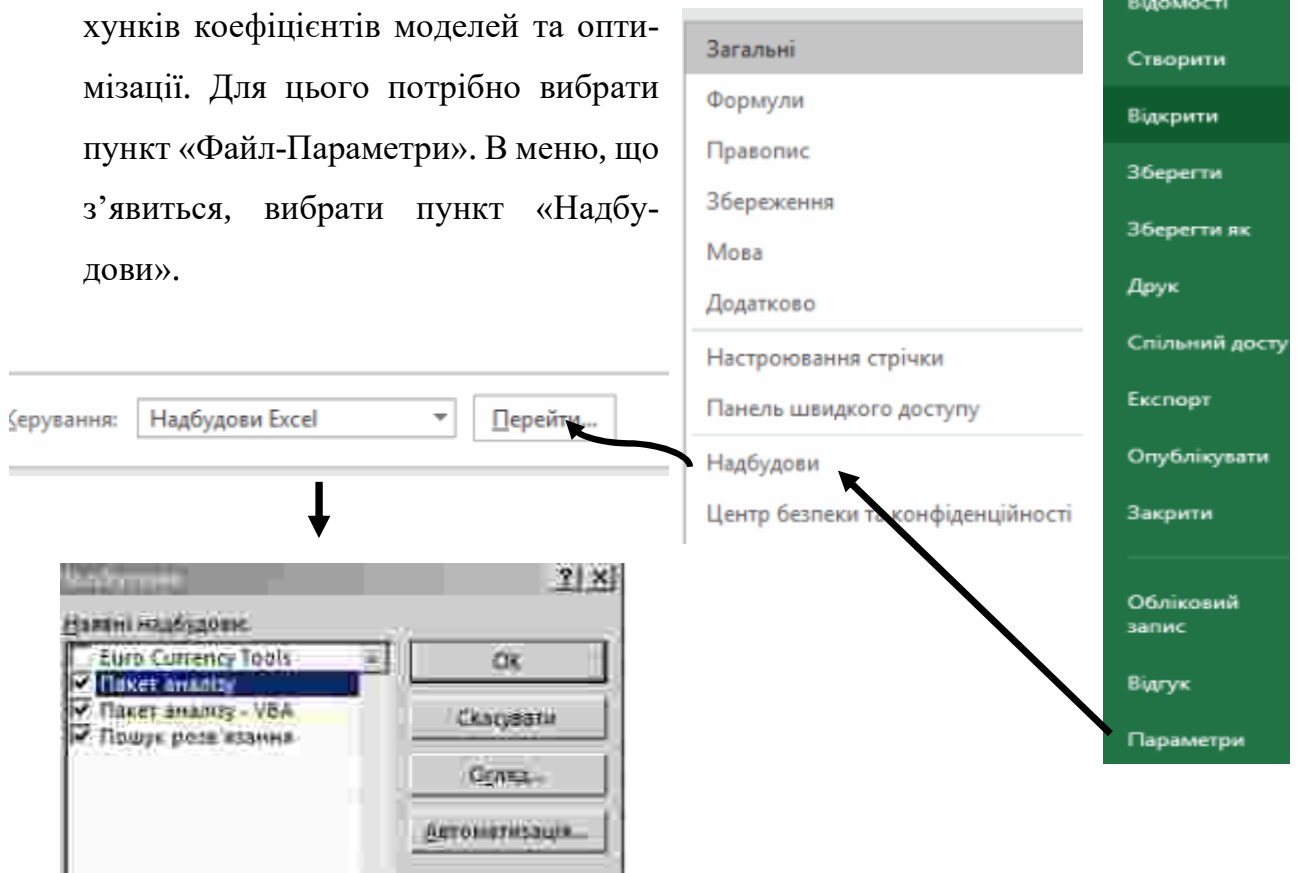
1.1. Налаштування доступу до підпрограм в Excel

Окрім вже відомих студентам функцій Excel, у цій програмі існують додаткові можливості у вигляді підпрограм, які об'єднують в собі декілька функцій, видаючи на сторінки Excel цілу низку готових розрахунків.

Деякі з цих підпрограм будуть описані у подальших розділах, щоб інтерактивні можливості розрахунків можна було б розповсюдити і на них.

Розглянемо порядок такого підключення:

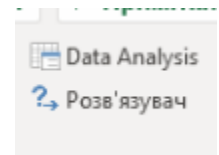
1. Після інсталяції Microsoft Office, куди входить і програма Excel, необхідно провести налаштування цієї програми для розрахунків коефіцієнтів моделей та оптимізації. Для цього потрібно вибрати пункт «Файл-Параметри». В меню, що з'явиться, вибрати пункт «Надбудови».



2. Далі клацнути на кнопку «перейти» та відмітити вказані на рисунку позиції і натиснути «ОК» у цьому і наступних вікнах. Тепер в головному меню програми за пунктом «Дані» у правому кутку меню з'являться пункти «Data Analysis» та «Розв'язувач».

3. Доступ до надбудов здійснюється через головне меню «Дані» і у правому кутку можна побачити ці надбудови.

На рис. 1.1 показано інтерфейс підпрограми Розв'язувач.



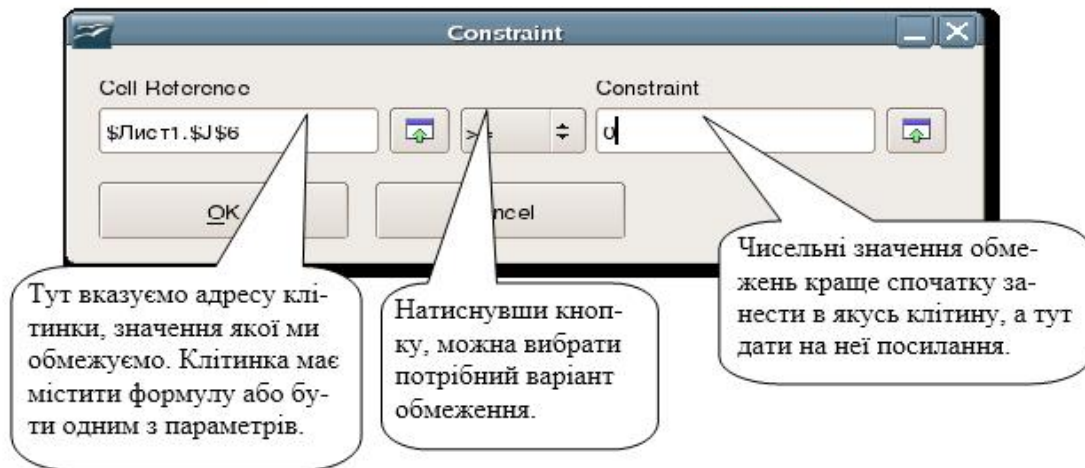
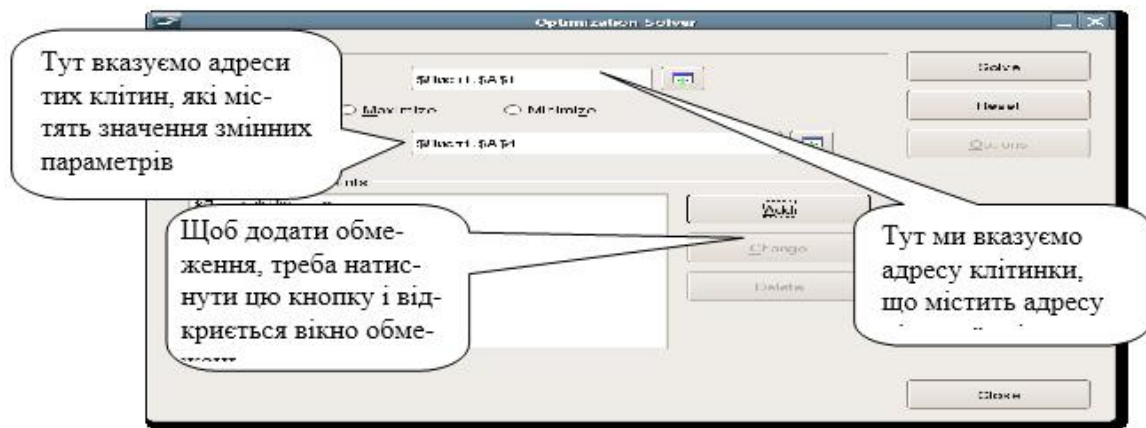
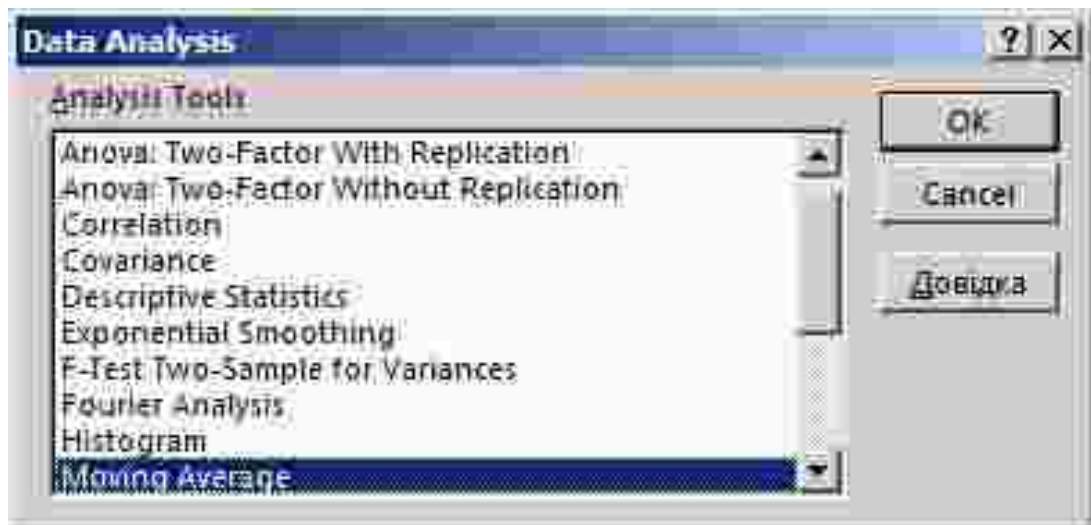
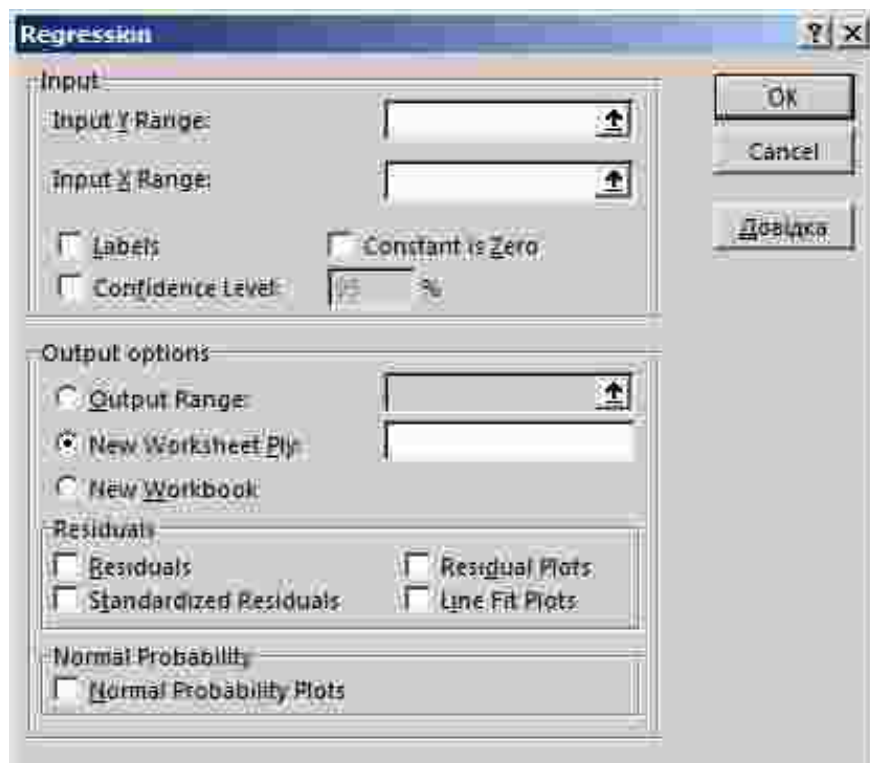


Рис.1.1. Інтерфейс підпрограми Розв'язувач, яка має кодову назву Solver

4. При використанні функції «data Analysis» ви обираєте зі списку потрібний вид аналізу

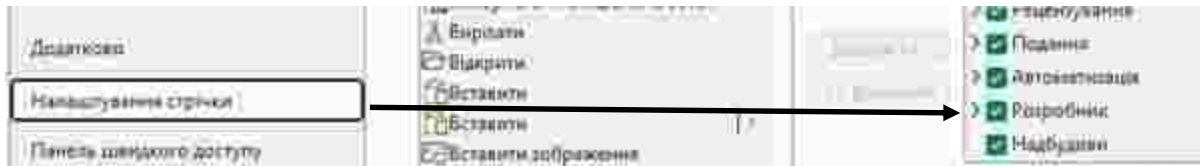


Наприклад, якщо обираєте пункт «Регресія», то побачите наступне вікно



1.2. Підключення меню розробника

Спочатку потрібно по меню **Файл-Параметри-Настроювання стрічки** викликати вікно з параметрами Excel. Далі вибрати пункти «Найуживаніші команди» та «Основні вкладки». Потім під останнім пунктом відмітити галочкою пункт «Розробник». Тоді на панелі інструментів з'явиться додатковий пункт, що дозволяє створювати різні форми та макроси.



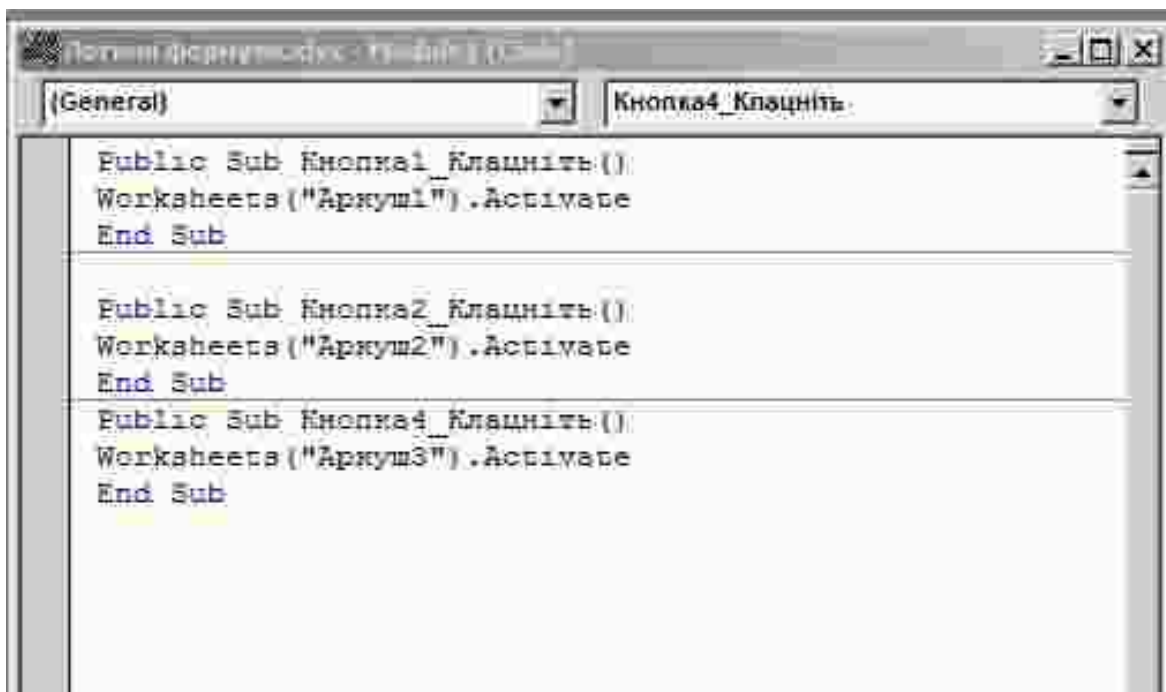


Рис. 1. 2. Список макросів у редакторі Visual Basic

У прикладі, Кнопка_1 та Кнопка_2 розташовані на листі «Аркуш3» і відправляють розрахунок на відповідний лист. Кнопка_3 вміщена на листах «Аркуш1» та «Аркуш2» і призначена для повернення в головне меню.

Далі, розташовуємо слова Так та Ні у стовпчик. Потім створюємо елементи керування форми типу «Поле зі списком» над кожною клітинкою, що містила 0 чи 1. На рис. 1.3 це

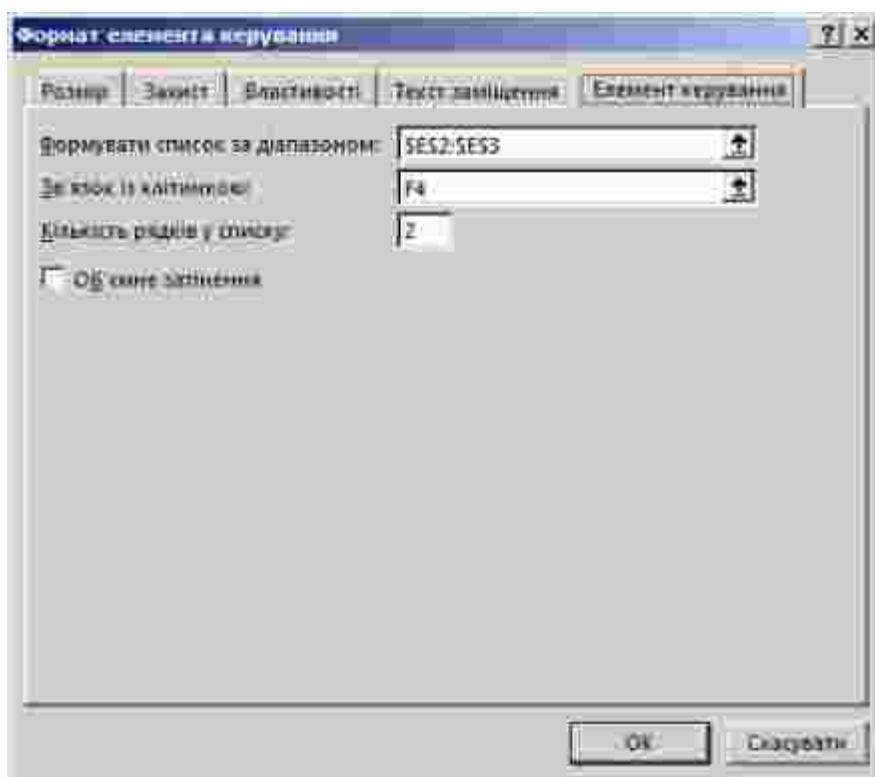


Рис. 1.3. Вікно керування елементом «Поле зі списком»

клітинки E22-E24. Клацаємо лівою кнопкою мишки і обираємо пункт контекстного меню «Формам елемента керування».

1.3. Налаштування інструментів Power Query та Power Pivot

Щоб налаштувати Power Query та Power Pivot у Excel, виконайте наступні кроки:

Power Query:

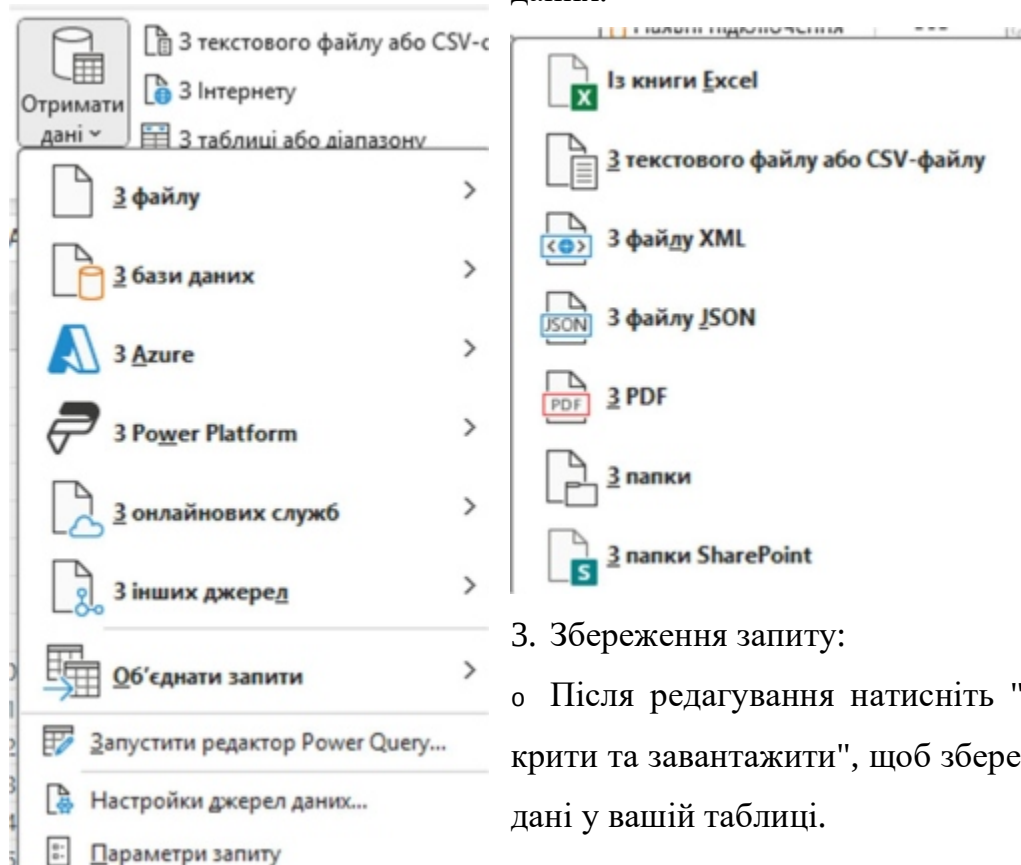
1. Увімкнення Power Query:

- У Excel перейдіть до вкладки "Дані".
- Знайдіть розділ "Отримати та перетворити дані". Це і є Power Query.

2. Імпорт даних:

- Натисніть "Отримати дані" та виберіть джерело (Excel, база даних, веб-сайт тощо).
- Використовуйте редактор запитів для очищення та трансформації

даних.



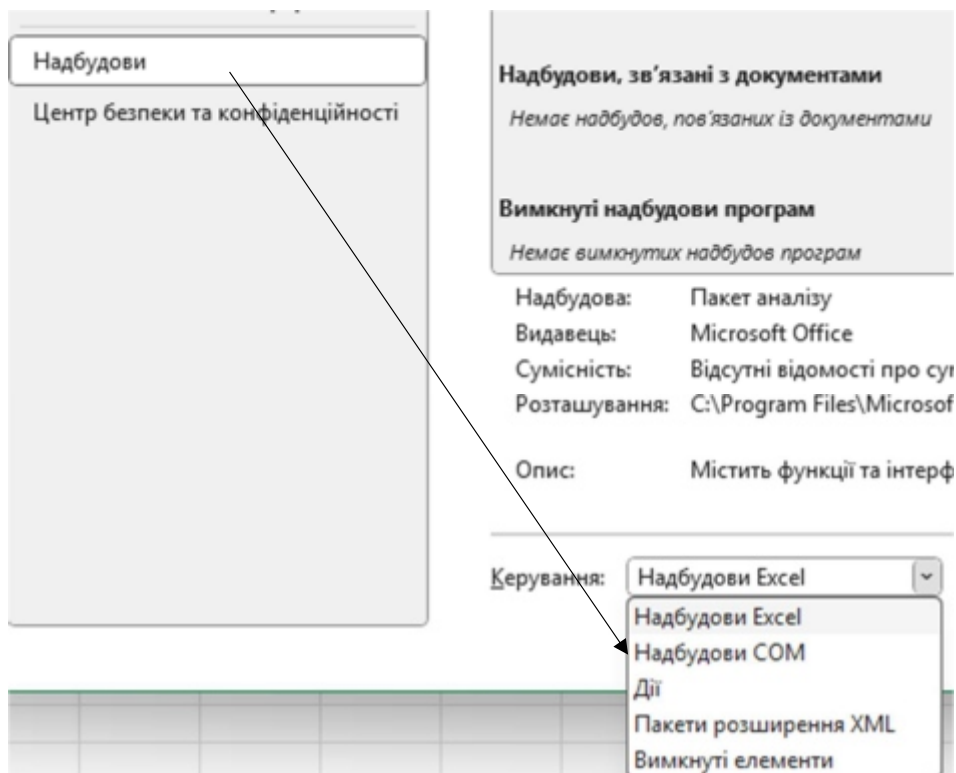
3. Збереження запиту:

- Після редагування натисніть "Закрити та завантажити", щоб зберегти дані у вашій таблиці.

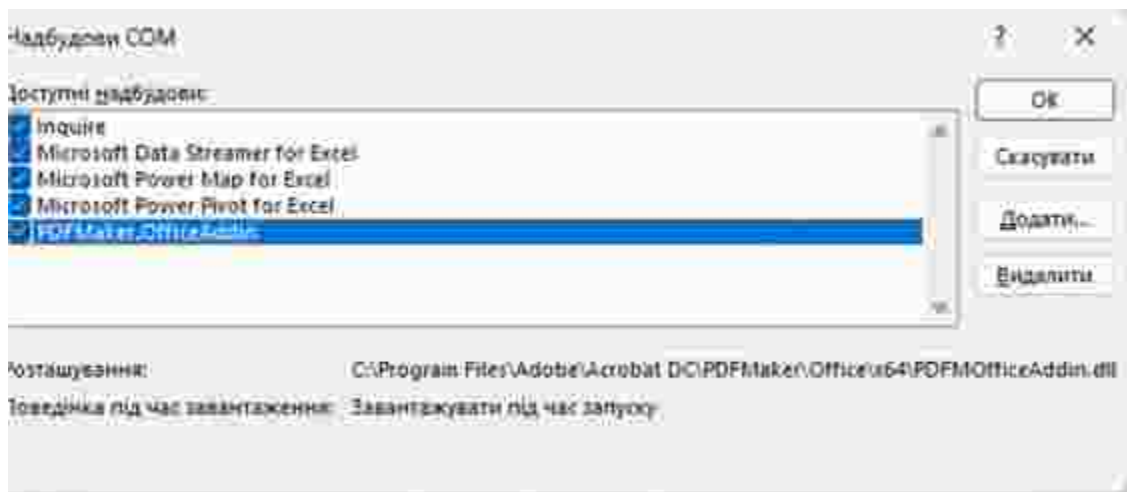
Power Pivot:

1. Увімкнення Power Pivot:

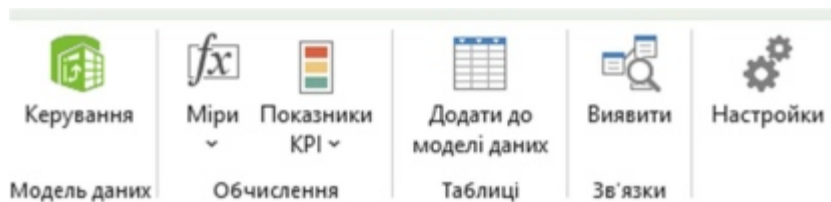
- Перейдіть до "Файл" → "Параметри" → "Надбудови".
- У розділі "Керування" виберіть "COM-надбудови" та натисніть "Перейти".
- Увімкніть "Microsoft Power Pivot для Excel".



Відкриється вікно, у якому треба відмітити всі пункти



Тоді в головному меню з'явиться пункт Power Pivot



2. Додавання даних до моделі даних:

- о Виділіть таблицю та натисніть "Додати до моделі даних" на вкладці Power Pivot.

3. Створення зв'язків між таблицями:

- о У Power Pivot використовуйте "Управління" для створення зв'язків між таблицями.

4. Використання DAX:

- о Використовуйте мову DAX для створення обчислюваних стовпців, мір та KPI.

1.4. Індивідуальне завдання №1. Налаштування Excel

Завдання: Інсталювати на вашому комп'ютері додаткові можливості Excel.

Порядок виконання: використовуються функції Excel, а також браузері на вашому комп'ютері.

Методичні вказівки: 1) Інсталювати підпрограми Розв'язувач, Data Analysis та мову програмування VBA.

- 2) Підключити меню «Розробник».
- 3) Підключити меню Power Pivot.
- 4) Скласти звіт з демонстрацією необхідних вікон.

Контрольні запитання

1. Як можна отримати доступ до підпрограм (макросів) в Excel?
2. Для чого використовується меню "Розробник" в Excel?
3. Які основні можливості надають інструменти Power Query та Power Pivot?
4. Як активувати підтримку макросів в Excel?
5. Що таке VBA, і як його використовувати в Excel?
6. Якими кроками можна додати вкладку "Розробник" в Excel?
7. Які інструменти доступні у вкладці "Розробник"?
8. Як активувати Power Query в Excel?
9. Яким чином Power Pivot розширює можливості роботи з даними?
10. Чим Power Query відрізняється від стандартних функцій обробки даних в Excel? Як можна змінити рівень безпеки макросів в Excel?
11. Які основні переваги використання Power Query для роботи з великими обсягами даних?
12. Чи можна використовувати Power Pivot без використання формул DAX? Чому?
13. Які формати даних можна імпортувати за допомогою Power Query?
14. Як зробити так, щоб Excel автоматично виконував завантаження та оновлення даних у Power Query?

У розділі студенти засвоїли основний порядок підключення підпрограм, таких як Solver та Data Analysis, підключення меню розробника, а також порядок звернення до таких можливостей Excel як Power Query та Power Pivot.

Розділ 2.

ОСНОВИ АВТОМАТИЗАЦІЇ В EXCEL

Вивчивши матеріали цього розділу студенти набувають знань щодо основ автоматизації Excel.

Автоматизація обчислень у Microsoft Excel є важливим аспектом роботи з електронними таблицями. Використання динамічних формул дозволяє значно підвищити ефективність розрахунків, зменшити кількість помилок і зробити дані більш інтерактивними. У цьому розділі розглянемо основні концепції динамічних формул, їхню структуру, принципи використання та приклади застосування.

Динамічні формули – це обчислення, які можуть автоматично оновлювати свої результати у відповідь на зміну вхідних даних. Вони забезпечують можливість автоматичної адаптації до змін у розмірах таблиць, значеннях осередків або умовах розрахунку.

Використання динамічних формул у Excel дозволяє значно підвищити ефективність обробки даних. Вони допомагають автоматизувати розрахунки, зменшити ймовірність помилок і зробити таблиці більш гнучкими та інтерактивними. Важливим аспектом є правильний вибір формул залежно від потреб завдання, а також використання сучасних інструментів, таких як динамічні масиви, Power Query та Power Pivot. Освоєння цих навичок дає змогу значно розширити можливості використання Excel у бізнесі та аналітиці.

Головними перевагами динамічних формул є:

- Автоматичне оновлення результатів при зміні вхідних даних.
- Використання розумних посилань для збереження цілісності даних.
- Мінімізація ручної роботи користувача.

2.1. Відносні, абсолютні та змішані посилання

При створенні формул в Excel важливо розуміти, як працюють посилання.

Відносні посилання, наприклад – «A1» змінюються при копіюванні формули. У прикладі буква «A» означає індекс стовпця, а цифра «1» – номер рядка на якому знаходиться комірка, що приймає участь у розрахунках.

У формулі $=1/(F280+\$B\$283)$ видно, що комірка з адресою F280 є змінною. Якщо цю формулу скопіювати на стовпець правіше, то адреса клітинки стане G280, бо стовпець, правіший від стовпця F є G. Якщо скопіювати на рядок нижче, то адреса зміниться на F281.

Символ \$ перед індексом стовпця не дає можливості міняти букву індексу при копіюванні. Так само, символ \$ перед номером рядка забороняє зміну номеру рядка при копіюванні. Такі посилання називаються абсолютними.

Абсолютні посилання (\$A\$1) залишаються незмінними незалежно від того, куди переміщено формулу. Створення абсолютного посилання можливе натисканням кнопки F4.

Змішані посилання (\$A1 або A\$1) фіксують лише рядок або стовпець.

Змішані посилання дозволяють спростити складні розрахунки.

Наприклад, маємо матрицю відповідей експертів, позначених як E1...E6 на запитання, позначені як Зап1...Зап5. Відповіді експерти дають у вигляді певних числових значень.

	A	B	C	D	E	F
3		Зап1	Зап2	Зап3	Зап4	Зап5
4	E1	72	84,48	11,87845304	7,596154	170,2
5	E2	77	106,24	13,8121547	6,634615	270,1
6	E3	71	67,84	13,8121547	4,134615	203,5
7	E4	41	56,32	12,70718232	8,846154	210,9
8	E5	57	117,76	22,6519337	9,038462	236,8
9	E6	56	89,6	17,12707182	4,807692	177,6

Необхідно для відповідей на кожне запитання визначити його ранг, тобто порядковий номер у зростаючому порядку. Наприклад, для Зап1 число 41 матиме ранг 1, а число 77 – ранг 6.

Існує функція, що виставить ці ранги автоматично, треба тільки задати адресу клітинки, для якої знаходиться ранг, та адресу груп клітинок, серед яких треба вистати ранги.

RANK(number, ref, [order])

Опис параметрів:

1. **number** - Число, для якого необхідно знайти ранг.
2. **ref** - Масив або діапазон чисел, серед яких ви хочете визначити ранг.
3. **[order]** - Необов'язковий параметр, який визначає порядок ранжування:
 - o Якщо order = 0 або параметр пропущений, ранжування буде за спаданням (найбільше число отримує ранг 1).
 - o Якщо order = 1, ранжування буде за зростанням (найменше число отримує ранг 1).

Для того ж прикладу, для Зап1, розташованого у стопці В та для числа, розташованого у комірці В4 функція має вигляд

=RANK(B4, B4:B9, 0) // За спаданням або

=RANK(B4, B4:B9, 1) // За зростанням

Але якщо потрібно знайти ранги для всього стовпця, то копіювання цієї формули на рядки, розташовані нижче, призведуть до зміни адрес діапазону чисел, серед яких ви хочете визначити ранг.

Отже, формулу треба переписати як =RANK(B4, \$B\$4:\$B\$9, 0)

А як бути, коли ми хочемо повторити цю формулу для сусідніх стовпців? Адже при цьому адреси \$B\$4:\$B\$9 не зміняться! Тобі треба прибрати знак \$ перед індексом стовпця =RANK(B4, B\$4:B\$9, 0) і ми отримуємо матрицю наступну рангів.

	A	B	C	D	E	F
12		Зап1	Зап2	Зап3	Зап4	Зап5
13	E1	5	3	1	4	1
14	E2	6	5	3	3	6
15	E3	4	2	3	1	3
16	E4	1	1	2	5	4
17	E5	3	6	6	6	5
18	E6	2	4	5	2	2

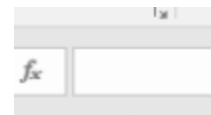
2.2. Майстер функцій

Функцію можна вводити в комірку в рядку формул або безпосередньо в комірку. Другий спосіб не є оптимальним, оскільки вимагає знання точного імені функції.

Майстер функцій – це спеціальна підпрограма, за допомогою якої можна вибрати потрібну функцію і виконати її, вказавши всі потрібні параметри.

Майстер функцій можна викликати таким чином:

- 1) Натисканням кнопки **Майстер функцій** (f_x) в рядку формул ;
- 2) Натисканням гарячих клавіш Shift+F3.



Вікно **Майстра функцій** складається з трьох частин (рис. 2.1). У першій можна ввести опис дії, яку необхідно виконати і натиснути кнопку **Знайти**. Цей метод використовується, якщо користувач не знає чи не пам'ятає, як називається потрібна йому функція.

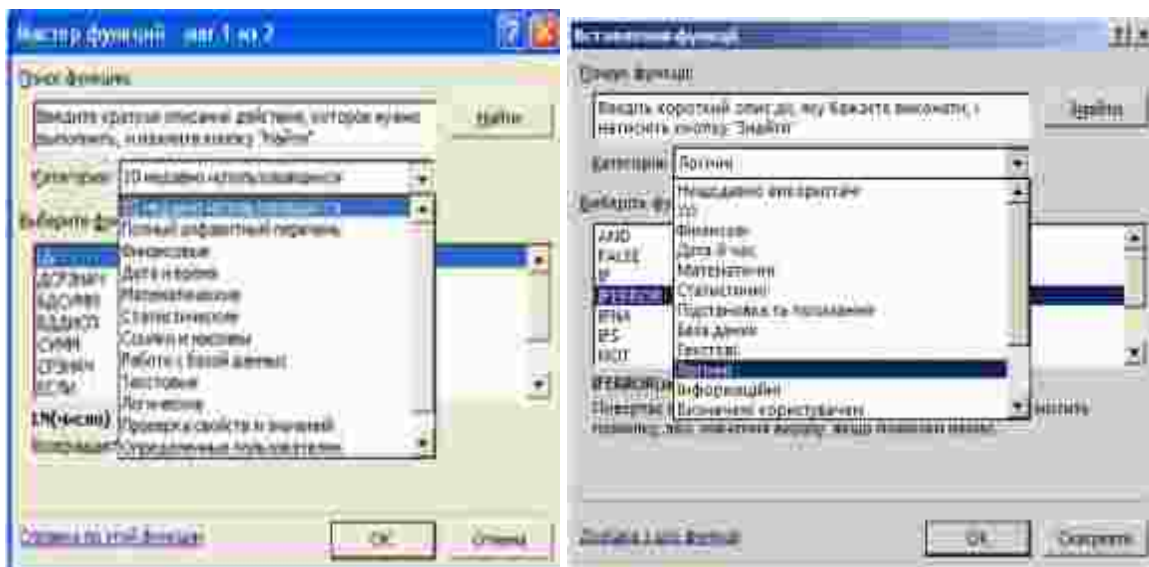


Рис.2.1 Вікно майстра функцій в різних версіях Windows

В електронних таблицях Excel часто для проведення розрахунків використовують різноманітні функції.

Функції – це заздалегідь визначені формули, що виконують обчислення за заданими величинами (аргументами) в зазначеному порядку.

Функції дозволяють виконувати як прості, так і складні обчислення. Функції в Excel використовуються для виконання стандартних обчислень. Значення, що використовуються для обчислення функцій, називають аргументами. Значення, що повертаються функціями як відповідь, називають результатом. Крім вбудованих функцій, можна використовувати в обчисленнях функції користувачів, що створюються за допомогою засобів Excel.

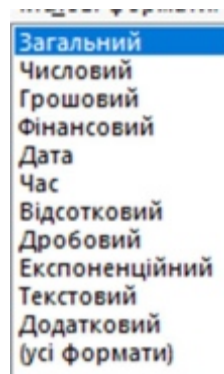
Щоб використати функцію, потрібно ввести її як частину формули в комірку робочого аркуша. Послідовність, у якій мають розміщуватися використовувані у формулі символи, називають синтаксисом функції. Всі функції використовують однакові основні правила синтаксису. Якщо порушити правила синтаксису, то в Excel з'явиться повідомлення про помилку у формулі.

Для спрощення роботи з функціями більшість із них була названа від скорочення російськомовних значень цих функцій:

Загальний вигляд функції починається зі знаку «дорівнює»:

=ім'я функції (параметри)

Якщо параметрів декілька, вони записуються через знак «крапка з комою». Існують різні типи аргументів, які можна побачити, викликавши контекстне меню і обравши пункт «Формати».



Довідку про необхідну функцію можна одержати, якщо вибрати її зі списку в довідковій системі Excel або натиснути кнопку F1.

Довідку про функції під час її введення в комірку робочого листа можна одержати за допомогою **Помічника**. Якщо ви знаходитесь у вікні діалогу **Майстра функцій**, то на панелі інструментів необхідно вибрати інструмент, який позначається знаком «?» для виклику **Помічника**.

Якщо ви вводите формулу безпосередньо в комірку робочого листа, то просто наведіть курсор миші в рядок формул та натисніть F1. Під час діалогу деталізуйте тему довідки – введіть ім'я функції або її частину.

У наступному діалоговому вікні необхідно виділити одну з запропонованих функцій, початок яких збігається з введеним фрагментом.

2.3 Використання логічних функцій для динамічних розрахунків

Логічні функції допомагають створити складні формули, що залежно від виконання тих чи інших умов робитимуть різні види обробки даних.

Ці функції набувають логічних значень 1 або 0, що відповідає значенням «Істина» або «Хибно». У Excel замість числових значень допускається використання слів TRUE – відповідає значенню 1, або FALSE – відповідає значенню 0. Причому, всі ці функції працюють однаково як з цифрами, так і зі словами. Ця категорія функцій містить небагато функцій, але вона є дуже важливою і часто використовується. В різних версіях Windows використовується як англійська версія функцій так і русифікована. В подальшому викладенні будуть вживатися обидві форми написання, англійська та російська, остання буде записуватися в дужках).

До логічних функцій відносять такі функції: IF (ЕСЛИ), AND (И), OR (ИЛИ), TRUE (ИСТИНА), FALSE (ЛОЖЬ), NO (НІ).

1) Логічна функція IF (ЕСЛИ)

Функція IF (ЕСЛИ) використовується для розв'язання задач, в яких необхідно перевірити деяку умову, і залежно від того, виконується вона чи ні, повертає одне з двох значень.

Ця функція записується таким чином:

=IF(Логічний вираз, який може містити всі логічні функції, окрім IF; Значення, якщо вираз істинний; Значення, якщо хибний).

В якості значення можуть бути будь які типи змінних або формули, що перетворюють змінні. У тому числі й сама функція **IF**.

Якщо логічний вираз після розрахунку має значення TRUE (ИСТИНА) або 1, то розраховується значення аргументу **Значення, якщо вираз істинний**, якщо значення умови після розрахунку буде FALSE (ЛОЖЬ) або 0 – значення аргументу **Значення, якщо хибний**. При цьому аргументи можуть мати вигляд вбудованої функції IF (ЕСЛИ) (рис. 2.2). У разі складання перевірок їх буває до семи.

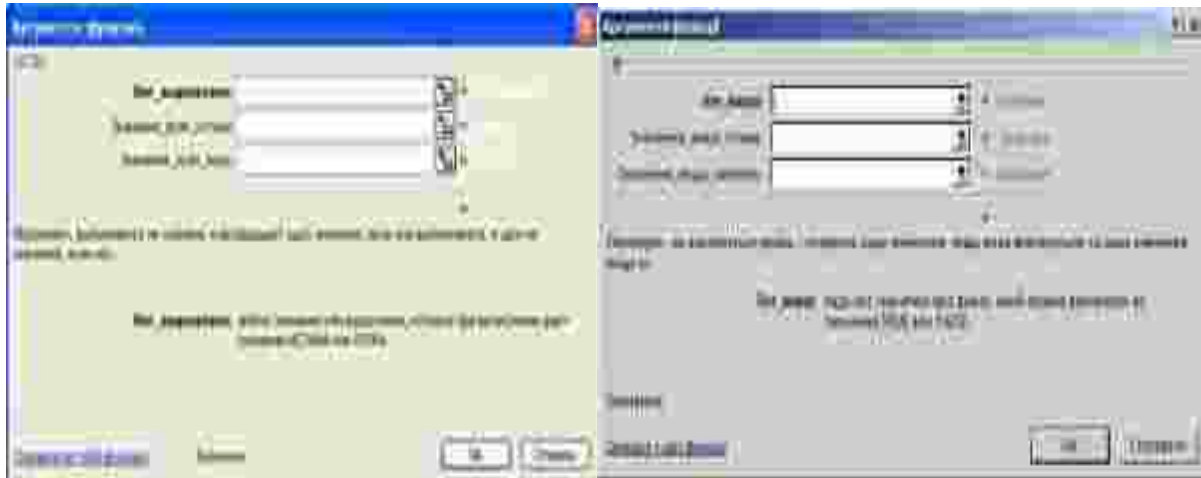


Рис. 2.2 Логічна функція IF (ЕСЛИ) для різних версій Windows

Приклад. Необхідно розрахувати функцію

$$Y = \begin{cases} X + 1, & \text{якщо } X \leq 1, \\ X - 1, & \text{якщо } X > 1. \end{cases}$$

Розв'язання

Для розв'язання даної задачі необхідно скористатися функцією IF (ЕСЛИ). Серед аргументів функції IF (ЕСЛИ) є логічний вираз, де необхідно перевірити виконання умови. Тут необхідно перевірити одну з умов, що задані в прикладі. Умову вибираємо довільно (~~$X \leq 1$~~ або $X > 1$). Перевіримо умову $X > 1$. Якщо ця умова виконується, ми потрапляємо в проміжок $X > 1$, де $Y = X + 1$, а якщо ця умова не виконується, то ми потрапляємо у проміжок $X \leq 1$, де

$Y = X - 1$. Розрахунок даного виразу за допомогою електронної таблиці Excel зображений на рис. 2.3.



	B	C	D
		Значення X	Значення Y
		1	2
		2	1
		3	2
		4	0

Рис. 2.3 Розрахунок виразу в Excel

2) Логічна функція AND (И)

Функція **AND (И)** повертає значення Істина, якщо всі аргументи мають значення «True» або «Істина» або 1.

Синтаксис функції буде таким:

=AND(логічне_значення_1;логічне_значення_2;..).

Її використовують для об'єднання двох і більше умов.

3) Логічна функція FALSE (ЛОЖЬ)

Функція **FALSE (ЛОЖЬ)** повертає логічне значення FALSE або ЛОЖЬ або 0.

Синтаксис функції буде таким

=FALSE() або ЛОЖЬ().

4) Логічна функція NO (НЕ)

Функція **NO (НЕ)** – змінює на протилежне логічне значення аргументу.

Якщо аргумент має значення «1», при використанні даної функції значення зміниться на протилежне, тобто на 0.

5) Логічна функція OR (ИЛИ)

Функція OR (ИЛИ) – повертає логічне значення істина, якщо коли хоч один з аргументів має значення істина.

6) Логічна функція TRUE (ИСТИНА)

Функція TRUE (ИСТИНА) – повертає логічне значення «TRUE» або «Істина» або 1.

В офісі 360 розробники додали іще декілька функцій, що дозволяють полегшити створення формул за логічними формулами.

7) Функція IFS

Функція IFS перевіряє, чи виконується одна або кілька умов, і повертає значення, що відповідає першій умові зі значенням TRUE. Нею можна замінити кілька вкладених операторів IF, щоб спростити сприйняття формули з кількома умовами.

Синтаксис

IFS(лог_вираз1; значення_якщо_істина1; [лог_вираз2; значення_якщо_істина2]; [лог_вираз3; значення_якщо_істина3];...)

Функція IFS дає можливість перевірити до 127 різних умов, фактично підвищуючи рівень вкладеності функції IF. Її відміною від IF треба вважати вимогу дотримання

Зазвичай не рекомендується використовувати забагато умов з операторами IF або IFS. Оскільки їх необхідно вводити в чітко заданому порядку, такі вирази дуже складно створювати, перевіряти й оновлювати.

Наприклад, це означає «Якщо (A1 дорівнює 1, відображається 1, якщо A1 дорівнює 2, відображається 2, а якщо A1 дорівнює 3, відображається 3)» або якщо записати це у вигляді формули

=IFS(A1=1;1;A1=2;2;A1=3;3)

Повертаючись до прикладу з функцією **IF**, цю функцію можна було застосувати наступним чином

$$=IFS(B1 \leq 1; C1 + 1; B1 > 1; C1 - 1)$$

На рис. 2.4 наведено іще один приклад використання функції IFS.

	Оцінка	Позначка	Результат
2	93	A	A, оскільки A2 > 89
3	89	B	B, оскільки B3 > 79
4	71	C	B, оскільки B4 > 69
5	60	D	Г, оскільки A5 > 59
6	58	F	Г, оскільки значення 58 не відповідає попереднім умовам, TRUE відповідає значення Г – це значення за замовчуванням, оскільки не виконувались інші умови.
7	58	F	Г, оскільки значення 58 не відповідає попереднім умовам, TRUE та відповідає значення Г – це пара значень за замовчуванням, яка використовується, коли інші умови не виконуються.

Рис. 2.4. Приклад використання функції IFS

Формула для клітинок A2:A6 має такий вигляд:

$$=IFS(A2 > 89; "A"; A2 > 79; "B"; A2 > 69; "B"; A2 > 59; "Г"; TRUE; "Г")$$

Це означає Якщо(A2 більше ніж 89, повертається значення "A", якщо A2 більше ніж 79, повертається "B" й так далі; для всіх інших значень, які менші за 59, повертається "Г").

Наступний приклад стосується операцій з текстовими змінними (рис. 2.5).

Формула в клітинці G7 має такий вигляд:

$$=IFS(F2=1; D2; F2=2; D3; F2=3; D4; F2=4; D5; F2=5; D6; F2=6; D7; F2=7; D8)$$

Це означає Якщо(значення в клітинці F2 дорівнює 1, повертається значення клітинки D2, якщо значення в клітинці F2 дорівнює 2, повертається

значення клітинки D3, і так далі; зрештою, якщо не виконується жодна умова, повертається значення клітинки D8).

Є	Д	Е	Ф	Г	Н
	День тижня		Номер дня	День	
	Неділя		3	Вівторок	
	понеділок				
	вівторок				
	середа				
	четвер				
	п'ятниця				
	Субота				

Рис. 2.5. Використання функції IFS для текстових змінних

Щоб указати результат за замовчанням, введіть для останнього аргументу "лог_вираз" умову, яка завжди виконуватиметься, наприклад TRUE або 1=1. Якщо не виконано жодну іншу умову, повертається те саме значення, яке було до цього в клітинці. У прикладі 1 така ситуація відтворена в рядках 6 і 7 (з оцінкою 58).

Якщо для аргументу *лог_вираз* не задано відповідний аргумент *значення_якщо_істина*, ця функція відобразить повідомлення про помилку: "Для функції введено кілька аргументів (Занадто мало)".

Якщо вираз в аргументі *лог_вираз* повертає значення, відмінне від TRUE або FALSE, функція повертає помилку #VALUE!.

Якщо умови зі значенням TRUE не знайдено, функція повертає помилку #N/A.

8) Функція SWITCH

Функція SWITCH обчислює одне значення (яке називається **виразом**) на основі списку значень і повертає результат, який збігається з першим відповідним значенням. Якщо немає збігів, може повертатися необов'язкове стандартне значення.

У найпростішому випадку функція SWITCH має такий вигляд:

=SWITCH(значення_для_змінення;значення_для_збігу1...[2–126];значення_для повернення_якщо_є_збіг1...[2–126];значення_для_повернення_якщо_немає_збігів)

За допомогою цієї функції можна оцінити до 126 зіставлених значень і результатів.

Розгляньте таку формулу:

The screenshot shows an Excel spreadsheet with the following content:

	A	B	C	D	E	F	G
1	Значення	Результат					
2	10.10.2016	Понеділок					
3							
4	Формула	1	2	3			4
5	=SWITCH(WEEKDAY(A2);1;"Неділя";2;"Понеділок";3;"Четвер";"Немає збігів")						

1. Яке значення для змінення? У цьому випадку WEEKDAY(A2) дорівнює 2.
2. Для якого значення потрібні збіги? У цьому випадку: 1, 2 і 3.
3. Якщо є збіг, чи потрібно повертати його як результат? У цьому випадку це "Понеділок" для значення 1, "Вівторок" для значення 2 та "Середа" для значення 3.
4. Стандартне значення для повернення, якщо немає збігів. У цьому випадку це текст "Немає збігів".

Якщо збігів немає та аргумент "за_замовчуванням" не вказано, функція SWITCH повертає помилку #N/A!.

Щоб дізнатися, як працює функція SWITCH, скопіюйте дані прикладу з наведеної нижче табл. 2.1 та вставте їх у клітинку A1 нового аркуша Excel. А в клітинку A2 вставте число 2, а в клітинки A3 та A4 число 99, в клітинку A5 – число 2, в A6 – 3. Тоді результати розрахунків будуть точно як в таблиці.

Якщо результат обчислення формул не відображається, виділіть їх, натисніть клавішу **F2**, а потім – **Enter**. За потреби можна змінити ширину стовпців, щоб відобразити всі дані.

Таблиця 2.1

Приклади використання функції SWITCH

Значення	Формула	Результат
2	=SWITCH(WEEKDAY(A2);1;"Понеділок";2;"Вівторок";3;"Середа";"Немає збігів")	Оскільки A2=2, а "Вівторок" – це аргумент результату, який відповідає значенню 2, функція SWITCH повертає значення "Вівторок"
99	=SWITCH(A3;1;"Понеділок";2;"Вівторок";3;"Середа")	Через відсутність збігів та іншого аргументу функція SWITCH повертає помилку #N/A!
99	=SWITCH(A4;1;"Понеділок";2;"Вівторок";3;"Середа";"Немає збігів")	Немає збігів
2	=SWITCH(A5;1;"Понеділок";7;"Неділя";"день тижня")	день тижня
3	=SWITCH(A6;1;"Понеділок";2;"Вівторок";3;"Середа";"Немає збігів")	Середа

9) XOR (функція XOR)

Повертає логічну операцію виключного OR для всіх аргументів.

Синтаксис **XOR(лог_значення1;[лог_значення2];...)**

Синтаксис функції XOR має такі аргументи:

лог_значення1;лог_значення2;... – аргумент «лог_значення1» обов'язковий, подальші логічні значення необов'язкові. Від 1 до 254 умов, які потрібно перевірити. Вони можуть бути істинні (TRUE) або хибні (FALSE) і можуть бути логічними значеннями, масивами чи посиланнями.

Аргументи мають приймати логічні значення TRUE чи FALSE або бути масивами чи посиланнями, які містять логічні значення.

Якщо масив або посилання містить текст чи пусті клітинки, ці значення ігноруються.

Якщо вказаний діапазон не містить логічних значень, функція XOR повертає значення помилки #VALUE!.

Щоб переглянути, чи є в масиві значення, можна скористатися формулою масиву XOR. Щоб ввести формулу масиву, натисніть сполучення клавіш Ctrl+Shift+Enter.

Результат функції XOR має значення TRUE, якщо кількість вхідних даних зі значенням TRUE непарна, і значення FALSE, якщо кількість вхідних даних зі значенням TRUE парна.

У табл. 2.2 подано приклади використання функції XOR. Для того, що спрацював третій та четвертий приклад, вставте ці формули у клітинку A4 з прикладу для функції IFS.

Таблиця 2.2

Приклад застосування функції XOR

Формула	Опис	Результат
=XOR(3>0;2<9)	Два перевірені значення істинні.	TRUE

Формула	Опис	Результат
=XOR(3>12;4>6)	Два перевірені значення хибні, тому повертається значення. Щоб повернулося значення TRUE, потрібно, щоб принаймні одне з перевірених значень було істинним.	FALSE
=XOR(A2<2;A3>3)	Два перевірені значення істинні.	TRUE
=XOR(A2=2;A3<3)	Повернулося значення TRUE, бо принаймні одне з перевірених значень було істинним.	TRUE

2.3.1. Використання логічних функцій електронних таблиць Excel для опису висловлювань, представлених у вигляді формул алгебри логіки

Для прикладу, з п. 1.6 візьмемо результати створення формул з алгебри логіки пункти 2 та 4.

$$3) A \rightarrow B \qquad 4) A \rightarrow (B \wedge C)$$

З матеріалів, засвоєних з п. 1.8 ми знаємо, що таких функцій як еквіваленція та імплікація в Excel не існує. Але ми маємо можливість перетворити отримані формули в інші, що міститимуть тільки операції заперечення, кон'юнкції та диз'юнкції.

Ось ці формули, що нам допоможуть

- $x \oplus y = (x \wedge \bar{y}) \vee (\bar{x} \wedge y) = (x \vee y) \wedge (\bar{x} \vee \bar{y})$.
- $x \sim y = \overline{x \oplus y} = (x \wedge y) \vee (\bar{x} \wedge \bar{y}) = (x \vee \bar{y}) \wedge (\bar{x} \vee y)$
- $x \rightarrow y = \bar{x} \vee y = ((x \wedge y) \oplus x) \oplus 1$.

Щоб записати ці формули у логічних функціях, скористаємося прийнятими в них умовними позначеннями:

$$1) = \text{OR}(\text{NOT}(A); B)$$

$$2) = \text{AND}(\text{OR}(A; \text{NOT}(\text{AND}(B; C))) ; \text{OR}(\text{NOT}(A); \text{AND}(B; C)))$$

Якщо замість умовних A та B підставити адреси клітинок та використати умовний оператор IF, отримаємо готову формулу для того, щоб за словесною формулою зробити висновок щодо успішності чи неуспішності заходів у конкретному випадку. Розглянемо приклад.

Приклад. Складне висловлення «Якщо підприємство в сфері промисловості та будівництва України має чисельність працюючих до 200 осіб, то це підприємство належить до категорії малих підприємств за Законом України «Про підприємство в Україні» містить два простих висловлення:

A – «Підприємство в сфері промисловості та будівництва України має чисельність працюючих до 200 осіб»,

B – «Підприємство належить до категорії малих підприємств за Законом України «Про підприємства в Україні».

Висловлення A і B зв'язані за допомогою, «Якщо..., то...». Таким чином, дане початкове складне висловлення представимо формулою:

$$A \rightarrow B.$$

або у формулах Excel

$$=\text{IF}(\text{OR}(\text{NOT}(E14); E15); \text{"male"}; \text{"big"})$$

Тут, логічна у мова оператора IF $\text{OR}(\text{NOT}(E14); E15)$ відповідає логічному перетворенню типу $x \rightarrow y = \bar{x} \vee y$.

На рис. 2.3 подано фрагмент електронних таблиць, що реалізують цю формулу. При зміні значень відповідей, результат міняється. Варіант відповіді – мале – буде показано при умові, що логічна формула отримала значення TRUE. Формула вставлена у клітинці E16.

	A	B	C	D	E	F	G
11	Якщо підприємство в сфері промисловості та будівництва України має чисельність працюючих до 200 осіб, то це підприємство належить до категорії малих підприємств за Законом України «Про підприємство в Україні»						
12		Дайте відповідь на запитання: 1- ТАК, 0 - НІ					
13							
14	Чисельність працюючих до 200 осіб?				1		
15	Чи належить до категорії малих підприємств?				0		
16	Ви маєте платити податки як				велике	підприємство	

Рис. 2.3. Приклад реалізації твердження щодо типу підприємства

Приклад. Складне висловлення «Якщо підприємство реалізує вироблену продукцію, то відшкодовує витрати на виробництво і отримує прибуток».

Позначимо:

A – «Підприємство реалізує вироблену продукцію»,

B – «Підприємство відшкодовує витрати на виробництво»,

C – «Підприємство отримує прибуток».

Таким чином, дане початкове складне висловлення представимо формулою:

$$A \rightarrow (B \wedge C).$$

Реалізація цього твердження, яке визначає успішність підприємства, представлено у вигляді наступної формули

=IF(AND(OR(E23; NOT(AND(E22;E24))); OR(NOT(E23);
AND(E22;E24))); "є";"не є")

На рис. 2.4 подано фрагмент вікна Excel, де реалізовано визначення успішності роботи підприємства. Сама формула стоїть у клітинці D25.

	A	B	C	D	E	F
19	Якщо підприємство реалізує вироблену продукцію, то відшкодовує витрати на виробництво і отримує прибуток					
20		Дайте відповідь на запитання: 1- ТАК, 0 - НІ				
21	Ваше підприємство:					
22	відшкодовує витрати на виробництво				0	
23	реалізує вироблену продукцію?				1	
24	отримує прибуток?				1	
25	Висновок: ваше підприємство		не є		успішним	

Рис. 1.2. Приклад створення рішення щодо успішності підприємства

2.4. Функції для роботи з текстовими даними

Для обробки текстової інформації Excel надає набір функцій:

LEFT(текст; n) – повертає перші n символів рядка.

Наприклад, формула

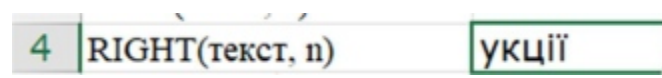
=LEFT("Статистика споживання готової продукції"; 5)

повертає 

RIGHT(текст; n) – повертає останні n символів.

Наприклад, формула

=RIGHT("Статистика споживання готової продукції"; 5)

повертає 

MID(текст; початок; довжина) – витягує частину рядка.

Наприклад, формула

=MID("Статистика споживання готової продукції"; 6; 3)

Повертає

5 MID(текст, початок, довжина) сти

LEN(текст) – визначає довжину тексту.

Наприклад, формула

=LEN("Статистика споживання готової продукції")

Повертає

6 LEN(текст) 39

TEXT(значення; "формат") – перетворює числа у відформатований текст.

У табл. 2.3 наведено варіанти роботи цієї функції. Особливістю цих функцій є неможливість задати посилання на вміст певної клітинки.

Треба текст наводити повністю прямо у функції.

Таблиця 2.3

Варіанти перетворення даних для функції TEXT

Формула	Опис
=TEXT(1234,567;"###0,00 ₴")	Грошовий формат із роздільником груп розрядів і двома розрядами дробової частини, наприклад 1234,57 ₴. Зверніть увагу, що Excel округлює значення до двох розрядів дробової частини.
=TEXT(TODAY();"DD.MM.YY")	Сьогоднішня дата у форматі ДД.ММ.РР, наприклад 14.03.12.
=TEXT(TODAY();"DDDD")	Сьогоднішній день тижня, наприклад понеділок.
=TEXT(NOW();"HH:MM")	Поточний час, наприклад 13:29.
=TEXT(0,285;"0,0%")	Відсотковий формат, наприклад 28,5%.

Формула	Опис
=TEXT(4,34 ,"# ?/?")	Дробовий формат, наприклад 13/3.
=TRIM(TEXT(0,34;"# ?/?"))	Дробовий формат, наприклад 1/3. Зверніть увагу, що функція TRIM використовується, щоб видалити пробіл перед дробовою частиною.
=TEXT(12200000;"0,00E+00")	Експоненційний формат, наприклад 1,22E+07.
=TEXT(1234567898;"[<=9999999]###-####;(###) ###-####")	Додатковий формат (номер телефону), наприклад (123) 456-7898.
=TEXT(1234;"0000000")	Додавання нулів на початку, наприклад 0001234.
=TEXT(123456;"##0° 00' 00'")	Спеціальний формат (широта або довгота

2.5. Робота з датами та часом

Excel має потужний набір функцій для роботи з датами:

TODAY() – повертає поточну дату.

NOW() – повертає поточну дату і час.

Наприклад

9	TODAY()	19.03.2025
10	NOW()	19.03.2025 11:12

DATE(рік; місяць; день) – створює дату на основі трьох параметрів.

Наприклад

11 | DATE(рік; місяць; день) | 14.08.1913 |

DATEDIF(start_date; end_date; "одиниця") – обчислює різницю між датами.

Параметри:

start_date: перша дата заданого періоду, **end_date**: остання дата заданого періоду;

одиниця – "Y" для повних років, "M" для повних місяців і "D" для повних днів. Явні дати та одиниці мають бути лапки у формулі.

Наприклад: =DATEDIF("24.10.23";E9;"D"), де D9 – дата початку, E9 – дата завершення, а "D" – це кількість повних днів між двома датами.

Або, для формули =DATEDIF(B9; "28.04.2025"; "M"), маємо результат

12 | DATEDIF(start_date; end_date; "одиниця") | 1 |

Якщо потрібна різниця просто у днях, в Excel діють прості оператори додавання і віднімання, наприклад, для формули =B9+15 маємо

19.03.2025 | 03.04.2025 |

або для двох дат 19.03.2025 03.04.2025 формула =B9+C9 дає результат у днях, що пройшли від 1 січні 1900 року – 91485.

EOMONTH(дата; зсув) – визначає останній день місяця з урахуванням зсуву.

Дата. Початкова дата. Дати слід вводити за допомогою функції DATE або як результати інших формул чи функцій. Наприклад, щоб ввести значення "23 травня 2008 р.", скористайтеся формулою DATE(2008;5;23). Наприклад, щоб ввести значення "23 травня 2008 р.", скористайтеся формулою DATE(2008;5;23). (Обов'язковий аргумент.)

Зсув. Кількість місяців до або після вказаного значення поч_дата. Додатне значення цього аргументу відповідає майбутній даті, а від'ємне – минулій (обов'язковий аргумент).

Формула	Опис	Результат
=EOMONTH(A2;1)	Дата останнього дня місяця через один місяць після дати, вказаної в A2.	28.2.2011
=EOMONTH(A2;-3)	Дата останнього дня місяця за три місяці до дати, вказаної в A2.	31.10.2010

2.6. Масиви та нові функції

Масиви – це набір значень, які повертаються формулою в кілька клітинок або працюють всередині однієї клітинки. Нові функції масивів Excel, представлені з появою динамічних масивів, автоматично заповнюють потрібну кількість клітинок, а не вимагають використання комбінації Ctrl+Shift+Enter, як раніше.

Масиви можуть містити довільну кількість рядків та стовпців, але у більшості випадків, комірки у масивах заповнюються однотипними даними.

Наприклад, ось можливі варіанти масивів

52	48	53
51	51	53
51	56	55
51	55	54
48	47	54

20.03.1964
21.03.1964
22.03.1964
23.03.1964
24.03.1964
25.03.1964
26.03.1964
27.03.1964
28.03.1964
29.03.1964
30.03.1964
31.03.1964
01.04.1964
02.04.1964

Особливості динамічних масивів:

1. **Динамічний діапазон:** Масиви автоматично розширюються або стискаються в залежності від обсягу результату.

2. Одночасні обчислення: Обробка кількох значень за допомогою однієї формули.

Для роботи з динамічними масивами були створені нові функції

2.7. Нові функції

2.7.1. Функція LET

Функція LET призначає імена результатам обчислення. Це дає змогу зберігати проміжні обчислення, значення або визначати імена всередині формули. Ці імена діють тільки в межах функції LET. Подібно до змінних у програмуванні, LET виконується за допомогою власного синтаксису формули Excel.

Щоб скористатися функцією LET в Excel, визначте пари імен і пов'язаних значень, а також обчислення, у якому всі вони використовуються. Визначте щонайменше одну пару "ім'я–значення" (змінну). Функція LET підтримує до 126 таких пар.

Синтаксис:

=LET(ім'я1, значення1, ім'я2, значення2, ..., результат)

Переваги LET:

- Зменшує кількість повторів у формулі.
- Робить формули більш структурованими й легшими для розуміння.

Приклади:

1) Знайдення ділення а на b. Спочатку присвоюється значення змінним а і b, а потім виконується дія ділення.

=LET(a;10; b;20; a/b)

2) Обчислення об'єму куба за довжиною його ребра а. Спочатку визначається числове значення довжини ребра, потім змінній Vol присвоюється куб числа а і результат цього розрахунку показується в комірці.

=LET(a; 5; Vol; a^3; Vol)

Результат: 125.

З останнього прикладу видно, що змінним можна присвоювати не тільки числа, а і результати розрахунків у вигляді формул і функцій. Причому, можливі значення змінних у формулі можуть братися за посилання на адреси комірок.

3) Визначення різниці чисел, що містяться у різних комірках

=LET(a; B9; b;C9; a - b)

Результат: -15.

Отже, за допомогою функції LET складну формулу можна розбити на окремі елементи.

4) Розрахувати значення формули $\varepsilon_D = \ddot{D}_x \Phi^{-1}(\beta) \sqrt{\frac{0,8N+1,2}{N(N-1)}}$.

Розіб'ємо цю формулу на окремі частини, розрахувавши спочатку чисельник під радикалом, потім знаменник, потім витягнемо корінь квадратний і наостанок помножимо результат на добуток двох чисел $\ddot{D}_x \Phi^{-1}(\beta)$. Ці два числа розташовані на іншому аркуші Excel і мають значення відповідно 325,0538417 та 0,7.

=LET(N; 20; C;0,8*N+1;D;N*(N-1);X;(C/D)^0,5;Аркуш1!D119*Аркуш1!B125*X)

Результат: 48,126691

2.7.2. Функція LAMBDA

Функція LAMBDA дозволяє створювати власні користувацькі функції прямо у книзі Excel, без використання VBA або макросів.

Функція LAMBDA дає змогу створювати спеціальні функції для формул, які часто використовуються. Функція має один або кілька параметрів до 253 параметрів і формулу для обчислення.

=LAMBDA([параметр1;параметр2;...;] обчислення)

параметр: значення або значення, які потрібно передати функції.

обчислення: формула для виконання з використанням параметрів.

Обчислення має бути останнім аргументом і повертати результат.

Синтаксис:

=LAMBDA(параметр1, параметр2, ..., обчислення)(значення1, значення2, ...)

Приклад:

Функція для обчислення квадрата числа:

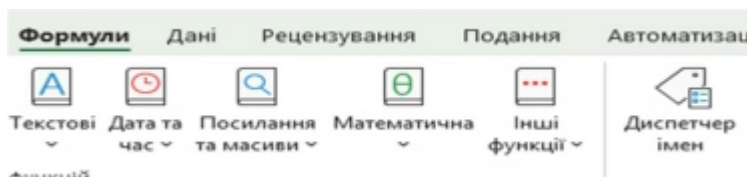
=LAMBDA(x, x^2)(5)

Результат: 25.

Можна також зберегти функцію LAMBDA через "Диспетчер імен", щоб використовувати її багаторазово в інших клітинках.

Приклад. Переведення значення температури зі шкали Фаренгейта у шкалу Цельсія.

Скористаємося диспетчером імен. Для цього На вкладці **Формули** натисніть кнопку **Диспетчер імен**. Або натисніть клавіші Ctrl+F3.

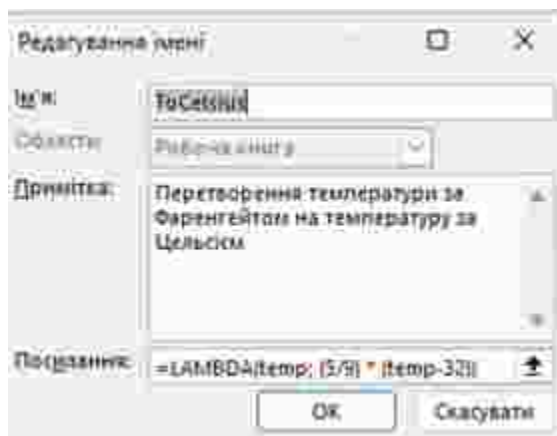


Треба натиснути кнопку «Створити» і заповнити вікно з даними про нову функцію.

Ім'я: ToCelsius

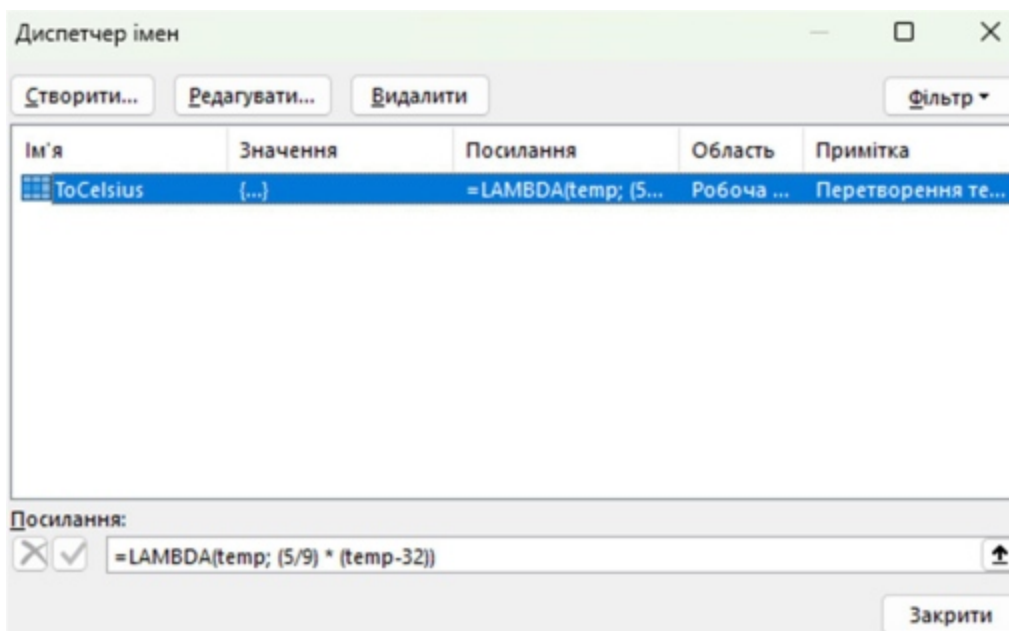
Область: Робоча книга

Примітка: Перетворення температури за Фаренгейтом на температуру за Цельсієм



Посилання: =LAMBDA(temp; (5/9) * (temp-32))

Ось підтвердження, що нове ім'я функції створено.



Тепер у будь якій клітинці робочої книги можна скористатися цією функцією, наприклад отак =ToCelsius(75) .

Результат: 23,88888889.

Існує можливість для створення функцій з декількома аргументами, наприклад =LAMBDA(a;b; a/b). Тоді формула =apb(12;10) дає результат 1,2.

2.7.3. Функція SEQUENCE

Функція SEQUENCE створює масив чисел на основі заданих параметрів.

Синтаксис:

=SEQUENCE(рядки, стовпці, початок, крок)

Аргумент	Опис
рядки	Кількість рядків, які потрібно повернути.
Обов'язковий	
[стовпці]	Кількість стовпців, які потрібно повернути.
Необов'язковий	
[початок]	Перше число в послідовності.
Необов'язковий	
[крок]	Крок приросту кожного наступного значення в масиві.

Аргумент	Опис
----------	------

Необов'язковий

Основні правила функції:

- Будь-які відсутні необов'язкові аргументи за замовчуванням дорівнюють 1. Якщо аргумент рядків не вказано, потрібно вказати принаймні один інший аргумент.
- Масив можна розглядати як рядок або стовпець значень чи поєднання рядків і стовпців значень. У наведеному вище прикладі діапазон C1:G4 – це масив формули SEQUENCE.
- Функція SEQUENCE повертає масив, який розгорнеться, якщо це остаточний результат формули. Це означає, що якщо натиснути клавішу ENTER, Excel динамічно створить діапазон масиву відповідних розмірів. Якщо допоміжні дані розташовано в таблиці Excel і ви використовуєте структуровані посилання, розмір масиву автоматично змінюватиметься після додавання даних до його діапазону або їх вилучення.
- Excel має обмежену підтримку перенесення динамічних масивів між книгами, і цей сценарій підтримується, лише якщо відкрито обидві книги. Якщо закрити вихідну книгу, формули пов'язаного динамічного масиву повертатимуть помилку [#REF!](#) під час оновлення.

Застосування:

- Створення таблиць із заданими шаблонами чисел.
- Динамічна генерація даних для аналізу.

Приклади:

1) Створення масиву з чисел від 1 до 10:

excel

=SEQUENCE(10, 1, 1, 1)

Результат: вертикальний стовпець із чисел 1, 2, 3... 10. Перше число стовця стоїть у комірці, де вказано формула функції.

У наведеному нижче прикладі ми за допомогою функції
=SEQUENCE(4;5) створили масив із 4 рядками та 5 стовпцями.

Formula bar: `=SEQUENCE(4;5)`

C	D	E	F	G
1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

2) Нижче наведено приклад функції SEQUENCE з аргументами [TEXT](#), [DATE](#), [YEAR](#) та [TODAY](#), за допомогою якої можна швидко створити набір даних рядка заголовка з динамічним списком місяців, де основна дата завжди відповідатиме поточному року. Ось яка потрібна формула: `=TEXT(DATE(YEAR(TODAY());SEQUENCE(1;6);1),"mmm")`.

Formula bar: `=TEXT(DATE(YEAR(TODAY());SEQUENCE(1;6);1),"mmm")`

C	D	E	F	G	H	I
Код ГК	Січ	Лют	Бер	Кві	Тра	Чер
1001	65	145	225	305	385	465
2001	545	625	705	785	865	945
3001	1025	1105	1185	1265	1345	1425
4001	1505	1585	1665	1745	1825	1905
5001	1985	2065	2145	2225	2305	2385

3) Нижче наведено приклад функції SEQUENCE із вкладеннями [INT](#) і [RAND](#). За її допомогою можна створити масив із 5 рядками по 6 стовпців і випадковим набором цілих чисел, які збільшуються. Ось яка потрібна формула: `=SEQUENCE(5;6;INT(RAND()*100);INT(RAND()*100))`.

Formula bar: `=SEQUENCE(5;6;INT(RAND()*100);INT(RAND()*100))`

Код ГК	Січ	Лют	Бер	Кві	Тра	Чер
1001	18	44	70	96	122	148
2001	174	200	226	252	278	304
3001	330	356	382	408	434	460
4001	486	512	538	564	590	616
5001	642	668	694	720	746	772

Крім того, за допомогою функції `=SEQUENCE(5;1;1001;1000)` можна створити послідовний список кодів головної книги з прикладів.

2.7.4. Функція UNIQUE(масив) – повертає унікальні значення з діапазону.

Синтаксис

Аргумент	Опис
масив Обов'язковий	Діапазон або масив, з якого необхідно повернути унікальні рядки або стовпці
[за_стовпцем] Необов'язковий	Аргумент <code>by_col</code> – це логічне значення, яке вказує на спосіб порівняння. TRUE – стовпці порівнюватимуться один з одним, і повертатимуться унікальні стовпці FALSE (або пропущено) – рядки порівнюватимуться один з одним, і повертатимуться унікальні рядки
[exactly_once] Необов'язковий	Аргумент <code>exactly_once</code> – це логічне значення, яке повертатиме рядки або стовпці, що з'являються лише раз у діапазоні або масиві. Це концепція унікальності в базі даних. TRUE – повертатимуться всі унікальні рядки або стовпці з діапазону або масиву, які з'являються лише раз FALSE (або пропущено) – повертатимуться всі унікальні рядки або стовпці з діапазону або масиву

- Масив можна розглядати як рядок чи стовпець значень або поєднання рядків і стовпців значень. У наведених вище прикладах у формулах **UNIQUE** використовуються діапазони масивів **D2:D11** і **D2:D17** відповідно.

- Функція **UNIQUE** повертає масив, який розгорнеться, якщо це остаточний результат формули. Це означає, що якщо натиснути клавішу **ENTER**, Excel динамічно створить діапазон масиву відповідних розмірів. Якщо допоміжні дані розташовано в [таблиці Excel](#) і ви використовуєте [структуровані посилання](#), розмір масиву автоматично змінюватиметься після додавання даних

до його діапазону або їх вилучення. Докладні відомості надано в цій статті про [поведінку розгорнутого масиву](#).

- Excel має обмежену підтримку перенесення динамічних масивів між книгами, і цей сценарій підтримується, лише якщо відкрито **обидві** книги. Якщо закрити вихідну книгу, формули пов'язаного динамічного масиву повертатимуть помилку **#REF!** під час оновлення.

Повертає унікальні записи зі списку значень. Повертає унікальні записи зі списку імен

The screenshot shows the Excel formula bar with the formula `=UNIQUE(B2:B11)`. Below it is a table with columns A, B, C, D, and E. Column B contains the values 1, 1, 2, 2, 3, 3, 4, 4, 5, 5. Column D shows the result of the UNIQUE function, which are the unique values: 1, 2, 3, 4, 5.

	A	B	C	D	E
1					
2		1		1	
3		1		2	
4		2		3	
5		2		4	
6		3		5	
7		3			
8		4			
9		4			
10		5			
11		5			

Приклад 1

У цьому прикладі функції SORT і UNIQUE використовуються разом, щоб повернути унікальний список

The screenshot shows the Excel formula bar with the formula `=UNIQUE(B2:B17)`. Below it is a table with columns A, B, C, D, and E. Column B contains a list of names: Tom, Fred, Eri, Sal, Eric, "Kirk", Kai, Hector, Tom, Fred, Eri, Sal, Eric, "Kirk", Kai, Hector. Column D shows the result of the UNIQUE function, which are the unique names: Tom, Fred, Eri, Sal, Eric, "Kirk", Kai, Hector.

	A	B	C	D	E
1					
2		Tom		Tom	
3		Fred		Fred	
4		Eri		Eri	
5		Sal		Sal	
6		Eric		Eric	
7		"Kirk"		"Kirk"	
8		Kai		Kai	
9		Hector		Hector	
10		Tom			
11		Fred			
12		Eri			
13		Sal			
14		Eric			
15		"Kirk"			
16		Kai			
17		Hector			

Приклад 2

У цьому прикладі для аргументу `exactly_once` задано значення `TRUE` і функція повертає тільки тих клієнтів, які отримали обслуговування один раз. Це може допомогти визначити людей, які не зверталися за додатковим обслуговуванням, щоб зв'язатися з ними.

імен за зростанням.

Formula bar: `=SORT(UNIQUE(B2:B17))`

	A	B	C	D	E
1		Торговий представник		Відсортовані	
2		Том		Емі	
3		Фред		Фред	
4		Емі		Фріц	
5		Сал		Гектор	
6		Фріц		Сал	
7		"Крик"		"Крик"	
8		Ксі		Том	
9		Гектор		Ксі	
10		Том			
11		Фред			
12		Емі			
13		Сал			
14		Фріц			
15		"Крик"			
16		Ксі			
17		Гектор			
18					

Formula bar: `=UNIQUE(A2:A12, TRUE)`

	A	B	C
1	Ім'я клієнта	Тип послуги	Дата
2	Fife, Грант	Обертання шини	01/01/2018
3	Прюїтт, Барбара	Заміна масла	03/05/2018
4	Релі, Френсіс	30-миль Service 30K	03/08/2018
5	Барретт, Алісія	40-миль Service 40K	03/10/2018
6	Барретт, Алісія	Заміна гальмівних колодок	03/15/2018
7	Парсон, Лінн	колодки для заміни масла	04/13/2018
8	Прюїтт, Барбара	Заміна масла	05/06/2018
9	Ентоні С.	на нафту	05/14/2018
10	Ентоні Спок	Перевірка датчиків	05/25/2018
11	Релі, Френсіс	Гальмівні колодки	06/29/2018
12	Браун, благодійний	Обертання шини	09/29/2018
13			
14	Відсортовані імена клієнтів		
15	Fife, Грант		
16	Парсон, Лінн		
17	Браун, благодійний		
18			

Приклад 3

У цьому прикладі використовується амперсанд (&), щоб об'єднати прізвище та ім'я в повне ім'я. Зверніть увагу, що формула посилається на весь діапазон імен в A2:A12 і B2:B12. Це дозволяє Excel повертати масив усіх імен.

Formula bar: `=UNIQUE(B2:B12&" "&A2:A12)`

	A	B	C	D
1	Прізвище	Ім'я		Повне ім'я
2	Fife	Грант		Грант Fife
3	Прюїтт	Барбара		Барбара Прюїтт
4	Хорн	Френсіс		Френсіс Хорн
5	Барретт	Алісія		Алісія Барретт
6	Барретт	Алісія		Лінн Парсон
7	Парсон	Лінн		Ентоні Спок
8	Прюїтт	Барбара		Благодійний Браун
9	Немає	Ентоні		
10	Немає	"Алюміній"		
11	Хорн	Френсіс		
12	Коринний	Шаріту		
13				

Приклад 4. У цьому прикладі порівнюються два стовпці і повертаються лише унікальні значення між ними.

	A	B	C	D	E	F	G	H
1	Північ	Південь	Північ	Південь	Північ		Північ	Південь
2	Північ	Південь	Північ	Південь	Північ		Північ	Південь
3	Північ	Південь	Північ	Південь	Північ			
4	Північ	Південь	Північ	Південь	Північ			
5	Північ	Південь	Північ	Південь	Північ			
6	Північ	Південь	Північ	Південь	Північ			
7	Північ	Південь	Північ	Південь	Північ			
8	Північ	Південь	Північ	Південь	Північ			
9	Північ	Південь	Північ	Південь	Північ			
10	Північ	Південь	Північ	Південь	Північ			
11	Північ	Південь	Північ	Південь	Північ			
12	Північ	Південь	Північ	Південь	Північ			
13	Північ	Південь	Північ	Південь	Північ			

2.7.5. Функція SORT(масив, стовпець, порядок) – сортує дані за вказаним критерієм.

Синтаксис

Аргумент	Опис
масив Обов'язковий	Масив або діапазон, який потрібно відсортувати.
[індекс_сортування] Необов'язковий	Число, яке вказує рядок або стовпець, за яким потрібно відсортувати дані.
порядок_сортування Необов'язковий	Число, яке вказує потрібний порядок сортування: 1 – за зростанням (стандартне значення), -1 – за спаданням.
[за_стовпцем] Необов'язковий	Логічне значення, яке вказує потрібний напрямок сортування: FALSE – за рядком (стандартне значення), TRUE – за стовпцем.

- Якщо аргумент "індекс_сортування" не вказано, дані сортуються за рядком 1 або за стовпцем 1. Якщо не вказано порядок, дані сортуються за зростанням. За замовчуванням Excel сортує дані за рядком. Сортування за

стовпцем застосовується, лише якщо аргумент "за_стовпцем" має значення TRUE. Якщо аргумент "за_стовпцем" має значення FALSE або відсутній, Excel сортує дані за рядком.

- Функція SORT дає змогу сортувати дані в масиві. Якщо потрібно відсортувати дані в сітці, радимо використовувати гнучкішу [функцію SORTBY](#). Функція SORTBY посилається на діапазон, тому враховує додані та видалені стовпці, а функція SORT посилається на номер індексу стовпця.

- Масив можна розглядати як рядок або стовець значень чи поєднання рядків і стовпців значень. У наведеному вище прикладі діапазон A5:D20 – це вихідний масив формули SORT.

- Функція SORT повертає масив, який розгорнеться, якщо це остаточний результат формули. Це означає, що якщо натиснути клавішу **ENTER**, Excel динамічно створить діапазон масиву відповідних розмірів. Якщо допоміжні дані розташовано в таблиці Excel і ви використовуєте структуровані посилання, розмір масиву автоматично змінюватиметься після додавання даних до його діапазону або їх вилучення.

- Excel має обмежену підтримку перенесення динамічних масивів між книгами, і цей сценарій підтримується, лише якщо відкрито **обидві** книги. Якщо закрити вихідну книгу, формули пов'язаного динамічного масиву повертатимуть помилку [#REF!](#) під час оновлення.

Приклади

У цьому прикладі ми за допомогою функції =SORT(A2:A17) відсортуємо дані за регіоном, торговим представником і продуктом. Значення, за якими потрібно відсортувати дані, отримано з клітинок F2, H2 та J2 відповідно.

включити Обов'язковий	Масив логічних значень, висота або ширина якого відповідає масиву, який потрібно відфільтрувати
[якщо_пустий] Необов'язковий	Значення, що повертається, якщо всі значення включеного масиву пусті (фільтр нічого не повертає)

- Масив можна розглядати як рядок або стовпець значень чи поєднання рядків і стовпців значень. У наведеному вище прикладі діапазон A5:D20 – це вихідний масив формули FILTER.

- Функція FILTER повертає масив, який розгорнеться, якщо це остаточний результат формули. Це означає, що якщо натиснути клавішу ENTER, Excel динамічно створить діапазон масиву відповідних розмірів. Якщо допоміжні дані розташовано в таблиці Excel і ви використовуєте структуровані посилання, розмір масиву автоматично змінюватиметься після додавання даних до його діапазону або їх вилучення.

- Якщо існує ймовірність того, що за вашим набором даних функція поверне пусте значення, додайте третій аргумент ([якщо_пустий]). Інакше ви отримаєте помилку #CALC!, оскільки Excel поки не підтримує пусті масиви.

- Якщо будь-яке значення аргументу include є помилкою (#N/A, #VALUE тощо) або не може бути перетворено на логічне значення, функція FILTER поверне помилку.

- Excel має обмежену підтримку перенесення динамічних масивів між книгами, і цей сценарій підтримується, лише якщо відкрито обидві книги. Якщо закрити вихідну книгу, формули пов'язаного динамічного масиву повертатимуть помилку #REF! під час оновлення.

Приклади

1) У наведеному нижче прикладі ми використали формулу =FILTER(A5:D20;C5:C20=H2;""), щоб повернути всі записи для Apple, як вибрано у клітинці H2, а якщо яблук немає, поверніть пустий рядок ("").

F5		=FILTER(A5:D20;C5:C20=H2)							
	A	B	C	D	E	F	G	H	I
1									
2	Операції збуту						Продукт:	Яблуко	
3									
4	Регіон	Торговий	Продукт	Одиниці	Регіон	Торговий	Продукт	Одиниці	
5	Схід	Андрій	Яблуко	6380	Схід	Андрій	Яблуко	6380	
6	Захід	Павло	Виноград	5619	Схід	Богдан	Яблуко	4394	
7	Північ	Яна	Груша	4565	Південь	Олександр	Яблуко	1310	
8	Південь	Олександр	Банани	5323	Південь	Микола	Яблуко	9814	
9	Схід	Богдан	Яблуко	4394					
10	Захід	Дмитро	Виноград	7195					
11	Північ	Кирило	Груша	5231					
12	Південь	Микола	Банани	2427					
13	Схід	Андрій	Банани	4213					
14	Захід	Павло	Груша	3239					
15	Північ	Яна	Виноград	6420					
16	Південь	Олександр	Яблуко	1310					
17	Схід	Богдан	Банани	6274					
18	Захід	Дмитро	Груша	4894					
19	Північ	Кирило	Виноград	7580					
20	Південь	Микола	Яблуко	9814					

2) Використання функції FILTER для повернення множинних умов

У цьому прикладі ми скористаємось оператором множення (*), щоб повернути всі значення в діапазоні масиву (A5:D20) для товару "Яблука" й регіону "Східний": =FILTER(A5:D20;(C5:C20=H1)*(A5:A20=H2);"").

fx		=FILTER(A5:D20;(C5:C20=H1)*(A5:A20=H2);"")			
E	F	G	H	I	J
		Продукт:	Яблуко		
		Регіон:	Схід		
	Регіон	Торговий	Продукт	Одиниці	
	Схід	Андрій	Яблуко	6380	
	Схід	Богдан	Яблуко	4394	

3) Використання функції FILTER для повернення множинних умов сортування

У цьому прикладі ми скористаємось попередньою функцією FILTER з функцією SORT, щоб повернути всі значення в діапазоні масиву (A5:D20) для

спаданням: =SORT(FILTER(A5:D20;(C5:C20=H1)*(A5:A20=H2);""),4,-1)

4) У цьому прикладі ми скористаємось функцією FILTER з додатковим оператором (+), щоб повернути всі значення в діапазоні масиву (A5:D20) для товару "Яблука" АБО регіону "Східний", а потім відсортувати значення стовпця "Кількість" за

спаданням: =SORT(FILTER(A5:D20;(C5:C20=H1)+(A5:A20=H2);""),4;-1).

Зверніть увагу: усі функції можна використовувати без абсолютних посилань, оскільки вони зберігаються лише в одній клітинці та розгортають свої результати в суміжних клітинках.

2.7.7. Функція XLOOKUP – пошук даних у таблиці за ключем

Скористайтесь функцією XLOOKUP , щоб знайти об'єктів у таблиці або діапазоні за рядком. Наприклад, знайдіть ціну автомобільної частини на номер деталі або знайдіть ім'я працівника на основі його ідентифікатора працівника. Функція XLOOKUP дає змогу шукати пошуковий запит в одному стовпці в одному стовпці та повертати результат із того самого рядка в іншому стовпці, незалежно від того, з якого боку ввімкнуто зворотний стовпець.

Синтаксис

Функція XLOOKUP шукає діапазон або масив, а потім повертає елемент, який відповідає першому збігу, який він знаходить. Якщо збігів немає, функція XLOOKUP може повернути найближчий (приблизний) збіг.

=XLOOKUP(lookup_value;lookup_array;return_array;if_not_found;[match_mode];[search_mode])

Аргумент	Опис
lookup_value Необхідний*	Значення для пошуку *Якщо його не вказано, функція XLOOKUP повертає пусті клітинки, знайдені в lookup_array.
lookup_array Обов'язковий	Масив або діапазон, які потрібно знайти
return_array Обов'язковий	Масив або діапазон, який потрібно повернути
[if_not_found] Необов'язковий	Якщо не знайдено припустимий збіг, поверніть введений текст [if_not_found]. Якщо не знайдено припустимий збіг і [if_not_found] відсутній, повертається #N/A .
[match_mode] Необов'язковий	Укажіть тип збігу:

Аргумент	Опис
	0 - Точний збіг. Якщо не знайдено, поверніть #N/A. Цей параметр встановлено за промовчанням. -1 - Точний збіг. Якщо нічого не знайдено, поверніть наступний менший елемент. 1 - Точний збіг. Якщо не знайдено, поверніть наступний більший елемент. 2 – збіг символу підстановки, де *, ? та ~ мають особливе значення: ? (знак питання) – Будь-який символ. Наприклад, sm?th знаходить слова "коваль" і "смит" * (зірочка) – Будь-яка кількість символів. Наприклад, *схід знаходить "Північний схід" і "Південний схід" ~ (тильда) зі знаком ?, * або ~ в кінці. Наприклад, фр06~? Наприклад, умові «ФР06~?» відповідає результат «ФР06?».
[search_mode] Необов'язковий	Укажіть режим пошуку для використання: 1. Виконайте пошук, починаючи з першого елемента. Цей параметр встановлено за промовчанням. -1 . Виконайте зворотний пошук, починаючи з останнього елемента. 2. Виконайте двійковий пошук, який залежить від lookup_array сортування <i>за зростанням</i> . Якщо його невідсортовано, буде повернено недійсні результати. 2 – здійснити бінарний пошук, який покладається на те, що масив lookup_array посортовано <i>за спаданням</i>. Якщо його невідсортовано, буде повернено недійсні результати.

Приклади

У прикладі 1 використовується функція XLOOKUP для пошуку назви країни в діапазоні, а потім повертається код країни телефону. Вона містить

аргументи `lookup_value` (клітинка F2), `lookup_array` (діапазон B2:B11) і `return_array` (діапазон D2:D11). Він не містить аргумент `match_mode`, оскільки функція XLOOKUP за замовчуванням видає точний збіг.

	A	B	C	D	E	F	G
1		Країна	Лінійчат	Префікс			
2		Китайська	CN	+86			
3		(Індія)	ДЮЙМ	+91			
4		Сполучені Штати США		+1			
5		Індонезія	ID	+62			
6		Бразилія	BR	+55			
7		Пакистан	PK	+92			
8		Нігерія	NG	+234			
9		Бангладеш	BD	+880			
10		Росія	RU	+7			
11		Мексика	MX	+52			

Функція XLOOKUP використовує масив підстановки та масив, що повертається, тоді як функція VLOOKUP використовує масив однієї таблиці та номер індексу стовпця. Еквівалентна формула VLOOKUP у цьому випадку матиме такий вигляд: `=VLOOKUP(F2;B2:D11;3;FALSE)`.

Приклад 2 підбирає відомості про працівників на основі ідентифікаційного номера працівника. На відміну від функції VLOOKUP, функція XLOOKUP може повертати масив із кількох елементами, тому одна формула може повертати як ім'я працівника, так і відділ із клітинок C5:D14.

Formula bar: `=XLOOKUP(B2:B5:B14,C5:D14)`

	A	B	C	D
1		Ідентифікатор працівника	Ім'я працівника	Відділ
2		11890	Тамара Бабич	Фінансовий відділ
3				
4		Ідентифікатор працівника	Ім'я працівника	Відділ
5		4390	Володимир Попович	Відділ маркетингу
6		8604	Ольга Тарасюк	Відділ збуту
7		8389	Тамара Бабич	Фінансовий відділ
8		4937	Вікторія Шуменко	Бухгалтерія
9		8299	Юлія Гнатенко	Операційний відділ
10		2643	Богдан Козар	Керівник
11		5243	Оксана Бойко	Відділ збуту
12		9693	Марина Шевченко	Фінансовий відділ
13		1636	Яна Петренко	Бухгалтерія
14		6703	Аліна Олексієнко	Відділ маркетингу

Приклад 3 додає аргумент if_not_found до попереднього прикладу.

Formula bar: `=XLOOKUP(B2:B5:B14,C5:D14,"Ідентифікатор не знайдено")`

	A	B	C	D
1		Ідентифікатор працівника	Ім'я працівника	Відділ
2		1234	Ідентифікатор не знайдено	
3				
4		Ідентифікатор працівника	Ім'я працівника	Відділ
5		4390	Володимир Попович	Відділ маркетингу
6		8604	Ольга Тарасюк	Відділ збуту
7		8389	Тамара Бабич	Фінансовий відділ
8		4937	Вікторія Шуменко	Бухгалтерія
9		8299	Юлія Гнатенко	Операційний відділ
10		2643	Богдан Козар	Керівник
11		5243	Оксана Бойко	Відділ збуту
12		9693	Марина Шевченко	Фінансовий відділ
13		1636	Яна Петренко	Бухгалтерія
14		6703	Аліна Олексієнко	Відділ маркетингу

Приклад 4 шукає в стовпці C особистий дохід, введений у клітинці E2, і знаходить відповідну ставку податку у стовпці B. Задає аргумент if_not_found повертати 0 (нуль), якщо нічого не знайдено. Для аргументу match_mode встановлено значення 1, що означає, що функція шукатиме точний збіг, а якщо не вдасться його знайти, вона повертає наступний більший елемент. Нарешті, аргумент search_mode має значення 1, що означає, що функція здійснюватиме пошук від першого елемента до останнього.

	Податкова ставка	Максимальний дохід
2	10%	9 700 €
3	22%	39 475 €
4	24%	84 200 €
5	32%	160 726 €
6	35%	204 100 €
7	37%	510 300 €

	Дохід	Податкова ставка
2	46 523 €	24%

Стовпець lookup_array XARRAY розташовано праворуч від стовпця return_array , тоді як функція VLOOKUP може виглядати лише зліва направо.

Приклад 5 використовує вкладену функцію XLOOKUP для виконання вертикального та горизонтального збігів. Спочатку він шукає валовий прибуток у стовпці B, а потім шукає Qtr1 у верхньому рядку таблиці (діапазон C5:F5), а потім повертає значення на перетині двох. Це схоже на спільне використання функцій [INDEX](#) і [MATCH](#) .

Тір: Функцію HLOOKUP також можна замінити за допомогою функції [XLOOKUP](#) .

	Quarter	Gross Profit	Net Profit	Profit
Qtr1	\$15,000	\$19,342	29.3%	

	Qtr1	Qtr2	Qtr3	Qtr4
Загальний обсяг продажів	\$50,000	\$78,200	\$89,500	\$91,250
Вартість збуту	(\$25,000)	(\$42,000)	(\$23,400)	(\$40,400)
Валовий прибуток	\$25,000	\$36,190	\$66,050	\$50,850
Амортизація	(\$100)	(\$75)	(\$200)	(\$12)
Інфляція	(\$11)	(\$33)	(\$150)	(\$50)
Заробіток перед оподаткуванням	\$23,589	\$34,506	\$65,650	\$49,732
Податок	(\$4,346)	(\$6,211)	(\$13,340)	(\$1,290)
Чистий прибуток	\$19,342	\$28,295	\$52,310	\$48,442
Прибуток	29.3%	27.8%	23.4%	27.6%

Note: Формула у клітинках D3:F3 така: =XLOOKUP(D2;\$B 6:\$B 17;XLOOKUP(\$C 3;\$C 5:\$G 5;\$C 6:\$G 17)).

Приклад 6 використовує функцію SUM і дві вкладені функції XLOOKUP, щоб підсумувати всі значення між двома діапазонами. У цьому випадку ми хочемо підсумувати значення винограду, бананів і включити груші, які знаходяться між ними.

	A	B	C	D	E	F	G	H
1								
2		Почати	Завершення	Підсумок				
3		Винограду	Банан	\$110,70				
4								
5		Продукт	Qty	Ціна	Підсумок			
6		Яблуко	23	\$0.52	\$11.90			
7		винограду	98	\$0.77	\$75.28			
8		груша	75	\$0.24	\$18.16			
9		Банан	95	\$0.18	\$17.25			
10		Вишня	42	\$0.16	\$6.80			

Формула в клітинці E3 така:
`=SUM(XLOOKUP(B3;B6:B10;E6:E10):XLOOKUP(C3;B6:B10;E6:E10))`

Принцип роботи Функція XLOOKUP повертає діапазон, тому, коли вона обчислюється, формула закінчується так: `=SUM($E$7:$E$9)`. Ви можете побачити, як це працює самостійно, вибравши клітинку з формулою XLOOKUP, подібною до цієї, а потім виберіть Пункт Формули > Аудит формули > [Обчислити формулу](#), а потім натисніть кнопку Обчислити, щоб виконати обчислення.

2.7.8. SUMIF – додавання за умови

Функція SUMIF додає діапазон значень, які відповідають певним умовам. Наприклад, щоб підсумувати лише значення, більші за 5, використовуйте формулу `=SUMIF(B2:B25;">5")`. Докладні відомості див. в [статті Функція SUMIF](#).

`SUMIF(діапазон; умова;[sum_range])`

Діапазон – це список значень, які потрібно підсумувати, умова – це значення клітинок, які потрібно додати, а sum_range (необов'язково) – це

список клітинок, які фактично підсумовуються, якщо вони відмінні від діапазону.

інтаксис функції **SUMIF** має такі аргументи:

- **Діапазон** — обов'язковий аргумент. Діапазон клітинок, які потрібно обчислити за умовами. Клітинки в кожному діапазоні мають бути числами, іменами, масивами або посиланнями, які містять числа. Пусті та текстові значення ігноруються. Вибраний діапазон може містити дати в стандартному форматі Excel (наприклад, нижче).
- **Умови.** Обов'язковий параметр. Умови у формі числа, виразу, посилання на клітинку, тексту або функції, що визначають, які клітинки потрібно підсумувати. Можна додати символи узагальнення – знак питання (?), який відповідає будь-якому окремому символу, зірочку (*) відповідно до будь-якої послідовності символів. Якщо потрібно знайти власне знак питання або зірочку, введіть тильду (~) перед символом. Наприклад, умови можна виразити як 32, ">32", B5, "3?", "яблуко*", "*~?", або TODAY().

Important: Текстову умову або умову, яка містить логічні або математичні символи, потрібно забрати в подвійні лапки ("). Якщо умова числова, подвійні лапки не обов'язкові.

- **Діапазон_для_суми.** Необов'язковий параметр. Фактичні клітинки, які потрібно підсумувати, якщо вони відрізняються від указаних в аргументі *діапазон*. Якщо аргумент *діапазон_для_суми* не вказано, у програмі Excel підсумовуються клітинки з аргументу *діапазон* (тобто ті самі клітинки, до яких застосовуються умови).

Sum_range має бути такого самого розміру та фігури, як і *діапазон*. Якщо це не так, продуктивність може потерпати, і формула підсумує діапазон клітинок, який починається з першої клітинки в **sum_range** але має ті самі розміри, що й *діапазон*. Наприклад:

<i>діапазон</i>	<i>Діапазон_для_суми.</i>	Фактичні підсумовані клітинки
A1:A5	B1:B5	B1:B5
A1:A5	B1:K5	B1:B5

Приклади

Приклад 1

Скопіюйте дані прикладу з наведеної нижче таблиці та вставте їх у клітинку A1 нового аркуша Excel. Щоб відобразити результат обчислення формул, виберіть їх, натисніть клавішу F2, а потім – клавішу Enter. За потреби можна змінити ширину стовпців, щоб відобразити всі дані.

Вартість майна	Комісія	Дані
100 000 грн.	7 000 грн.	250 000 грн.
200 000 грн.	14 000 грн.	
300 000 грн.	21 000 грн.	
400 000 грн.	28 000 грн.	
Формула	Опис	Результат
=SUMIF(A2:A5;">160000";B2:B5)	Сума комісії за майно вартістю більше 160 000 грн.	63 000 грн.
=SUMIF(A2:A5;">160000")	Сума вартості майна більше 160 000 грн.	900 000 грн.
=SUMIF(A2:A5;300000;B2:B5)	Сума комісії за майно, вартість якого дорівнює 300 000 грн.	21 000 грн.
=SUMIF(A2:A5;">" & C2;B2:B5)	Сума комісії за майно вартістю більше, ніж значення у клітинці C2.	49 000 грн.

Приклад 2

Скопіюйте дані прикладу з наведеної нижче таблиці та вставте їх у клітинку A1 нового аркуша Excel. Щоб відобразити результат обчислення

формул, виберіть їх, натисніть клавішу F2, а потім – клавішу Enter. За потреби можна змінити ширину стовпців, щоб відобразити всі дані.

Категорія	Продукти	Продаж
Овочі	Помідори	2 300 грн.
Овочі	Селера	5 500 грн.
Фрукти	Апельсини	800 грн.
	Масло	400 грн.
Овочі	Морква	4 200 грн.
Фрукти	Яблука	1 200 грн.
Формула	Опис	Результат
=SUMIF(A2:A7;"Фрукти";C2:C7)	Сума продажу всіх продуктів у категорії «Фрукти».	2 000 грн.
=SUMIF(A2:A7;"Овочі";C2:C7)	Сума продажу всіх продуктів у категорії «Овочі».	12 000 грн.
=SUMIF(B2:B7;"*и";C2:C7)	Сума продажу всіх продуктів, які закінчуються на «и» (помідори та апельсини).	4 300 грн.
=SUMIF(A2:A7;"";C2:C7)	Сума продажу всіх продуктів без указаної категорії.	400 грн.

Приклад 3. Загальні значення в клітинках A1 і A4, якщо вони більші за 100:

=SUMIF(A1:A4;">100")

Приклад 4. Загальна кількість значень у клітинках B1 і B4, якщо текст, починаючи з Y, міститься в стовпці A:

=SUMIF(A1:A4;"Y*";B1:B4)

2.7.9. Функція INDIRECT – пошук за посиланням

Повертає посилання, указане текстовим рядком. Посилання негайно обробляються для відображення їхнього вмісту. Функція INDIRECT використовується для змінення посилання на клітинку в межах формули без змінення самої формули.

Синтаксис

INDIRECT(посилання_на_клітинку;[a1])

Синтаксис функції INDIRECT має такі аргументи:

- Ref_text — обов'язковий аргумент. Посилання на клітинку, яка містить посилання у стилі A1, посилання у стилі R1C1, ім'я, визначене як посилання, або посилання на клітинку як текстовий рядок. Якщо ref_text не є припустимим посиланням на клітинку, функція INDIRECT повертає #REF! .
- Якщо аргумент "посилання_на_клітинку" посилається на іншу книгу (зовнішнє посилання), потрібно відкрити цю книгу. Якщо вихідну книгу не відкрито, функція INDIRECT повертає помилку #REF!. Примітка.
- Якщо аргумент "посилання_на_клітинку" посилається на клітинку за межами діапазону, обмеженого 1 048 576 рядками та 16 384 (XFD) стовпцями, функція INDIRECT повертає помилку #REF!. Примітка.
- A1. Обов'язковий аргумент. Логічне значення, яке вказує тип посилання, що міститься в аргументі "посилання_на_клітинку".
- Якщо клітинка A1 має значення TRUE (істина) або пропущена, значення аргументу "посилання_на_клітинку" інтерпретується як посилання у стилі A1.
- Якщо клітинка A1 має значення FALSE (хибність), значення аргументу "посилання_на_клітинку" інтерпретується як посилання у стилі R1C1.

Приклад

Скопіюйте дані прикладу з наведеної нижче таблиці та вставте їх у клітинку A1 нового аркуша Excel. Щоб відобразити результат обчислення формул, виберіть їх, натисніть клавішу F2, а потім – клавішу Enter. За потреби можна змінити ширину стовпців, щоб відобразити всі дані.

Дані		
B2	1,333	
B3	45	
Георгій	10	
5	62	

Формула	Опис	Результат
=INDIRECT(A2)	Значення посилання у клітинці A2. Посилання веде до клітинки B2, яка містить значення 1,333.	1,333
=INDIRECT(A3)	Значення посилання у клітинці A3. Посилання веде до клітинки B3, яка містить значення 45.	45
=INDIRECT(A4)	Оскільки клітинка A4 має визначене ім'я «Георгій», посилання на це визначене ім'я веде до клітинки B4, яка містить значення 10.	10
=INDIRECT("B"&A5)	Комбінує «B» зі значенням у клітинці A5, яке становить 5. Це, у свою чергу, посилається на клітинку B5, яка містить значення 62.	62

2.7.10. Функція SUBTOTAL

Повертає проміжні підсумки списку або бази даних. Загалом простіше створити список із проміжними підсумками, скориставшись командою Проміжні підсумки із групи Структура на вкладці Дані у програмі Excel для настільних комп'ютерів. Після створення списку проміжних підсумків його можна змінювати редагуванням функції SUBTOTAL.

Синтаксис

SUBTOTAL(номер_функції;посилання1;[посилання2];...)

Синтаксис функції SUBTOTAL має такі аргументи:

- **Номер_функції** (обов'язковий аргумент). Число від 1 до 11 або від 101 до 111, що вказує, яку функцію використовувати для проміжних підсумків. Число від 1 до 11 містить приховані вручну рядки, а число від 101 до 111 виключає ці рядки; відфільтровані клітинки виключаються завжди.

Function_num(включає приховані рядки)	Function_num(ігнорує приховані рядки)	Функція
1	101	AVERAGE
2	102	COUNT
3	103	COUNTA
4	104	MAX
5	105	MIN
6	106	PRODUCT
7	107	STDEV
8	108	STDEVP
9	109	SUM
10	110	VAR
11	111	VARP

- **посилання1** – обов'язковий аргумент. Перший іменований діапазон або посилання, для яких потрібно обчислити проміжні підсумки.

- посилання2;... – необов'язковий аргумент. Від 2 до 254 іменованих діапазонів або посилань, для яких потрібно обчислити проміжні підсумки.
Примітки
- Якщо в посиланні1 є інші проміжні підсумки, посилання2,... (або вкладені проміжні підсумки) ці вкладені проміжні підсумки ігноруються, щоб уникнути подвійного підрахунку.
- Для констант аргументу «номер_функції» від 1 до 11 функція SUBTOTAL включає значення рядків, прихованих командою Приховати рядки із вкладеного меню Приховати або відобразити команди Форматувати у групі Клітинки на вкладці Основне у програмі Excel для настільних комп'ютерів. Ці константи використовуються, коли потрібно відображати підсумки для прихованих і неприхованих чисел у списку. Для констант аргументу «номер_функції» від 101 до 111 функція SUBTOTAL ігнорує значення рядків, прихованих командою Приховати рядки. Ці константи використовуються, коли потрібно відображати підсумки тільки для неприхованих чисел у списку.
- Функція SUBTOTAL ігнорує будь-які рядки, не включені до результату фільтра, незалежно від того, яке значення номера функції використовується.
- Функція SUBTOTAL призначена для стовпців даних або вертикальних діапазонів. Вона не призначена для рядків даних або горизонтальних діапазонів. Наприклад, якщо обчислюються підсумки горизонтального діапазону з використанням номера функції 101 або більше, як-от SUBTOTAL(109;B2:G2), приховання стовпця не вплине на проміжний підсумок. Але приховання рядка у проміжному підсумку вертикального діапазону вплине на проміжний підсумок.
- Якщо посилання містять об'ємні посилання, функція SUBTOTAL повертає #VALUE! .

Приклад

Скопіюйте дані прикладу з наведеної нижче таблиці та вставте їх у клітинку A1 нового аркуша Excel. Щоб відобразити результат обчислення формул, виберіть їх, натисніть клавішу F2, а потім – клавішу Enter. За потреби можна змінити ширину стовпців, щоб відобразити всі дані.

Дані		
120		
10		
150		
23		
Формула	Опис	Результат
=SUBTOTAL(9;A2:A5)	Сума проміжних підсумків діапазону клітинок A2:A5 із використанням 9 як першого аргументу.	303
=SUBTOTAL(1;A2:A5)	Середнє проміжних підсумків діапазону клітинок A2:A5 із використанням 1 як першого аргументу.	75,75
Примітки		
Для функції SUBTOTAL як перший аргумент завжди потрібно		

Дані		
використовувати числовий аргумент (від 1 до 11, від 101 до 111). Цей числовий аргумент застосовується до проміжних підсумків значень (діапазонів клітинок, іменованих діапазонів), які вказуються як наступні аргументи.		

2.8. Автоматизація розрахунків із Power Query та Power Pivot

Якщо стандартних формул Excel недостатньо для автоматизації розрахунків, можна використовувати інструменти Power Query та Power Pivot:

Power Query дозволяє імпортувати, очищувати та перетворювати дані перед їх використанням у звичайних розрахунках.

Power Pivot розширює можливості роботи з великими обсягами даних та дозволяє створювати складні звіти на основі реляційних моделей.

Зведені таблиці (Pivot Tables) – це потужний інструмент для аналізу великих обсягів даних, які дозволяють швидко створювати узагальнення, обчислення та перегляд різних зрізів даних. Зведені таблиці значно полегшують аналіз даних.

Ось покрокова інструкція, як ефективно працювати зі зведеними таблицями:

1. Створення зведеної таблиці

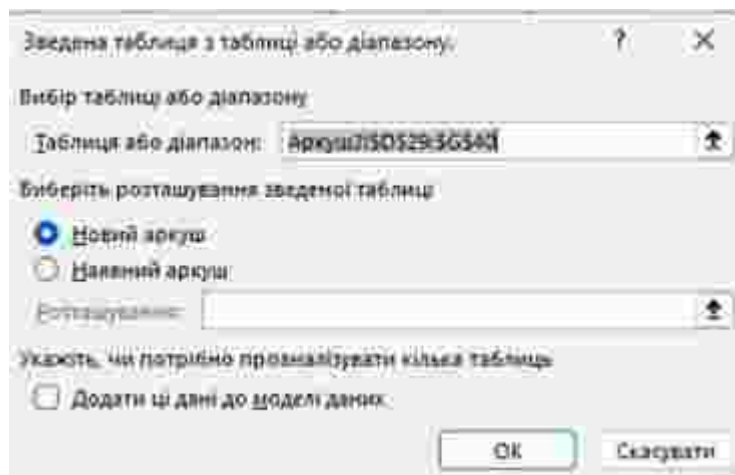
1. Підготовка даних:

- о Переконайтеся, що ваші дані організовані у вигляді таблиці з заголовками.
- о Видаліть порожні рядки та заповніть пропущені дані, якщо можливо.

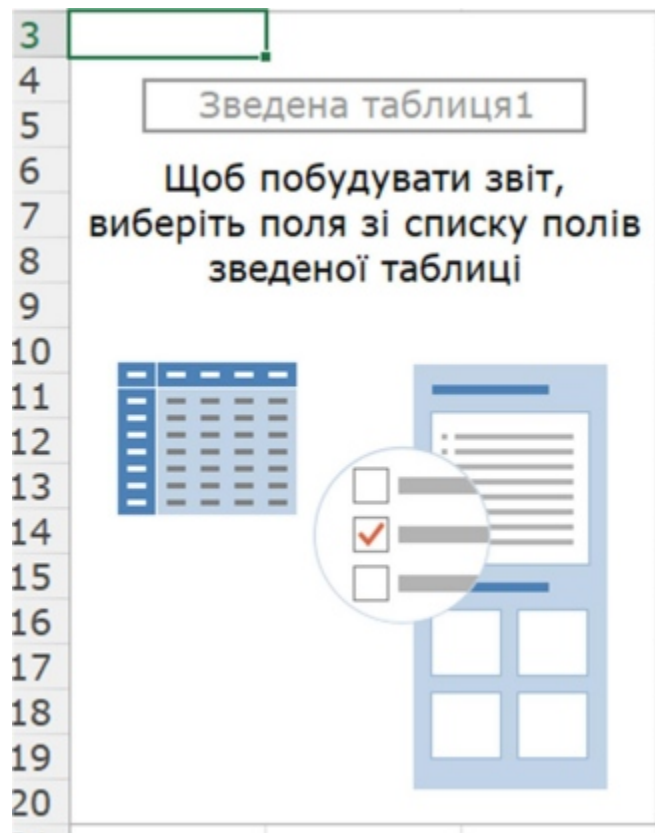
Для прикладу було створено наступну таблицю

Дата	Продукт	Продажі	Регіон
01.01.2025	Ноутбук	1500	Київ
01.01.2025	Мишка	25	Харків
03.02.2025	Клавіатура	416	Дніпро
03.02.2025	Дисплей	403	Дніпро
03.02.2025	Модем	485	Київ
12.03.2025	Web-камера	450	Харків
12.03.2025	Сканер	281	Харків
12.03.2025	Принтер	534	Київ
17.04.2025	Блок живлення	426	Київ
17.04.2025	SSD-диск	156	Дніпро
17.04.2025	Флешка	491	Дніпро

2. Вставка зведеної таблиці:
 - о Виділіть діапазон даних.
 - о Перейдіть до вкладки Вставка → Зведена таблиця.
 - о Виберіть розташування (новий аркуш або існуючий). У прикладі було обрано новий аркуш.



Після натискання кнопки «ОК» на новій сторінці з'явилося запрошення з налаштування зведеної таблиці.



2. Налаштування зведеної таблиці

1. Додавання полів:

о У правій частині вікна ви побачите список полів.

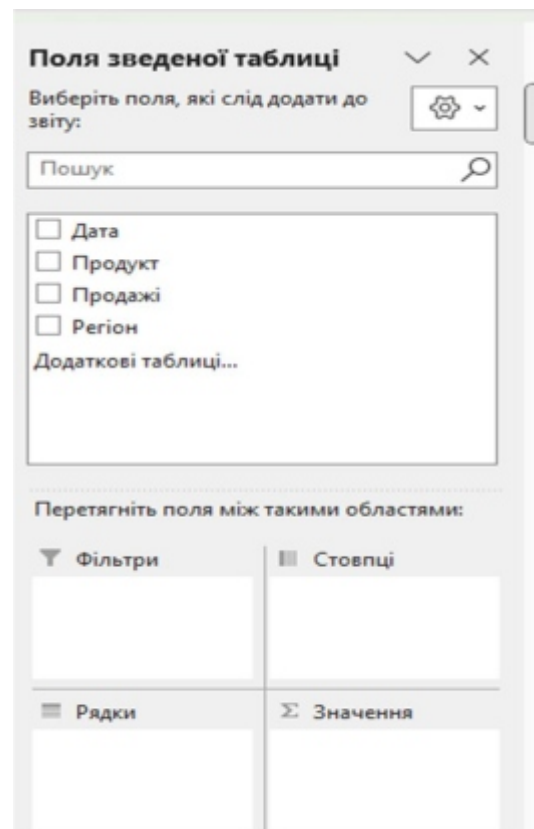
о Перетягніть поля у відповідні області:

- Рядки: Дані, які відображаються як заголовки рядків.

- Стовпці: Дані, які відображаються як заголовки стовпців.

- Значення: Обчислювані дані (сумма, середнє тощо).

- Фільтри: Додаються для створення умовного відображення.

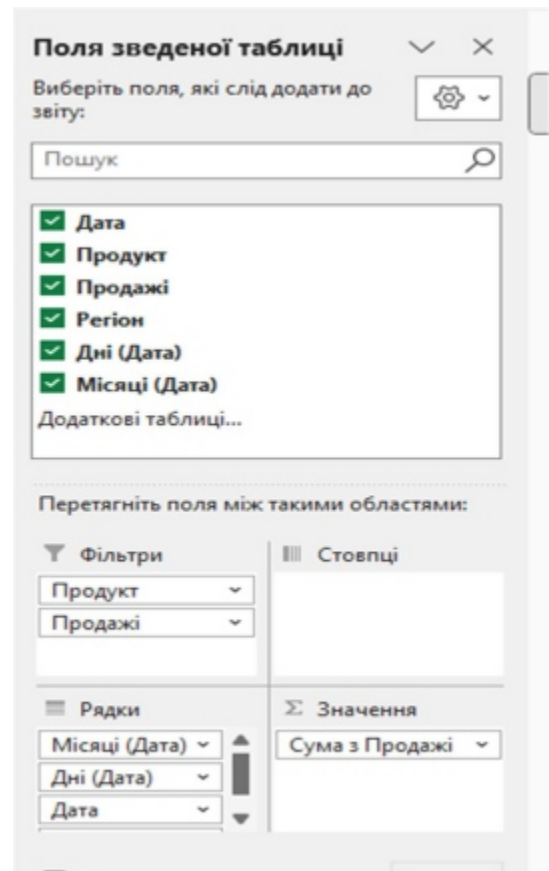


2. Форматування полів:

- о Клацніть правою кнопкою миші на полі, щоб відкрити параметри.
- о Наприклад, змініть функцію обчислення на "Середнє", "Максимум", "Кількість" тощо.

Для прикладу у поле «Рядки» було перетягнуто поле «Дата», а система автоматично додала іще пункти «Дні» та «Місяці». У поле «Фільтри» перетягнуто поля «Продукти» і «Продажі». У поле «Значення» перетягнуто поле «Продажі» і там утворилася позначка «Сума Продажі».

Тоді на новому листі з'явилася наступна зведена таблиця.



	A	B
1	Продукт	(усі)
2	Продажі	(усі)
3		
4	В	Сума з Продажі
5	⊕ Січ	1525
6	⊕ Лют	1304
7	⊕ Бер	1265
8	⊕ Кві	1073
9	Загальний підсумок	5167
10		

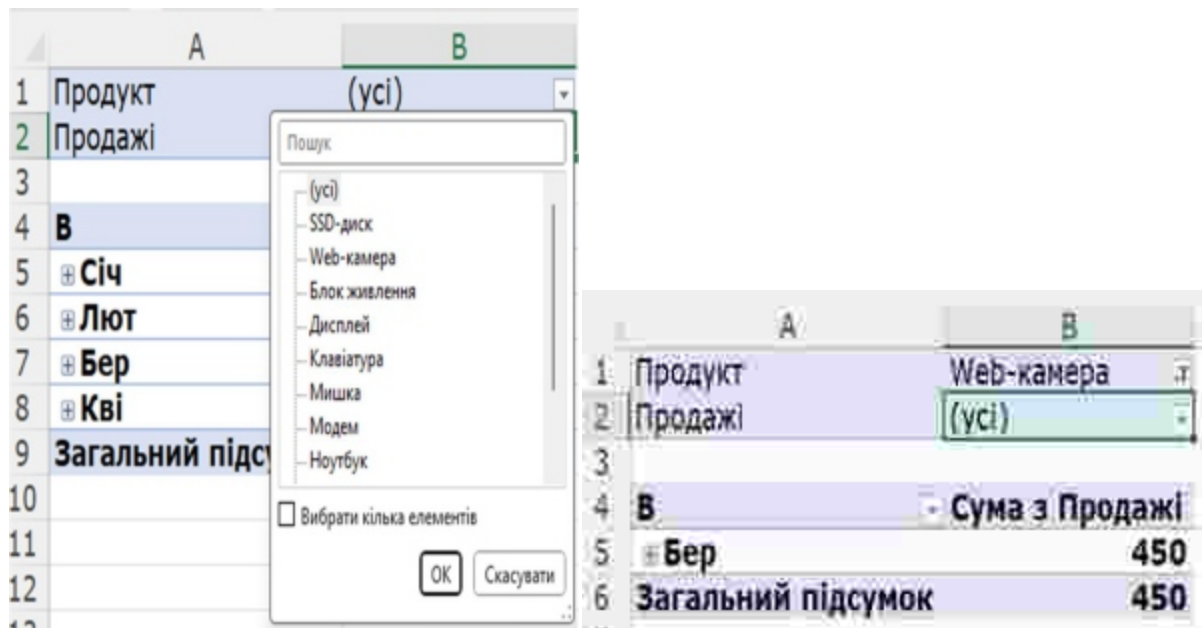
3. Аналіз даних

1. Групування даних:

- о Виділіть дані (наприклад, дати) і натисніть праву кнопку миші → Групувати.
- о Наприклад, можна групувати дати за місяцями або роками.

2. Сортування та фільтри:

Використовуйте стрілки в заголовках, щоб сортувати або фільтрувати дані. Ось тут обрано Web-камеру і ми бачимо потрібний результат.



3. Застосування зрізів (Slicers):

Додайте зрізи для зручного фільтрування даних. Це робиться через вкладку Вставка → Зріз.

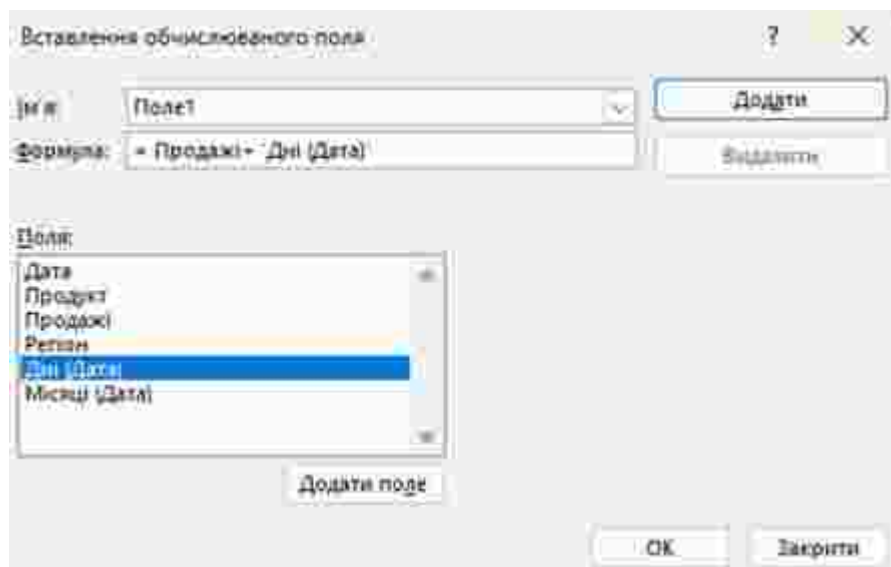
4. Оновлення даних

1. Якщо вихідні дані змінюються, виділіть зведену таблицю та натисніть Оновити (на вкладці Аналіз або правою кнопкою миші).

2. Для автоматичного оновлення можна записати макрос.

5. Використання обчислюваних полів

- Ви можете створювати свої формули всередині зведеної таблиці:
 - Перейдіть до Аналіз → Обчислюване поле.
 - Наприклад, створіть новий стовпець для підрахунку маржинального доходу. Формула створюється у вікні «Вставлення обчислювального поля». Внизу показано список полів. Поле потрібно відмітити і натиснути кнопку «Додати поле». У прикладі до кількості продажів додано дату, хоча це не вплинуло на результат.



	A	B	C
1	Продукт	(усі)	
2	Продажі	(усі)	
3			
4	В	Сума з Продажі	Сума з Поле1
5	⊕ Січ	1525	1525
6	⊕ Лют	1304	1304
7	⊕ Бер	1265	1265
8	⊕ Кві	1073	1073
9	Загальний підсумок	5167	5167

6. Застосування форматування

1. Виділіть таблицю та налаштуйте стиль через вкладку Дизайн.
2. Використовуйте умовне форматування для виділення ключових показників.

Для ипренування радимо виконати наступну прост вправу зі зведеною таблицею.

Вихідні дані:

Дата	Продукт	Продажі	Регіон
01.01.2025	Ноутбук	1500	Київ
01.01.2025	Мишка	25	Харків

Мета: Аналіз загальних продажів за регіонами.

Результат зведеної таблиці:

Регіон	Сума продажів
Київ	1500
Харків	25

Ось поглиблені приклади роботи зі зведеними таблицями в Excel, які допоможуть розкрити їх потенціал для аналізу великих і складних наборів даних:

Приклад 1: Аналіз продажів із групуванням по місяцях

Задача:

У вас є таблиця з даними про продажі за кожен день, і ви хочете підсумувати продажі за місяцями, а також візуалізувати це.

1. Дані:
2. Дії:
 - о Виділіть дані й перейдіть до Вставка → Зведена таблиця.
 - о У секцію "Рядки" перетягніть поле Дата.
 - о У секцію "Значення" перетягніть поле Продажі.
 - о Групуйте дати за місяцями:
 - Клацніть правою кнопкою на дату → Групувати → Виберіть "Місяці".
3. Результат:
4. Додатково:
 - о Перетягніть поле Категорія у стовпці, щоб розбити продажі за категоріями.
 - о Вставте графік для візуалізації.

Приклад 2: Використання обчислюваного поля

Задача:

Розрахувати маржинальний дохід (прибуток) на основі наданих витрат і продажів.

1. Дані:
2. Дії:

- о Створіть зведену таблицю, де Товар буде у "Рядках", а Продажі й Витрати у "Значеннях".

- о Додайте обчислюване поле:

- Перейдіть до Аналіз → Поля, елементи та набори → Обчислюване поле.

- Введіть формулу: =Продажі - Витрати.

3. Результат:

Приклад 3: Використання зрізів і часових шкал

Задача:

Фільтрувати дані за категоріями та роками, щоб показувати лише потрібні зрізи.

1. Дані:

- о Додайте зріз: У вкладці Аналіз виберіть Вставити зріз і виберіть поле Категорія.

- о Додайте часову шкалу: Виберіть Часову шкалу для поля Дата.

- о Використовуйте зрізи й часову шкалу для швидкого фільтрування даних.

2. Результат: Ви можете швидко відображати продажі за певними категоріями, наприклад, лише "Електроніка" за 2025 рік.

Приклад 4: Гніздові рівні деталізації

Задача:

Показати загальні продажі за регіонами та деталізувати до рівня міст.

1. Дані:

2. Дії:

- о Додайте Регіон до "Рядків".

- о Додайте Місто під Регіоном.

- о Додайте Продажі до "Значень".

- о Переконайтеся, що дані згруповані ієрархічно.

3. Результат:

Приклад 5: Використання умовного форматування

Задача:

Підсвітити найбільші продажі у зведеній таблиці.

1. Дії:

- о Виділіть зону даних у зведеній таблиці.
- о Перейдіть до Головна → Умовне форматування → Набори іконок.
- о Виберіть відповідний стиль, наприклад, градієнт кольорів чи стрілки.

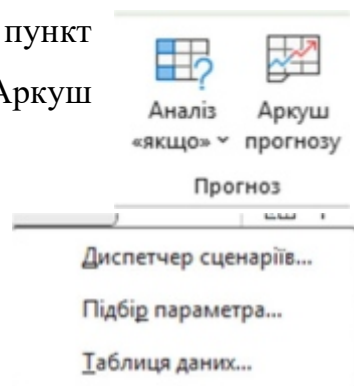
2. Результат: Дані автоматично підсвічуються, і вам легко знайти найбільші значення.

Ці поглиблені приклади демонструють, як зведені таблиці дозволяють аналізувати дані на різних рівнях та використовувати потужні інструменти для візуалізації, групування й обчислення

2.9. Прогноз

За пунктом меню «Дані» можна знайти пункт «Прогноз», що має два підпункти – «Аналіз якщо» та «Аркуш прогнозу». Перший має три підпункти.

Розберемо кожен з них.



2.9.1. Диспетчер сценаріїв

Диспетчер сценаріїв (Scenario Manager) в Excel — це потужний інструмент для проведення аналізу «що-якщо», який дозволяє створювати різні варіанти для набору вхідних даних і відображати відповідні результати, зберігаючи їх для подальшого порівняння.

Опис інструменту:

Диспетчер сценаріїв дозволяє створювати і зберігати різні варіанти значень для заданих клітинок, після чого можна легко порівнювати результати в

таблицях та на графіках. Цей інструмент корисний для прогнозування результатів при зміні вхідних параметрів.

Кроки для використання диспетчера сценаріїв:

1. Створення сценаріїв:

- o Кожен сценарій містить набір значень для певних клітинок (вхідні змінні) і результат, який залежить від цих значень.
- o Наприклад, для планування бюджету можна створити сценарії для різних варіантів доходу та витрат і подивитись, як це змінить кінцевий результат.

2. Порівняння результатів сценаріїв:

- o Диспетчер сценаріїв дозволяє порівнювати результати різних варіантів (сценаріїв) в таблиці або графіку.

3. Підсумкові дані для сценаріїв:

- o Крім того, можна створити підсумкову таблицю для порівняння різних результатів від кількох сценаріїв на основі заданих вхідних змінних.

Як працює Диспетчер сценаріїв:

1. Вибір клітинок для сценаріїв:

- o Необхідно вибрати клітинки (або кілька клітинок), значення яких ви хочете змінювати у кожному сценарії (наприклад, ціни, витрати або інші змінні параметри).

2. Створення сценаріїв:

- o Відкривається вікно, де ви вводите значення для цих клітинок для кожного сценарію.
- o Кожен сценарій має своє ім'я і опис.

3. Перегляд результатів:

- o Після створення сценаріїв можна порівняти їх результати, не змінюючи безпосередньо значення у клітинках таблиці. Це дозволяє аналізувати різні варіанти без необхідності вручну змінювати дані.

Як використовувати Диспетчер сценаріїв:

1. Вибери клітинки, які будуть змінюватися у різних сценаріях.
2. Перейдіть на вкладку **Дані** в меню Excel.
3. У групі **Аналіз** натисніть "**Диспетчер сценаріїв**".
4. У вікні **Диспетчер сценаріїв** натисніть "**Додати**".
5. Введіть ім'я сценарію, а також вказівки для змінних клітинок.
6. Повторіть кроки для додавання інших сценаріїв.
7. Після додавання кількох сценаріїв натискайте "**Переглянути**" для порівняння результатів.
8. Для створення підсумкової таблиці натисніть "**Підсумкові дані**", щоб автоматично створити таблицю з результатами для всіх сценаріїв.

Приклад:

1. **Таблиця:** У вас є таблиця з розрахунками доходів і витрат на основі кількох змінних (наприклад, ціна одиниці товару та кількість проданих одиниць).
2. **Сценарії:** Ви хочете створити різні варіанти ціни та кількості і подивитись, як це вплине на загальний дохід:
 - о Сценарій 1: ціна — 100, кількість — 500
 - о Сценарій 2: ціна — 120, кількість — 400
 - о Сценарій 3: ціна — 150, кількість — 350
3. **Додавання сценаріїв:**
 - о Після вибору клітинок, які містять ці значення (наприклад, A1 для ціни та B1 для кількості), додайте три сценарії через **Диспетчер сценаріїв**.
4. **Підсумкова таблиця:**
 - о Створіть підсумкову таблицю для порівняння результатів (наприклад, загальний дохід для кожного сценарію).

Покрокова інструкція:

1. Уведіть таблицю з розрахунками:
 - о Наприклад, у клітинці **C1** — формула для загального доходу:
 $=A1*B1$ (де A1 — ціна, B1 — кількість).

2. Перейдіть на вкладку **Дані > Аналіз > Диспетчер сценаріїв**.
3. У вікні **Диспетчер сценаріїв** натисніть **"Додати"**:
 - о Уведіть ім'я сценарію (наприклад, "Сценарій 1").
 - о Виберіть клітинки для змінних (A1 і B1).
 - о Уведіть значення для сценарію (ціна 100 і кількість 500).
4. Повторіть цей процес для двох інших сценаріїв:
 - о "Сценарій 2" з ціною 120 і кількістю 400.
 - о "Сценарій 3" з ціною 150 і кількістю 350.
5. Перейдіть до **Підсумкові дані** і виберіть клітинку (C1), де буде обчислюватися результат для кожного сценарію.
6. В Excel автоматично створиться таблиця з результатами для кожного сценарію, що дозволить вам порівняти їх.

Переваги використання Диспетчера сценаріїв:

- Зручність в аналізі різних варіантів, не змінюючи вручну дані в таблиці.
- Легкість порівняння різних сценаріїв і їх результатів.
- Можливість збереження варіантів і швидкий доступ до них.

Примітка: Диспетчер сценаріїв — це одна з функцій **Аналізу «що-якщо»** в Excel, і вона може бути дуже корисною для фінансового планування, бюджетування, прогнозування результатів або будь-якого іншого аналізу, де варіанти зміни вхідних значень впливають на результати.

2.9.2. Підбір параметра

Підбір параметра (Goal Seek) в Excel – це інструмент для вирішення задачі з пошуку значення вхідної змінної, при якому певна формула або вираз дасть бажаний результат. Це класичний метод вирішення задач типу "що-якщо", коли потрібно знайти такий параметр, який приводить до заданого результату.

Опис інструмента Підбір параметра:

Підбір параметра дозволяє знаходити значення, яке потрібно для досягнення певного результату в формулі або виразі. Наприклад, якщо у вас є формула для підрахунку процентної ставки, і ви хочете дізнатись, яку ставку

потрібно встановити, щоб отримати певний результат (наприклад, місячний платіж), то цей інструмент допоможе вам.

Як працює інструмент Підбір параметра:

1. **Формула:** У вас є формула або обчислення, яке залежить від однієї змінної.
2. **Результат:** Ви хочете, щоб результат формули досяг певного значення.
3. **Задача:** Ви вказуєте Excel, яку змінну потрібно змінити для досягнення цього результату.

Кроки для використання Підбору параметра:

1. **Заповніть формулу:** Введіть формулу, яка містить одну змінну, значення якої ви хочете знайти.
2. **Використовуйте функцію:**
 - о Перейдіть на вкладку **Дані**.
 - о У групі **Аналіз** виберіть **Підбір параметра** (Goal Seek).
3. **Вказання значень:**
 - о В вікні **Підбір параметра** потрібно вказати:
 - **В клітинці** — клітинка, яка містить формулу, що залежить від змінної.
 - **Досягти значення** — значення, яке ви хочете отримати в результаті обчислення.
 - **Змінити клітинку** — клітинка, значення якої ви хочете змінити, щоб отримати бажаний результат.
4. **Натискайте ОК:** Excel автоматично визначить значення змінної, яке дасть бажаний результат.

Приклад використання Підбору параметра:

Завдання:

У вас є формула для обчислення **місячного платежу за кредитом:**

$$PMT = \frac{P \cdot r}{1 - (1 + r)^{-n}}$$

де:

- PMT – місячний платіж,
- P – основна сума кредиту,
- r – процентна ставка за місяць,
- n – кількість місяців.

Припустимо, ви хочете дізнатися, яку процентну ставку потрібно встановити, щоб місячний платіж був 500 грн, якщо сума кредиту 10 000 грн і термін 24 місяці.

Кроки:

1. Введення формули:

- У клітинці **B1** введіть основну суму кредиту: 10000.
- У клітинці **B2** введіть термін: 24.
- У клітинці **B3** введіть бажаний місячний платіж: 500.
- У клітинці **B4** введіть процентну ставку (в місяць): 0.05 (5% на місяць).
- У клітинці **B5** введіть формулу для місячного платежу: =PMT(B4, B2, -B1).

2. Використання інструмента:

- Перейдіть на вкладку **Дані > Аналіз > Підбір параметра**.
- В полі **В клітинці** виберіть клітинку, в якій знаходиться результат (у цьому випадку **B5**).
- В полі **Досягти значення** вкажіть значення місячного платежу, яке ви хочете отримати, тобто **500**.
- В полі **Змінити клітинку** виберіть клітинку **B4**, оскільки саме процентну ставку ви хочете змінити.

3. Натискайте ОК:

- о Excel автоматично підбере значення процентної ставки в клітинці **B4**, яке дозволить досягти місячного платежу в 500 грн.

Результат:

Після натискання **ОК**, Excel розрахує необхідну процентну ставку для досягнення бажаного місячного платежу і виведе результат у клітинці **B4**.

Додаткові можливості:

- **Підбір параметра** працює тільки для задач, де є одна змінна (наприклад, процентна ставка). Якщо ви маєте кілька змінних, можна використати інші методи, наприклад, **Пошук рішення** (Solver).

Інші приклади:

1. **Задача для податку:** Ви хочете дізнатися, яку суму дохід потрібно отримати, щоб після податку залишилось певне значення чистого доходу. Ви можете змінювати суму доходу, щоб досягти бажаного чистого результату.
2. **Пошук ціни:** Якщо у вас є формула для розрахунку прибутку на основі ціни товару і кількості проданих одиниць, ви можете використати **Підбір параметра**, щоб знайти таку ціну, яка дасть бажаний рівень прибутку.

Переваги використання Підбору параметра:

- Простота використання.
- Зручний для задач, де є одна змінна.
- Швидко знаходить значення параметра, яке необхідно для досягнення бажаного результату.

2.9.3. Таблиця даних

Таблиця даних (Data Table) в Excel – це потужний інструмент для проведення аналізу «що-якщо». Вона дозволяє швидко побудувати таблицю з результатами для різних варіантів значень вхідних змінних і наочно порівнювати їх. Таблиці даних в Excel можна використовувати для аналізу впливу зміни одного або двох параметрів на результат формули.

Опис інструменту Таблиця даних (Data Table):

Таблиця даних дає змогу побудувати таблицю, яка автоматично обчислює результати для декількох варіантів вхідних значень. Це дозволяє бачити, як зміна одного або двох вхідних параметрів впливає на кінцевий результат. Це особливо корисно для фінансових, економічних або інженерних розрахунків, де потрібно враховувати варіативність певних змінних.

Які типи таблиць даних існують:

1. Таблиця даних з однією змінною (One-Variable Data Table):

- У такій таблиці змінюється тільки один параметр, а Excel обчислює результат для кожного значення цього параметра.
- Наприклад, можна варіювати процентну ставку і бачити, як змінюється місячний платіж за кредитом.

2. Таблиця даних з двома змінними (Two-Variable Data Table):

- У такій таблиці змінюється два параметри одночасно, і Excel обчислює результат для кожної комбінації значень цих параметрів.
- Наприклад, можна варіювати і процентну ставку, і термін кредиту, і бачити, як це впливає на місячний платіж.

Як створити Таблицю даних (Data Table):

1. Таблиця даних з однією змінною (One-Variable Data Table):

Приклад:

Припустимо, ви хочете побудувати таблицю для місячних платежів по кредиту, змінюючи тільки процентну ставку, а інші параметри (сума кредиту і термін) залишаються сталими.

1. Введення початкових даних:

- В клітинку **B1** введіть суму кредиту (наприклад, 10000).
- В клітинку **B2** введіть термін (наприклад, 24).
- В клітинку **B3** введіть процентну ставку (наприклад, 5%).
- В клітинку **B4** введіть формулу для місячного платежу: $=PMT(B3/12, B2, -B1)$.

2. Створення таблиці даних:

- o Створіть таблицю, де в одному стовпці будуть різні варіанти процентних ставок (наприклад, від 4% до 8%).
- o В клітинку поруч з першим значенням процентної ставки введіть формулу для місячного платежу (яка вже є в **B4**).
- o Виділіть весь діапазон, що містить вашу таблицю.
- o Перейдіть на вкладку **Дані > Аналіз > Таблиця даних**.
- o У вікні **Таблиця даних**:
 - У полі **Вхідний параметр** виберіть клітинку з процентною ставкою (у нашому випадку **B3**).
 - Натисніть **ОК**.

3. Результат:

- o Excel автоматично обчислить місячні платежі для кожної процентної ставки в таблиці.

2. Таблиця даних з двома змінними (Two-Variable Data Table):

Приклад:

Тепер уявімо, що ви хочете побудувати таблицю, яка показує місячний платіж для різних варіантів процентної ставки і терміну кредиту.

1. Введення початкових даних:

- o В клітинку **B1** введіть суму кредиту (наприклад, 10000).
- o В клітинку **B2** введіть термін (наприклад, 24).
- o В клітинку **B3** введіть процентну ставку (наприклад, 5%).
- o В клітинку **B4** введіть формулу для місячного платежу: **=PMT(B3/12, B2, -B1)**.

2. Створення таблиці даних:

- o Створіть таблицю, де в одному стовпці будуть варіанти процентних ставок (наприклад, від 4% до 8%), а в рядках — різні терміни кредиту (наприклад, від 12 до 60 місяців).
- o В клітинку на перехресті першого рядка і першого стовпця введіть формулу для місячного платежу (як і в **B4**).
- o Виділіть весь діапазон, що містить вашу таблицю.

- о Перейдіть на вкладку **Дані > Аналіз > Таблиця даних**.
- о У вікні **Таблиця даних**:
 - У полі **Вхідний параметр (рядки)** виберіть клітинку з терміном кредиту (**B2**).
 - У полі **Вхідний параметр (стовпці)** виберіть клітинку з процентною ставкою (**B3**).
 - Натисніть **ОК**.

3. Результат:

- о Excel автоматично обчислить місячні платежі для кожної комбінації процентної ставки і терміну кредиту в таблиці.

Переваги використання Таблиці даних:

- **Швидкість обчислень:** дозволяє обчислювати множину варіантів результатів без потреби вручну змінювати параметри.
- **Зручність:** наочно відображає, як зміна значень параметрів впливає на результат.
- **Порівняння варіантів:** дозволяє легко порівнювати результати для різних значень вхідних параметрів.

Основні моменти:

- **Таблиця даних з однією змінною** змінює лише один параметр, а Excel обчислює результат для всіх варіантів цього параметра.
- **Таблиця даних з двома змінними** змінює два параметри одночасно, і Excel обчислює результат для кожної комбінації цих параметрів.
- **Вхідний параметр (рядки)** і **вхідний параметр (стовпці)** вказують, які клітинки потрібно змінювати для отримання результату.

2.9.4. Аркуш прогнозу

Аркуш прогнозу (Forecast Sheet) в Excel – це інструмент для прогнозування майбутніх значень на основі історичних даних. Використовуючи цей інструмент, ви можете створювати прогнози на основі наявних даних, що дозволяє виявити

тенденції і робити прогнозування для майбутніх значень, таких як продажі, витрати, попит на продукцію і т.д.

Цей інструмент використовує методи статистичного аналізу, такі як **експоненціальне згладжування (ETS)**, щоб створити модель, яка підходить для вашого набору даних і прогнозує майбутні значення на основі трендів, сезонності та циклічності.

Опис інструмента Аркуш прогнозу:

Інструмент **Аркуш прогнозу** в Excel дозволяє автоматично створити прогнози, побудувати графіки і оцінити точність прогнозів на основі історичних даних. Ви можете задати різні параметри для налаштування прогнозу, наприклад, період, на який хочете зробити прогноз, та метод обчислення.

Як працює Аркуш прогнозу:

1. **Історичні дані:** Необхідно мати дані, які містять дату або час, а також значення, які ви хочете прогнозувати.
2. **Прогнозування:** Excel автоматично визначає модель, яка найбільш підходить для вашого набору даних і прогнозує значення для майбутніх періодів.
3. **Графік:** Excel будує графік для історичних і прогнозованих значень, що допомагає наочно порівняти дані.

Як створити Аркуш прогнозу в Excel:

1. Підготовка даних:

Для коректного прогнозу необхідно мати дві колонки:

- **Дата або періоди (X-значення)** — наприклад, місяці або роки.
- **Значення (Y-значення)** — значення, які потрібно прогнозувати, наприклад, продажі або витрати.

2. Використання інструмента Аркуш прогнозу:

1. **Вибір даних:** Виділіть діапазон даних, який містить історичні значення. Наприклад, стовпець з місяцями та стовпець з відповідними продажами.
2. **Перехід до інструмента прогнозу:**
 - о Перейдіть на вкладку **Дані** в Excel.

- У групі **Аналіз** виберіть **Аркуш прогнозу** (Forecast Sheet).

3. Налаштування прогнозу:

- У вікні **Аркуш прогнозу**:
 - **Діапазон**: Переконайтеся, що діапазон даних правильний.
 - **Прогнозувати на**: Введіть кількість періодів, на яку потрібно створити прогноз (наприклад, 12 місяців).
 - **Метод прогнозу**: Excel автоматично вибирає метод експоненціального згладжування (ETS), але ви також можете вручну налаштувати параметри, якщо це необхідно.
 - **Графік**: Ви можете вибрати варіант створення графіка для прогнозу (лінійний графік, стовпчиковий графік і т.д.).

4. Перегляд результату:

- Excel автоматично побудує графік для історичних і прогнозованих даних.
- Результати будуть представлені на новому аркуші з прогнозованими значеннями на майбутні періоди.
- Excel також надасть статистику, яка включає рівень довіри для прогнозу (наприклад, верхня і нижня межі прогнозу).

3. Завершення:

Натисніть **ОК**, щоб створити аркуш прогнозу. Excel побудує новий аркуш з таблицею прогнозованих значень і графіком.

Приклад використання Аркуша прогнозу:

Завдання:

Уявімо, що у вас є дані про місячні продажі за останній рік, і ви хочете спрогнозувати продажі на наступні 6 місяців.

1. Введення даних:

- В колонці **A** (місяці): січень, лютий, березень, квітень, травень і т.д.
- В колонці **B** (продажі): 1500, 1700, 1800, 1600, 2000 і т.д.

2. Створення прогнозу:

- Виділіть ваші дані (місяці та продажі).

- o Перейдіть на вкладку **Дані > Аркуш прогнозу**.
- o У вікні **Аркуш прогнозу** встановіть, що ви хочете створити прогноз на наступні 6 місяців.
- o Натисніть **ОК**.

3. Результат:

- o Excel створить новий аркуш, в якому буде прогноз на наступні 6 місяців, а також графік, який показує історичні дані та прогнозовані значення.
- o Ви отримаєте таблицю з прогнозами, а також верхні і нижні межі довіри для кожного прогнозу.

Параметри прогнозу:

1. **Метод прогнозу (ETS):** Основний метод, який використовує Excel для створення прогнозу — це експоненціальне згладжування (ETS). Цей метод добре працює з даними, які мають тренд або сезонність.

2. Перегляд результатів:

- o **Прогнозовані значення:** Excel створить таблицю з прогнозованими значеннями для майбутніх періодів.
- o **Графік:** Excel побудує графік, що з'єднує історичні та прогнозовані значення, щоб наочно показати тренди.
- o **Довірчі інтервали:** Excel також додає довірчі інтервали, які вказують, у якому діапазоні можуть перебувати прогнозовані значення з певною ймовірністю.

Переваги використання Аркуша прогнозу:

1. **Автоматизація процесу:** Excel автоматично обирає найбільш підходящий метод прогнозування для ваших даних, зменшуючи необхідність вручну налаштовувати статистичні моделі.
2. **Графічне відображення:** Графіки допомагають наочно побачити тренди і прогнозовані зміни, що є дуже корисним для аналізу та презентації даних.

3. **Зручність:** Це потужний інструмент для швидкого прогнозування на основі історичних даних, який можна використовувати в бізнес-аналізі, фінансах, економіці і багатьох інших сферах.

Обмеження:

- **Малий обсяг даних:** Прогнозування може бути менш точним, якщо дані для трендів і сезонності мають невеликий обсяг або не відображають явних закономірностей.
- **Типи даних:** Excel прогнозує тільки для даних з чіткими трендами або сезонністю. Для випадкових або хаотичних даних цей інструмент може бути менш ефективним.

2.10. Індивідуальне завдання № 2.

Робота з новими можливостями Excel

Завдання: засвоїти додаткові можливості Excel.

Порядок виконання: використовуються функції Excel.

Методичні вказівки: 1) Створити таблицю, яка б мала наступні поля: Номер по порядку, Дата, продукт, Продавень, Ціна товару, Кількість реалізації, Місто.

- 2) Кількість типів товарів, прізвищ та міст має бути не менше трьох, а кількість рядків таблиця – не менше 20.
- 3) Для створення числових даних скористайтесь функціями SEQUENCE, UNIQUE та RANDBITWEN.
- 4) Створіть поле, куди ви будете вводити номер по порядку, який викличе функцію XLOOKUP, яка має показати прізвище у цьому рядку.
- 5) Створіть поле, куди ви будете вводити назву міста. Це поле має викликати Функція INDIRECT, яка має показати прізвище, що відповідає цьому місту.

- 6) Створіть власну функцію з двома змінними за допомогою функції LAMBDA.
- 7) Створіть зведену таблицю на іншому листі.
- 8) Проведіть із цією таблицею всі операції, описані у п. 2.8.
- 9) Працюючи з режимом «Вставлення обчислювального поля», використайте власну функцію для полів Ціна товару та Кількість реалізації,
- 10) Звітні наведіть весь порядок роботи з демонстрацією активних вікон і поясненнями до них.

Контрольні запитання

- 1) Яка різниця між відносними, абсолютними та змішаними посиланнями в Excel?
- 2) Як відносне посилання змінюється при копіюванні формули в іншу клітинку?
- 3) Як використовувати абсолютне посилання при створенні формул для конкретної клітинки?
- 4) Коли слід використовувати змішане посилання (наприклад, \$A1 чи A\$1)?
- 5) Як використовувати Майстер функцій для швидкого вибору та вставлення формул?
- 6) Чим відрізняються функції з категорії "Дата та час" від функцій з категорії "Логічні"?
- 7) Як функція IF допомагає приймати рішення в Excel? Можете навести приклад?

- 8) Що робить функція AND, і як вона поєднується з іншими логічними функціями?
- 9) Як функція OR може бути корисною при складанні умов у формулах?
- 10) Як у Excel можна представити логічне висловлювання "Якщо А більше за В і одночасно С менше за D"?
- 11) Яку роль відіграють функції логіки для вирішення завдань, де важливі умови?
- 12) Як використовувати функцію CONCATENATE для об'єднання текстових даних з кількох клітинок?
- 13) Яким чином функція TEXT може допомогти в форматуванні чисел як текст?
- 14) Як використовувати функцію TODAY() для автоматичного визначення поточної дати?
- 15) Як обчислити різницю між двома датами в Excel?
- 16) Що таке масиви в Excel і як їх використовувати для виконання складних розрахунків?
- 17) Які основні переваги нових функцій Excel, таких як UNIQUE, SORT і FILTER?
- 18) Як функція LET допомагає спростити складні формули в Excel?
- 19) Що таке функція LAMBDA, і як її можна застосувати для створення власних функцій?
- 20) Як функція SEQUENCE може допомогти при створенні списків чисел або дат в Excel?

У розділі студенти вивчили нові можливості Excel для роботи з таблицями таких функцій Excel як Power Query та Power Pivot.

Розділ 3.

ІНТЕРАКТИВНІ ЕЛЕМЕНТИ EXCEL

*Вивчивши матеріали цього розділу
студенти опанують роботу з
інтерактивними елементами Excel.*

3.1. Використання списків, що фільтруються, та таких, що випадають

Списки, що фільтруються, є корисним інтерактивним елементом в Excel, який дозволяє користувачам вибирати значення з попередньо заданого списку, скорочуючи можливі помилки і забезпечуючи однорідність введених даних. Ось як їх використовувати:

Створення списку:

1. Підготовка джерела даних.

- о Введіть перелік значень для списку в окремому стовпці або діапазоні на вашому листі. Наприклад, список міст: Київ, Львів, Одеса.

Місто	Посада	Прізвище
Київ	Менеджер	Шевченко
Львів	Аналітик	Коваленко
Одеса	Програміст	Ткаченко
Харків	Менеджер	Іваненко
Дніпро	Аналітик	Сидоренко
Запоріжжя	Програміст	Поліщук
Черкаси	Менеджер	Мельник
Чернівці	Аналітик	Бондаренко
Івано-Франківськ	Програміст	Гончаренко
Вінниця	Менеджер	Петренко
Тернопіль	Аналітик	Кравченко

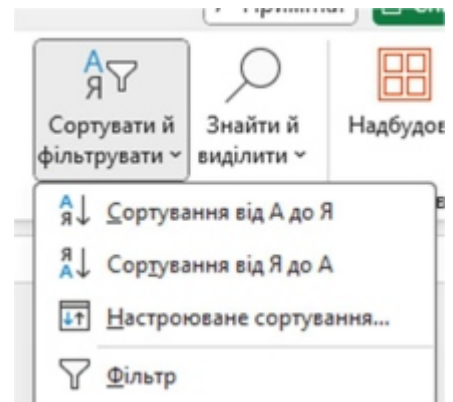
Місто	Посада	Прізвище
Полтава	Програміст	Бойко
Суми	Менеджер	Лисенко
Миколаїв	Аналітик	Савченко
Херсон	Програміст	Жук

2. Вибір клітинок для списку.

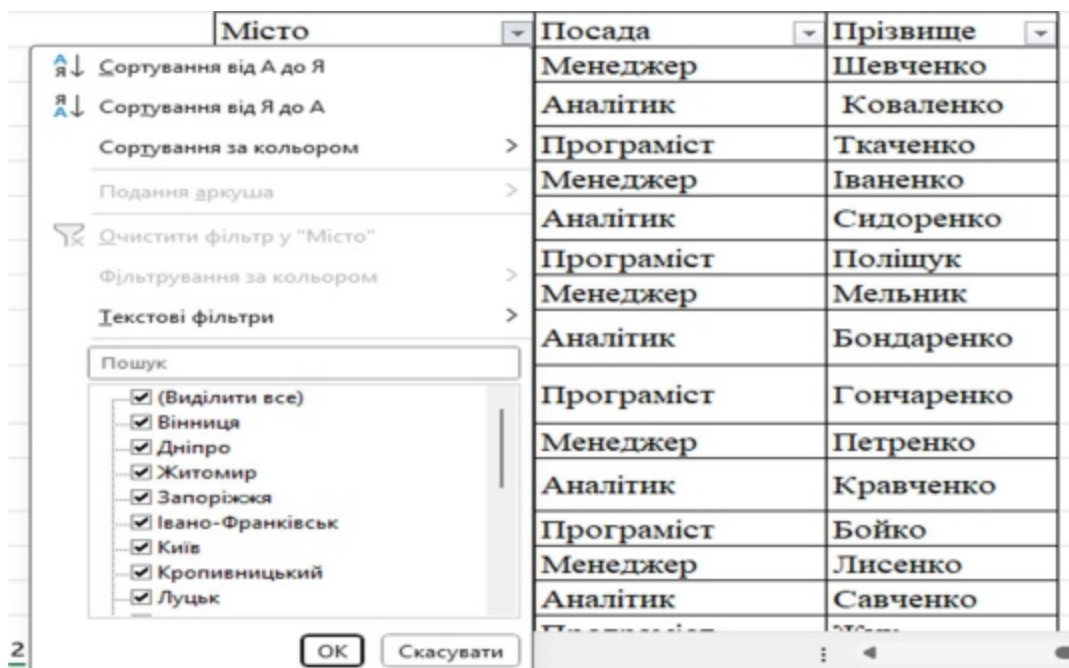
- о Виділіть клітинки, де має бути список, що випадає.

3. Налаштування перевірки даних.

- о Перейдіть на вкладку «Основне» «Сортування та фільтрація».
- о Оберіть Фільтр.
- о У полі Джерело введіть діапазон, де знаходяться ваші значення, наприклад: `=A$1:A$3`.



Тепер кожне з відмічених полів має стрілочку, що дозволяє фільтрувати дані.



Варто тільки прибрати галочки зі списку, і у таблиці ці рядки стануть невидимими.

Місто	Посада	Прізвище
Київ	Менеджер	Шевченко
Дніпро	Аналітик	Сидоренко
Запоріжжя	Програміст	Поліщук
Кропивницький	Менеджер	Хоменко

Додаткові налаштування:

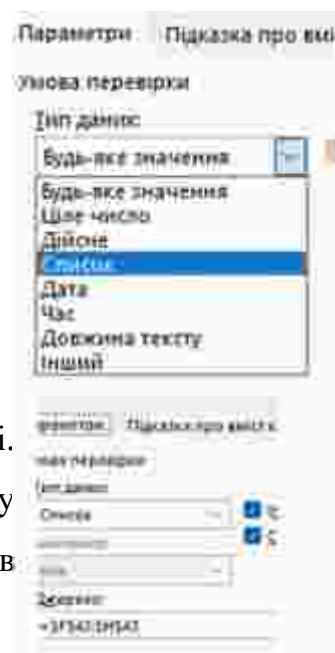
- Повідомлення про введення даних: Ви можете налаштувати вікно підказки для пояснення, що потрібно вибрати.
- Обмеження введення: Якщо введене значення не відповідає списку, можна налаштувати попередження або блокування.

Практичне застосування:

- Автоматизація процесів введення даних.
- Забезпечення однорідності в анкетах, таблицях чи звітах.
- Полегшення роботи з великими наборами даних.

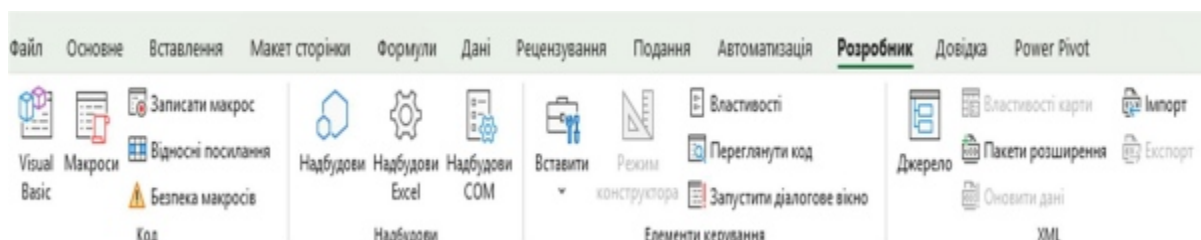
Щоб створити списки, що випадають:

- Виділіть комірку де буде розташований цей список, наприклад, F1,
- Перейдіть в "Дані" → «Знаряддя даних» → "Перевірка даних",
- Виберіть "Список" і введіть посилання на список товарів. Цей список можна розташувати у колонку на будь-якій частині аркушу, або й на іншому аркуші. Якщо він на поточному аркуші, то літери списку можна зробити кольору тла, щоб він не заважав основним елемента управління.
- Тепер у комірці F1 можна вибирати елементи списку.



3.2. Використання кнопок, прапорців та перемикачів

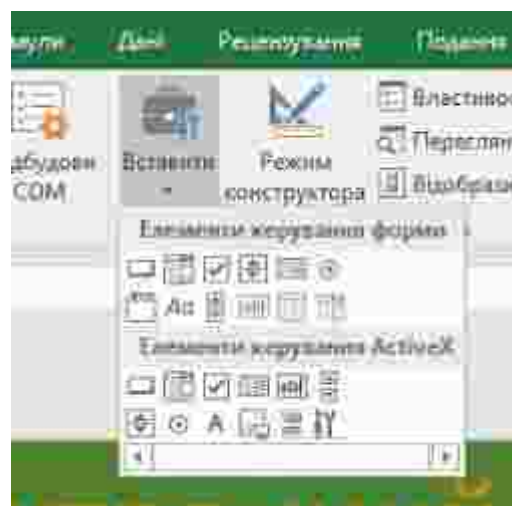
Клацніть по пункту головного меню «Розробник» і оберіть пункт «Вставити».



Там відкриється меню з усіма елементами керування форми, робота з якими буде описана у наступних підпунктах.

В подальшому описі ми будемо обирати елементи форми тільки з верхнього меню «Елементи керування формами».

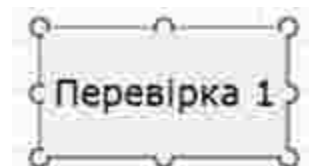
Якщо підвести курсор мишки до конкретної піктограми, з'явиться назва цього елемента, тому при їх описі піктограми показуватися додаткова не будуть.



3.2.1. Кнопка

Якщо вибрати елемент «Кнопка», то курсор мишки перетвориться на маленький хрестик. Тепер потрібно підвести його до потрібного місця та відмітити позицію кнопки, як певний прямокутник.

Тепер цей елемент доступний до редагування. Клацніть по ньому правою кнопкою мишки, виберіть пункт «Редагування тексту» і напишіть той текст, що має бути на кнопці.

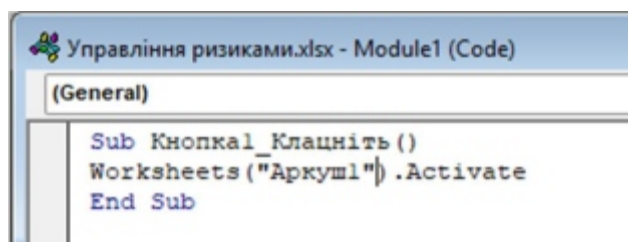
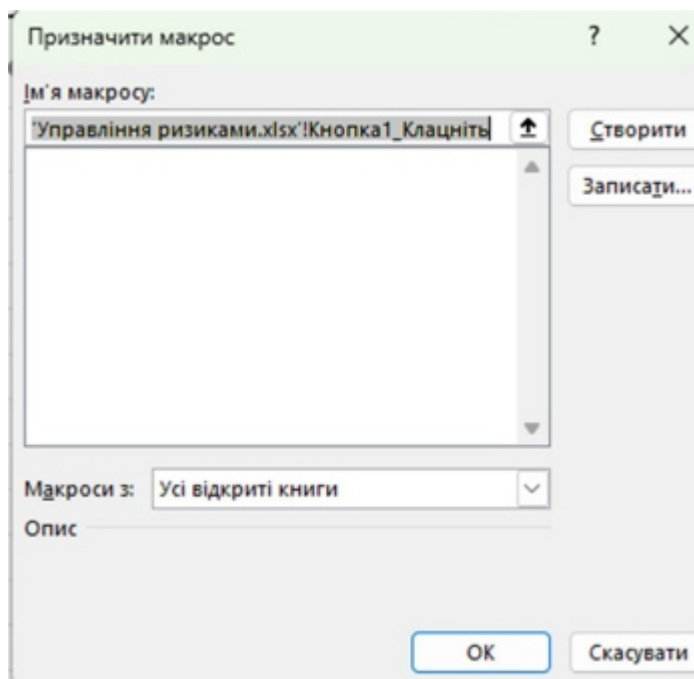


Рамка також дозволяє змінювати габарити кнопки так само, як ми змінюємо габарити рисунку.

Одразу з'являється вікно «Призначити макрос». Його можна ігнорувати, але пізніше через праву кнопку мишки та контекстне меню його можна знову обрати і натиснути «ОК».

У новому вікні відкриється редактор команд, у якому ви побачите два рядки тексту, між яким треба вставити потрібну команду.

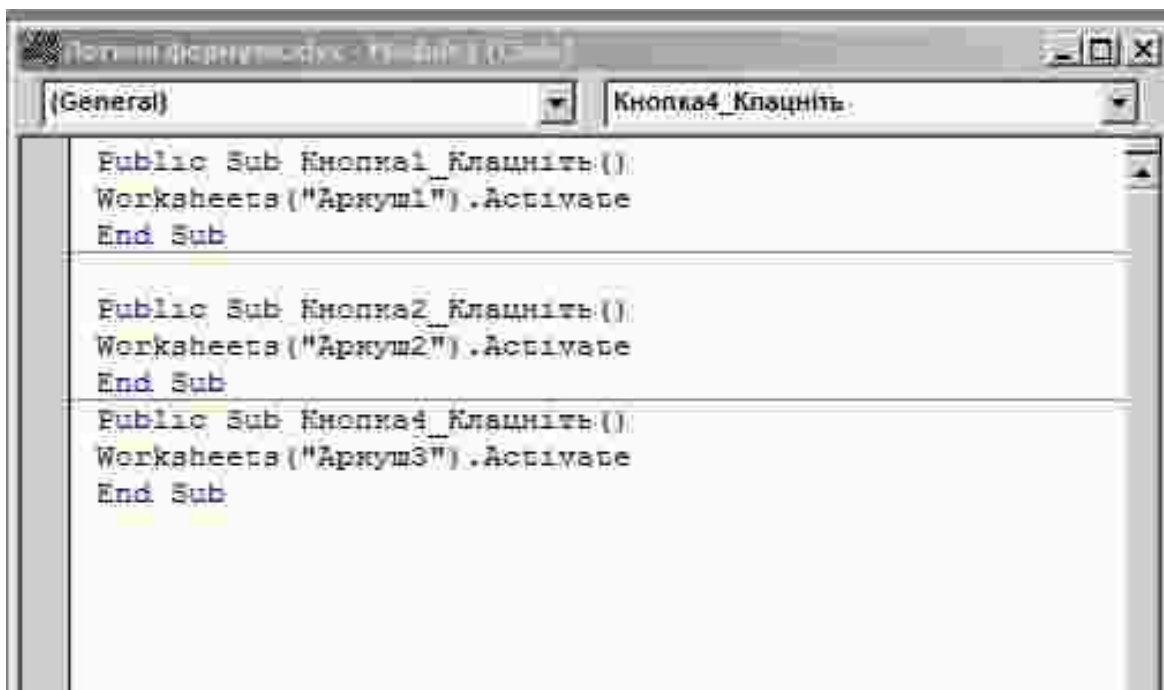
Оскільки кнопки частіше за все використовуються як елементи керування переходам зі сторінки на сторінку, то команда має бути такою:



Worksheets (“ім’я листа, на який потрібно перейти”).Activate

Так потрібно зробити для всіх кнопок, щоб забезпечити перехід до кожного листа. Коли ви створите багато кнопок, ви зможете передивитися список макросів. Він викликається пунктом «Макроси» в меню «Розробник». Зауважте, що всі вони розташовуються в одному вікні і розділені горизонтальною рисою.

У цьому прикладі показано макроси переходу між сторінками. Кнопка_1 та Кнопка_2 розташовані на листі «Аркуш3» і відправляють розрахунок на відповідний лист. Кнопка_3 вміщена на листах «Аркуш1» та «Аркуш2» і призначена для повернення в головне меню.

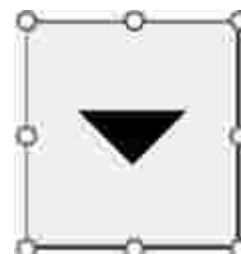


Внутрішні назви кнопок типу «Кнопка1_Кладніть()» генеруються системою автоматично.

3.2.2. Поле зі списком

Його створення аналогічне створенню кнопки, після якого ми побачимо наступний елемент.

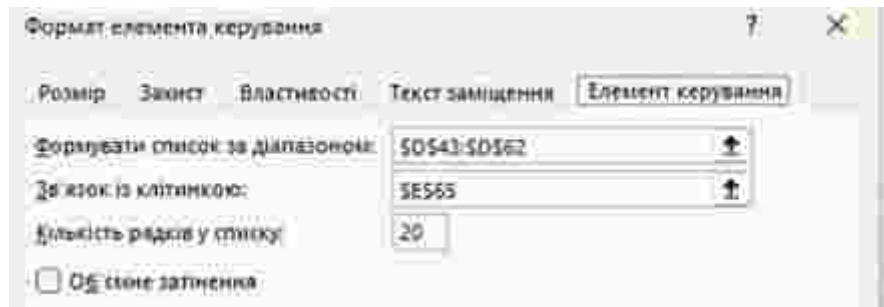
Контекстне меню, викликане правою кнопкою мишки, за пунктом «Формат елемента керування» показує наступне вікно.



Щоб скористатися ним, необхідно спочатку створити список. Скористаємося списком із розділу 2.

Місто	Посада	Прізвище
Київ	Менеджер	Шевченко
Дніпро	Аналітик	Сидоренко
Запоріжжя	Програміст	Поліщук
Кропивницький	Менеджер	Хоменко

Ми відмічаємо список міст у вікні пункту «Формувати список за діапазоном», у пункті «Зв'язок із клітинкою», вказуємо адресу комірки, де буде стояти номер за списком, а у пункті «Кількість рядків у списку», вказуємо таку кількість елементів списку, як буде показуватися одночасно.



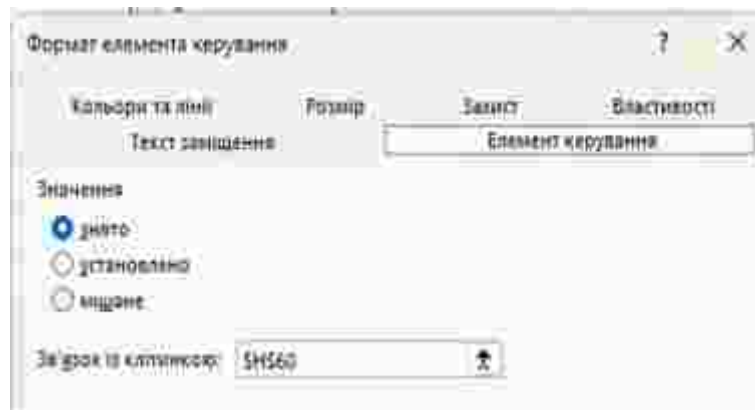
Тепер, якщо клацнути по цій кнопці, ми побачимо список що випадає. Варто лівою кнопкою мишки вибрати якийсь пункт, як у комірці **Зв'язок із клітинкою:** **\$E\$65** випаде порядковий номер цього елемента списку.



3.2.3. Прапорець

Його створення аналогічне попереднім елементам управління, але цей елемент використовується для кожного пункту таблиці.

Для цього у вікні «Зв'язок із клітинкою» указуються клітинки, окремі для кожного елемента списку. Тобто, створивши один такий елемент, його можна скопіювати для всіх пунктів таблиці, але потім у вікні «Формат елементів керування» вказати окремі клітинки, куди буде видано елемент вибору цього прапорця.



Нижче подано варіант роботи прапорців. Якщо ми відмічаємо якийсь прапорець, то у полі, позначеному як зв'язок, буде логічна змінна .

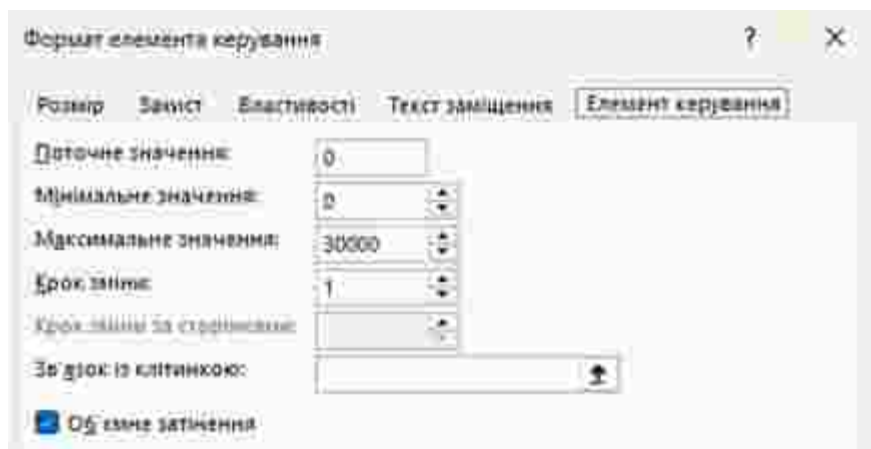
Кропивницький	Менеджер	Хоменко	☐ Прапор	FALSE
Рівне	Аналітик	Романенко	☒ Прапор	TRUE
Луцьк	Програміст	Олійник	☐ Прапор	FALSE

Прапорці в елементах керування Excel корисні для створення інтерактивних таблиць і форм. Вони дозволяють користувачам швидко та зручно вибирати параметри, керувати даними або відмічати виконання певних завдань. Ось кілька прикладів, для чого їх можна використовувати:

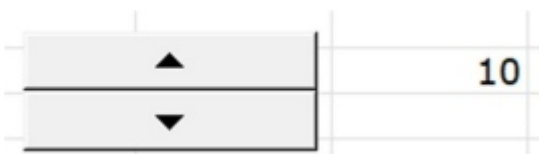
1. **Контроль введення даних:** Наприклад, для вибору "Так/Ні", "Активний/Неактивний".
2. **Списки завдань:** Позначати виконання завдань у списку.
3. **Фільтрація даних:** Використовувати прапорці для застосування або зняття фільтрів у даних.
4. **Налаштування звітів:** Вибирати параметри для побудови графіків або діаграм.
5. **Інтерактивні форми:** Створення форм для опитувань чи анкет.

3.2.4. Лічильник

Його встановлення аналогічне попереднім, але налаштування значно відрізняється тим, що необхідно встановити поточне, мінімальне, максимальне значення, а також задати крок керування і адресу комірки, де буде результат роботи лічильника.



Значення цих чисел можуть бути тільки цілим з коливанням не більше 30000. Крок зміни теж має бути цілим. Тоді, натискання на стрілку вгору чи вниз змінить число у клітинці на величину заданого кроку.

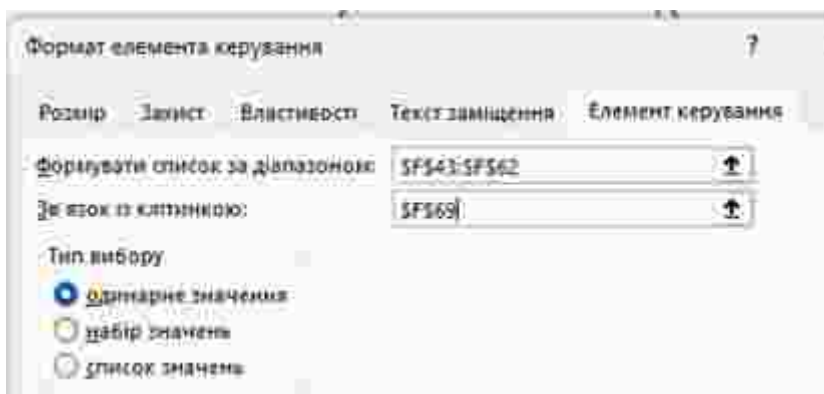


Цей елемент керування використовується при:

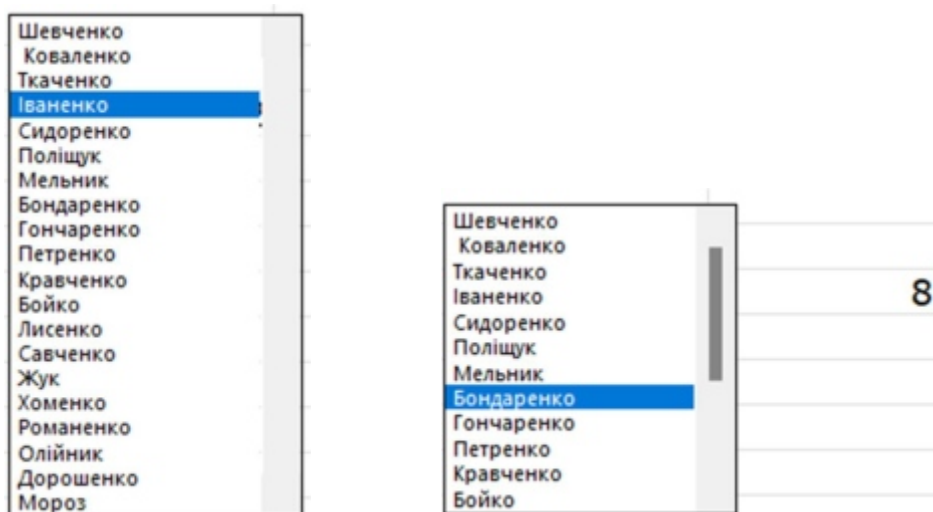
1. Для числового вводу: Замість того щоб вручну вводити значення, користувач може натискати кнопки, щоб змінювати число в потрібному діапазоні.
2. Для форм і анкет: Лічильник зручний для вибору кількості, наприклад, замовлених товарів або днів.
3. Динамічні графіки: Лічильник можна прив'язати до комірки, яка змінює значення, і графік буде автоматично оновлюватися.
4. Автоматизація розрахунків. Значення, змінене через лічильник, можна використовувати у формулах для автоматичних підрахунків.

3.2.5. Список

Цей елемент керування діє аналогічно полю зі списком. Так само необхідно вказати сам список і комірку, де буде номер обраного елемента списку.



Відміною є зображення списку в самому елементі. Причому, якщо ви розтягнете вікно в достатній мірі, список буде видний весь. Якщо вікно зменшити, то праворуч з'явиться смуга прокрутки, яка дає можливість передивитися весь список. Відмічений елемент списку позначається кольоровою смужкою.



Цей елемент керування використовується при:

1. Вибір параметрів: Зручний для вибору одного або декількох варіантів зі списку (наприклад, вибір категорії товару).
2. Фільтрація даних: Автоматизація відображення даних залежно від вибраного значення.
3. Інтерактивні форми: Для створення професійних та зручних інтерфейсів у таблицях.

3.2.6. Кнопка-перемикач

Цей елемент спрацьовує тільки один раз. Повторне натискання кнопки не викликає ніяких дій. В комірці, що пов'язана з кнопкою-перемикачем, буде 1, якщо кнопка натиснута і 0, якщо не натиснута.



Первинне положення перемикача визначається пунктом «Значення».



Excel елемент керування типу кнопка-перемикач (також відомий як Option Button або Radio Button) є частиною інтерфейсу форма елементів керування, що дозволяє користувачам вибирати одну опцію з кількох. Це дуже корисно для створення інтерактивних форм або для здійснення вибору в межах декількох варіантів.

Кнопки-перемикачі, які належать до однієї групи, дозволяють вибирати лише одну опцію з кількох. Тому вони повинні бути розміщені в межах одного групи, щоб працювати правильно (не більше ніж один елемент може бути вибраний одночасно).

3.2.7. Вікно групи

В Excel вікно групи (або Group Box) є елементом керування, який дозволяє групувати інші елементи керування, такі як кнопки-перемикачі, прапорці, текстові поля тощо. Він не має функції вибору чи вводу, але використовується для організації та візуального групування елементів керування на аркуші.

Основне призначення:

- Вікно групи дозволяє вам логічно об'єднувати кілька елементів керування в одну групу, щоб було зручніше керувати ними.
- Це корисно, коли є кілька елементів керування, наприклад, кілька кнопок-перемикачів або прапорців, і вам потрібно забезпечити, щоб ці елементи працювали разом, або просто для кращої організації інтерфейсу.

Як використовувати вікно групи в Excel:

1. Додавання вікна групи

1. Активуйте вкладку "Розробник" (якщо вона ще не активована):
 - о Перейдіть до Файл > Параметри.
 - о У розділі Налаштування стрічки (Customize Ribbon) поставте галочку біля Розробник (Developer).
2. Додавання елемента "Вікно групи":
 - о Перейдіть на вкладку Розробник.
 - о Натисніть Вставити в групі Елементи керування (Controls).
 - о У списку Елементи керування форми (Form Controls) виберіть Вікно групи (Group Box) і натисніть на робочий лист.
 - о Мишкою намалюйте область для вашого вікна групи.

2. Додавання елементів керування всередину вікна групи

- Після того як ви створили вікно групи, ви можете додавати до нього різні елементи керування, такі як:
 - о Кнопки-перемикачі (Option Buttons)
 - о Прапорці (Check Boxes)
 - о Текстові поля (Text Boxes)

- о Кнопки (Buttons)

Просто виберіть потрібний елемент керування (наприклад, кнопка-перемикач) і розмістіть його в межах вікна групи.

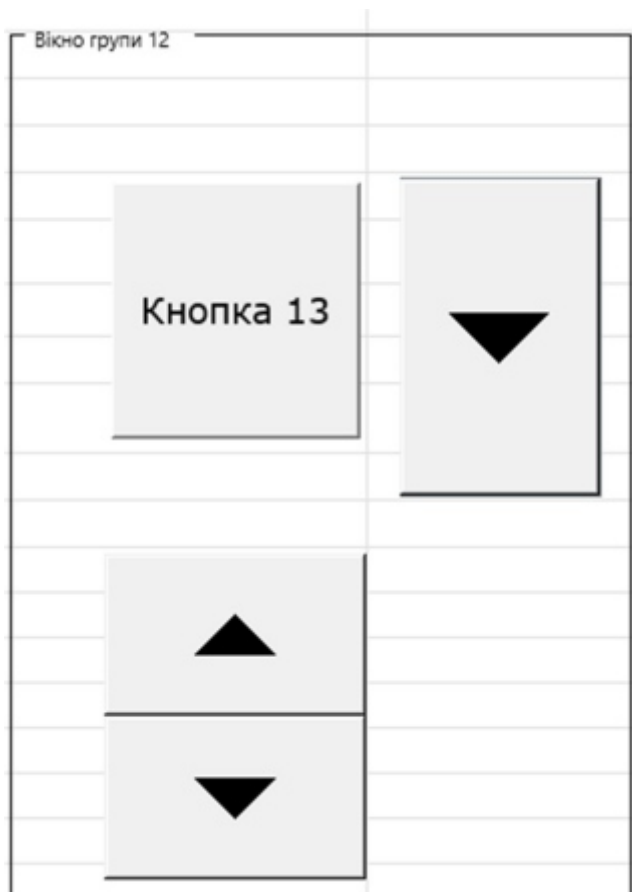
3. Налаштування властивостей вікна групи

- Щоб налаштувати вікно групи, клацніть правою кнопкою на ньому і виберіть Формат елемента керування (Format Control).
- Ви можете налаштувати різні параметри:
 - о Назва групи — можна задати текст, який буде відображатись у верхній частині вікна групи, щоб описати, що міститься в групі (наприклад, "Типи опцій" або "Налаштування звітності").
 - о Шрифт та кольори — для стилізації групи.

4. Організація елементів керування в групі

- Групування елементів: Якщо ви додаєте кілька елементів керування в групу (наприклад, кілька кнопок-перемикачів), вони автоматично стають частиною цієї групи. Це допомагає уникнути помилок, оскільки такі елементи можуть працювати разом в межах групи.

- о Наприклад, ви можете додати кілька кнопок-перемикачів в одне вікно групи, і це дозволить вам



гарантувати, що користувач може вибрати лише один варіант із кількох.

5. Приклад використання вікна групи

Припустимо, у вас є форма для вибору типу доставки (експрес-доставка, стандартна доставка, безкоштовна доставка), і ви хочете згрупувати ці варіанти разом, щоб користувач міг вибрати лише один. Ось як можна це зробити:

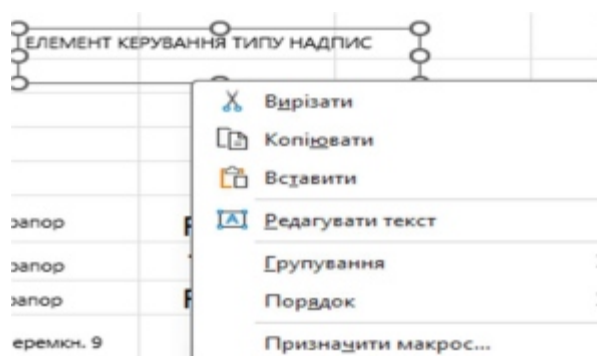
1. Створіть вікно групи і додайте до нього кілька кнопок-перемикачів (для кожного варіанту доставки).
2. У кожній кнопці-перемикачі задайте підпис, наприклад: "Експрес доставка", "Стандартна доставка", "Безкоштовна доставка".
3. Усі ці кнопки будуть належати до однієї групи і користувач зможе вибрати лише одну з них.

Переваги вікна групи:

- Організація інтерфейсу: Допомагає візуально організувати елементи керування, що робить форму зрозумілішою для користувачів.
- Управління вибором: Групуючи елементи керування, ви можете легко визначити, що користувач може вибрати лише один варіант, якщо це потрібно.
- Візуальна чіткість: Користувачі можуть легше зрозуміти, до яких варіантів належать певні елементи керування (наприклад, різні категорії опцій).

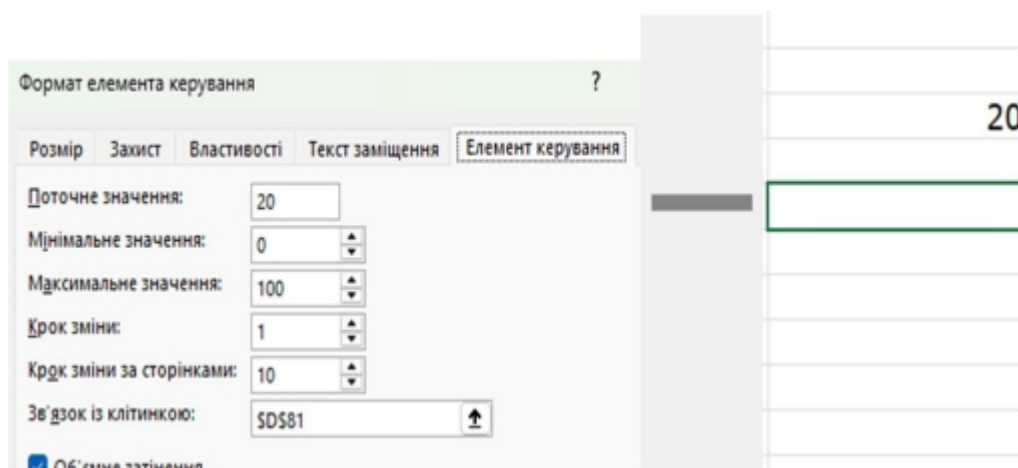
3.2.8. Надпис

Необхідний для того, щоб створити можливість вставляти макроси якийсь текст. Тому, після створення напису, треба звернутися до пункту «Призначити макрос» і зробити, наприклад макрос переходу на іншу сторінку.



3.2.9. Смуга прокручування

Схожа за своєю дією на лічильник, але має інше зображення: у вигляді смуги з повзунком, переміщення якої показує зміни з пов'язаній комірці. При переміщенні повзунка ліва кнопка мишки має бути постійно натиснута.



Використання смуги прокручування:

- Після налаштування смуги прокручування, ви можете її використовувати для зміни значення в заданому діапазоні.
- Наприклад, якщо смуга прокручування налаштована на діапазон від 1 до 100, переміщуючи повзунок, користувач змінює значення у прив'язаній клітинці.
- Це значення можна використовувати в формулах для подальших обчислень або динамічних оновлень.

Припустимо, ви хочете використовувати смугу прокручування для вибору коефіцієнта зниження для калькуляції ціни товару. Ось як це зробити:

1. Додайте смугу прокручування до аркуша та прив'яжіть її до клітинки, наприклад, A1.
2. Встановіть мінімум 0 і максимум 100, щоб значення змінювалося від 0% до 100%.
3. У клітинці B1 використовуйте формулу для обчислення нової ціни, наприклад:

$$=A2 * (1 - A1/100)$$

де A2 – це початкова ціна товару, а A1 – це значення, вибране через смугу прокручування.

4. Смугу прокручування можна стилізувати, змінивши її кольори, шрифти, розмір тощо.

5. Для цього можна використовувати налаштування Формат елемента керування (Format Control) або змінювати параметри через VBA.

Переваги використання смуги прокручування

- Інтерактивність: Користувачі можуть легко вводити значення в межах обмеженого діапазону, що зручно для динамічних форм або налаштувань.
- Простота: Смуга прокручування дозволяє швидко змінювати значення, не вводячи цифри вручну.
- Зручність: Особливо корисна для вибору числових значень, коли є певний діапазон варіантів (наприклад, для налаштувань чи визначення значень у моделях).

3.3. Графічні елементи (SmartArt, діаграми, фігури)

Графічні елементи в Excel — це елементи, які дозволяють представити дані та інформацію в більш візуальній, зрозумілій та інтуїтивно зрозумілій формі. Вони включають графіки, діаграми, фігури, малюнки та інші візуальні компоненти, які допомагають покращити сприйняття даних і забезпечити кращу комунікацію інформації.

Для чого в Excel потрібні графічні елементи:

1. Візуалізація даних

- Графіки та діаграми – це один з основних способів представлення даних в Excel. Вони допомагають зрозуміти тенденції, порівнювати значення, аналізувати взаємозв'язки між різними змінними.
 - о Гістограми – для порівняння категорій.

- o Лінійні графіки — для відображення змін значення на протязі часу.
- o Кругові діаграми — для показу часток від цілого.
- o Діаграми розсіювання — для вивчення кореляції між двома змінними.

Наприклад, якщо у вас є дані про продажі по місяцях, то графік допоможе чітко побачити, в які місяці продажі зросли чи зменшились.

2. Полегшення аналізу та прийняття рішень

- Графічні елементи дозволяють швидше зрозуміти, що відбувається з даними, без необхідності детально аналізувати числові значення.
- Наприклад, кольорові індикатори (підсвічування клітинок кольором) можуть допомогти виявити аномалії, навести акценти на важливі дані, вказати на помилки або недоліки в розрахунках.

Наприклад ви можете використовувати умовне форматування для того, щоб автоматично виділяти найвищі або найнижчі значення в таблиці різними кольорами. Це полегшить виявлення ключових точок на основі візуальних сигналів.

3. Додавання естетичності та професіоналізму

- Використання графічних елементів дозволяє зробити ваші робочі книги більш привабливими та професійними.
- Наприклад, вставлення логотипів, малюнків, фігур і підписів може підвищити презентабельність звітів чи таблиць, зробити їх більш чіткими та організованими.

Наприклад, якщо ви готуєте звіт для презентації, додавання відповідних графічних елементів може зробити інформацію зрозумілішою та більш привабливою для аудиторії.

4. Ілюстрація концепцій і ідей

- Фігури та малюнки дозволяють створювати більш детальні ілюстрації для підтримки інформації в звітах. Це корисно для створення процесів, схем або організаційних діаграм.

- Наприклад, можна вставити фігури для відображення структурних або блок-схем (наприклад, використовуючи елементи для демонстрації бізнес-процесів або етапів проекту).

Наприклад ви можете створити схему потоку для бізнес-процесів або використовувати стрілки та кола для позначення напрямків або важливих кроків у проекті.

5. Робота з даними, що часто змінюються

- Динамічні діаграми та графіки допомагають наочно бачити, як змінюються дані у реальному часі. Ви можете налаштувати графіки так, щоб вони автоматично оновлювались, коли змінюються дані.

- Це особливо корисно для відслідковування показників продуктивності, наприклад, коли потрібно переглядати зміни в розрахунках кожного дня або тижня.

Наприклад, динамічний графік, що показує зміни курсу валют або акцій в режимі реального часу, дозволяє негайно побачити тенденції.

6. Аналіз і прогнозування

- Графічні елементи допомагають прогнозувати тренди і моделювати сценарії. Наприклад, побудова графіків на основі історичних даних дозволяє зробити припущення щодо майбутніх змін.

- Графіки і діаграми також використовуються для відображення результатів аналізу даних, що можуть бути частиною прогнозних моделей або бізнес-аналітики.

Наприклад ви можете побудувати графік на основі даних про попит і пропозицію для прогнозування обсягів виробництва в майбутньому.

7. Покращення інтерфейсу для користувача

- Графічні елементи можна використовувати для покращення інтерфейсу користувача в Excel. Це включає створення кнопок, повзунків, смуг прокручування для інтерактивних форм та таблиць.

- Це дозволяє користувачам взаємодіяти з даними без необхідності вносити зміни вручну, що покращує досвід користувача.

Наприклад, створення динамічної форми для введення даних за допомогою кнопок і графічних елементів дозволить користувачам швидше вводити і змінювати параметри, що автоматично оновлюються.

Типи графічних елементів, які використовуються в Excel:

1. Графіки та діаграми:
 - o Лінійні, стовпчасті, кругові, гістограми, графіки розсіювання, областеві діаграми і т.д.
2. Малюнки та зображення:
 - o Вставка зображень у вигляді фотографій, скріншотів або логотипів.
3. Фігури та об'єкти:
 - o Квадрати, кола, стрілки, лінії, прямокутники для створення схем, блок-схем, організаційних діаграм.
4. Текстові елементи:
 - o Вставка текстових полів, заголовків, підписів до графіків, а також вільний текст для пояснень.
5. Умовне форматування:
 - o Використання кольорів для виділення важливих даних або показників.
6. Індикатори та шкали:
 - o Лінійки, смуги прогресу, що відображають рівень виконання або досягнення цілей.

Графічні елементи в Excel є важливими інструментами для візуалізації, аналізу та подання даних. Вони покращують сприйняття інформації, дозволяють швидко розпізнати тренди та аномалії, полегшують процес прийняття рішень і роблять робочі книги більш привабливими та зрозумілими для користувачів. Графічні елементи значно підвищують ефективність роботи з даними, особливо коли йдеться про великі обсяги або складні моделі.

3.3.1. Картинки та фігури як елемент інтерактивного вікна

Для оформлення інтерактивного вікна можна використовувати різні види зображень, таких як картинки, піктограми та фігури.

Всі вони розташовуються за одним принципом: те зображення, що з'явилося пізніше – лежить зверху. Це ж правило діє і на будь-які інтерактивні елементи.

На рис. 3.1 показані всі три елементи таких зображень.

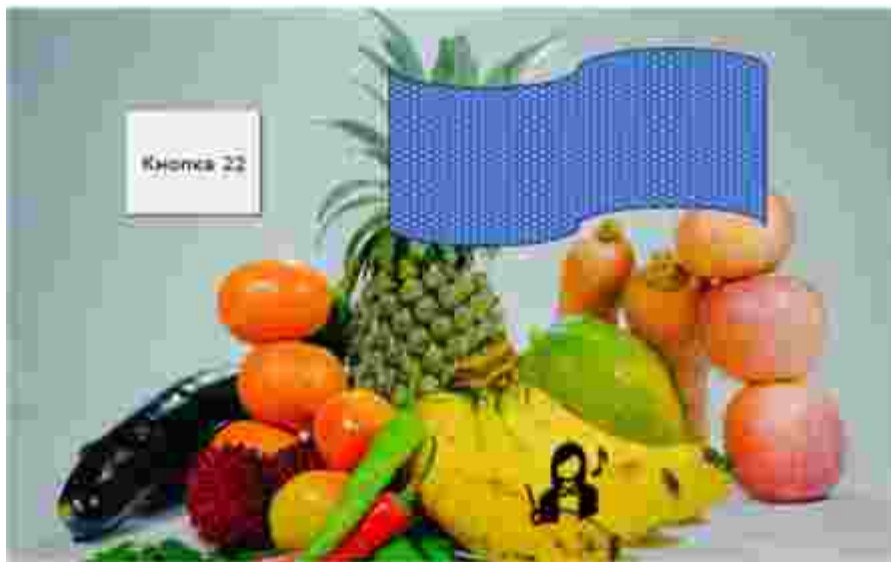
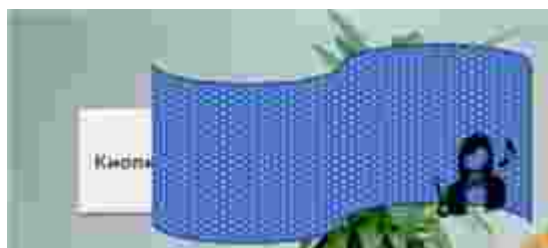


Рис. 3.1. На рисунок із фруктами накладена у такому порядку: кнопка, фігура у синю клітинку та піктограма, що зображає диригента

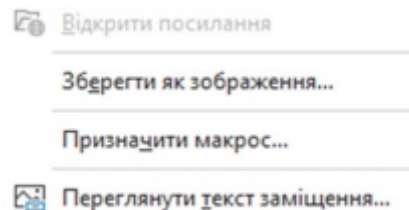
Якщо тепер пересунути накладені фігури одна на одну, вони будуть перекривати одна одну у порядку, зворотному моменту накладення. А саме: найнижчий рівень – картинка з фруктами, наступний, – кнопка, далі – фігура, а піктограма буде поверх усіх.



Якщо клацнути правою кнопкою мишки на будь який елемент зображення, з'явиться рамка, що дозволяє змінити габарити картинки.



Ця рамка також діє як елемент контекстного меню, серед пунктів якого є пункт «Призначити макрос». Клацнувши по цьому пункту ви отримаєте доступ до редактора VBA і таким чином зробите це зображення інтерактивним елементом керування.



Наприклад, за елементом фігури «Стрілка вправо з вирізом»  ви можете створити макрос для переходу на якусь клітинку у цьому ж листі Excel.

Це буває корисним, коли ваш розрахунок досить великий, щоб розміститися в одному вікні і щоб його роздивитися, вам необхідно переміщатися в межах одного листа на декілька екранів. Така стрілка полегшить перміщення.

Створення макросу для переходу до клітинки:

1. Після вставлення кнопки з'явиться діалогове вікно Призначення макросу (Assign Macro).
2. Натисніть Створити (Create), щоб відкрити редактор VBA.
3. В редакторі VBA напишіть код для переходу до певної клітинки.

Наприклад:

Sub GoToCell()

' Перехід до клітинки A1

Application.Goto Range("A1")

End Sub

- Application.Goto Range("A1") – цей рядок коду виконує перехід до клітинки A1. Ви можете змінити "A1" на будь-яку іншу адресу клітинки.
4. Закрийте редактор VBA, натиснувши X або вибравши File > Close and Return to Microsoft Excel.
 5. Тепер, коли ви натискаєте на картинку, курсор автоматично переміщується до клітинки A1 або до будь-якої іншої клітинки, яку ви вказали в макросі.

3.3.2. SmartArt

SmartArt в Excel – це інструмент для створення графічних схем, діаграм, організаційних структур та інших типів візуальних елементів для покращення сприйняття та представлення даних. SmartArt дозволяє користувачам додавати стилізовані елементи для наочності та кращого відображення інформації, що полегшує її розуміння.

SmartArt в Excel – це потужний інструмент для створення графічних представлень даних. Він дозволяє покращити вигляд таблиць, документів і доповідей, а також забезпечує простоту в створенні організаційних схем, процесів, діаграм та інших візуальних елементів.

Ось основні особливості SmartArt в Excel:

1. Графічні шаблони: SmartArt містить багато готових шаблонів, які можна налаштувати для різних видів візуалізації даних. Наприклад, це можуть бути:

- o Організаційні схеми (для показу структури організації або процесу).
- o Циклічні процеси (для зображення етапів циклу чи повторюваних процесів).
- o Ієрархічні структури (для подання відносин між елементами).
- o Підсумки та списки (для структурованих даних, таких як перелік пунктів).
- o Підтримка зв'язків (для відображення зв'язків між об'єктами).

2. Легкість у налаштуванні:

o Після вибору шаблону SmartArt можна редагувати його елементи: змінювати кольори, розміри, шрифти, додавати або видаляти елементи, а також налаштовувати анімацію або ефекти.

o Користувач може додавати текст до кожного елемента, змінювати форми, додавати нові рівні або етапи до схем.

3. Інтерактивність:

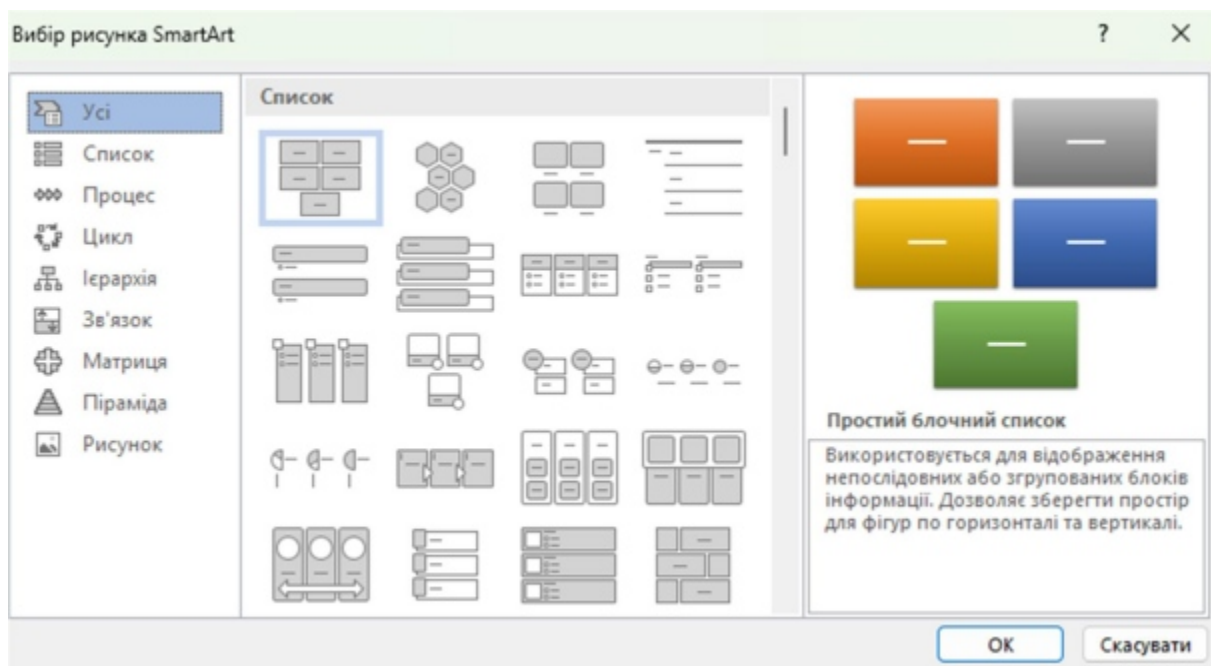
о Ви можете змінювати структуру SmartArt в реальному часі, перетягуючи елементи, щоб змінити порядок, або додавати нові форми та блоки.

4. Можливості налаштування:

- о Можна змінювати кольорову схему, шрифти, стиль і товщину ліній.
- о Підтримує функції зміни розміру окремих блоків або об'єктів для більш гнучкої роботи.

Як додати SmartArt в Excel:

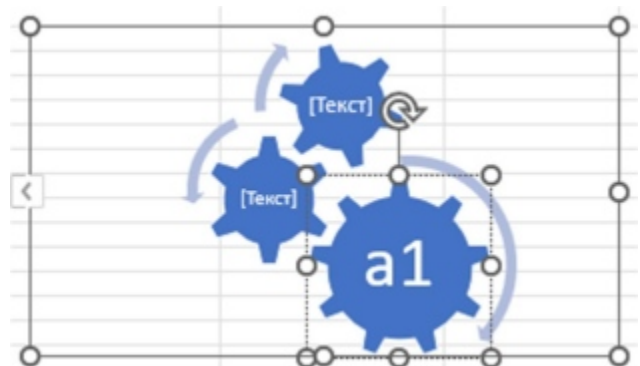
1. Перейдіть на вкладку "Вставка" на стрічці.
2. У групі "Ілюстрації" виберіть "SmartArt".



3. Виберіть один із доступних шаблонів у вікні, що з'явиться, та натисніть "ОК".

4. Введіть текст або додайте елементи до SmartArt через панель "Текст" (або безпосередньо у графічні блоки).

5. Кожен елемент структури може бути переміщений або йому можна



змінити розмір чи повернути на довільний кут, якщо по ньому клацнути правою кнопкою мишки.

6. Елемент SmartArt не може мати гіперпосилання, але вся структура цих елементів може стати інтерактивною, якщо клацнути правою кнопкою мишки по зовнішній рамці.

Приклад використання SmartArt:

1. Якщо ви хочете показати організаційну структуру компанії, виберіть відповідний шаблон SmartArt і заповніть його даними про співробітників і відділи.

2. Якщо вам потрібно показати етапи процесу, виберіть циклічну схему, яка допоможе візуалізувати кроки або послідовність дій.

Переваги використання SmartArt:

- Полегшує створення візуальних схем та графіків.
- Покращує розуміння та сприйняття даних, роблячи їх більш зрозумілими та доступними.
- Допомагає представити складні дані в простій і зручній для сприйняття формі.

3.3.3. Рухомі графіки

Щоб створити рухомий графік в Excel, можна використовувати анімовані графіки, які змінюються з часом, відображаючи зміни даних. Це можна зробити за допомогою VBA (Visual Basic for Applications) для динамічної зміни значень або шляхом оновлення графіку через зміни даних.

Ось покрокова інструкція для створення простого рухомого графіка, використовуючи VBA:

Кроки для створення рухомого графіка:

1. Створіть графік у Excel:

- о Введіть дані, які ви хочете відобразити на графіку. Наприклад, у вас є дані по датах в стовпці А і значеннях у стовпці В.

- о Виберіть дані, а потім перейдіть до вкладки "Вставка" → "Графік" і виберіть тип графіку (наприклад, лінійний графік або стовпчиковий графік).
2. Додайте VBA код для анімації: Для створення анімації графіка вам потрібно використовувати VBA. Цей код буде змінювати діапазон даних на графіку через певні інтервали часу, створюючи ефект руху.
 3. Створіть іменовані клітинки такими іменами в Excel: StartDay, NumDays та Increment.
 4. Порядок іменування клітинок розглянемо на прикладі StartDay:
 - Виберіть клітинку на аркуші, яку ви хочете використати як StartDay.
 - Перейдіть на вкладку "Формули" в Excel.
 - В групі "Іменовані діапазони" натисніть "Визначити ім'я".
 - У вікні, що з'явиться, введіть ім'я StartDay та натисніть ОК
 5. У комірку, іменовану StartDay, запишіть першу дату, з якої починається графік, у комірку NumDays – загальне число точок графіку, у Increment – число точок, які будуть показуватися на графіку водночас.
 6. Поіменуйте діапазон значень X, наприклад Date, та діапазони значень Y. Наприклад, якщо ви виводите два графіка водночас, то ці діапазони треба поіменувати різними іменами, наприклад ProdA та ProdB.
 7. Замість конкретних адрес клітинок для X та Y, потрібно скористатися функцією

OFFSET(посилання;рядки;стовпці;[висота];[ширина]).

Для нашого прикладу ця функція має вигляд, однаковий для всіх трьох масивів даних.

=OFFSET(Лист1!\$C\$1;StartDay;0;NumDays;1)

Тут Лист1!\$C\$1 означає першу клітинку, з якої починається стовпець даних для ProdB, який містить заголовок таблиці.

8. Тоді у диспетчері імен з'явиться їх опис

Диспетчер імен				
Створити... Редагувати... Видалити				
Ім'я	Значення	Посилання	Область	Прим
Date	{...}	=OFFSET(Лист1!\$A...	Робоча ...	
Increment	30	=Лист1!\$F\$3	Робоча ...	
NumDays	1000	=Лист1!\$F\$2	Робоча ...	
ProdA	{...}	=OFFSET(Лист1!\$B...	Робоча ...	
ProdB	{...}	=OFFSET(Лист1!\$C...	Робоча ...	
StartDay	4 201	=Лист1!\$F\$1	Робоча ...	

9. Напишіть VBA код для анімації: Ось приклад коду, який дозволяє анімувати графік, оновлюючи його дані:

Sub AnimateChart()

Dim StartVal As Long, r As Long

If AnimationInProgress Then

AnimationInProgress = False

End

End If

AnimationInProgress = True

StartVal = Range("StartDay")

For r = StartVal To 5219 - Range("NumDays") Step Range("Increment")

Range("StartDay") = r

DoEvents

Next r

AnimationInProgress = False

End Sub

StartDay

Пояснення коду:

1. Sub AnimateChart()

- Це заголовок макроса, який визначає назву підпрограми. Sub означає, що це підпрограма (макрос), і AnimateChart — це її ім'я. Коли ви запускаєте макрос, ця підпрограма виконується.

2. Dim StartVal As Long, r As Long

- Dim — це оператор, який оголошує змінні у VBA.
- StartVal As Long — змінна StartVal визначається як ціле число великої величини (тип Long). Ця змінна використовується для збереження початкового значення для стартового дня анімації.
- r As Long — змінна r також визначена як ціле число, яке використовується як лічильник у циклі.

3. If AnimationInProgress Then

- Це умовний оператор, що перевіряє, чи є анімація в процесі виконання.
- AnimationInProgress — це змінна, яка зберігає інформацію про те, чи триває анімація. Якщо анімація вже йде, цей код зупинить поточну анімацію (змінюючи значення AnimationInProgress на False).

4. AnimationInProgress = False

- Якщо анімація вже запущена, то змінна AnimationInProgress встановлюється в False (тобто, анімація завершується).

5. End

- Це команда для завершення виконання макроса, якщо анімація вже триває. Вона припиняє подальше виконання коду і виходить із підпрограми.

6. AnimationInProgress = True

- Якщо анімація не була запущена раніше, то змінна AnimationInProgress встановлюється в True, що означає, що анімація починається.

7. StartVal = Range("StartDay")

- Це присвоєння значення змінній StartVal. Значення StartDay береться із комірки на аркуші Excel (це може бути діапазон, з якого ви отримуєте початкову точку для анімації).

8. For r = StartVal To 5219 - Range("NumDays") Step Range("Increment")

- For — початок циклу, який ітераційно змінює значення змінної `r` від початкового значення `StartVal` до заданої межі.
- `StartVal` — початкова точка, значення якої збережене в змінній.
 - `5219 - Range("NumDays")` – кінцева точка циклу. Вона обчислюється як 5219 мінус значення з клітинки `NumDays` (це, ймовірно, кількість днів, на яку потрібно зменшити кінцеву межу для анімації).
 - `Step Range("Increment")` – крок ітерацій. Крок визначається значенням клітинки `Increment`. Це значення додається до поточного значення `r` на кожній ітерації, що дозволяє контролювати швидкість анімації.

9. `Range("StartDay") = r`

- Це присвоєння значення змінної `r` клітинці `StartDay` на аркуші. Це значення використовується для оновлення даних на графіку чи в іншій частині аркуша в процесі анімації. Таким чином, значення в клітинці `StartDay` змінюється на кожній ітерації циклу.

10. `DoEvents`

- Ця команда дозволяє Excel обробляти події (наприклад, оновлення інтерфейсу або реагування на введення користувача) під час виконання коду. Це важливо для анімації, оскільки без `DoEvents` Excel може не оновлювати графічний інтерфейс під час виконання довгих операцій.

11. `Next r`

- Це кінець циклу `For`. Коли програма доходить до цього рядка, вона повертається до початку циклу і повторює його до тих пір, поки значення `r` не досягне кінцевої точки.

12. `AnimationInProgress = False`

- Після завершення анімації значення змінної `AnimationInProgress` знову встановлюється в `False`, щоб позначити, що анімація завершена, і в подальшому можна буде запустити нову анімацію.

На рис. 3.2 подано зображення вікна Excel з кнопкою, що запускає графік таблицею даних і описом розміру таблиць.



Рис. 3.2. Вікно з налаштованим рухомим графіком

Щоб прокручувати графік кілька разів за допомогою вашого VBA-коду, вам потрібно додати можливість повторного запуску анімації. Проблема, яку ви описуєте, полягає в тому, що після завершення анімації змінна StartDay досягає кінця і більше не оновлюється, тому анімація припиняється. Для того, щоб анімація могла повторюватися кілька разів, вам потрібно додати логіку для повторного запуску анімації після досягнення кінця.

Ось як можна змінити код, щоб анімація повторювалася:

1. Замість того, щоб просто закінчувати анімацію після досягнення кінця, додайте перевірку і скидання значень так, щоб анімація починалась знову, коли досягнуто кінця.

2. Додайте цикл для безперервного виконання анімації.

Оновлений код:

```
Dim AnimationInProgress As Boolean  
Sub AnimateChart()
```

```

Dim StartVal As Long, r As Long
If AnimationInProgress Then
    AnimationInProgress = False
End
End If
AnimationInProgress = True
StartVal = Range("StartDay")
' Повторне прокручування анімації
Do
    For r = StartVal To 5219 - Range("NumDays") Step
Range("Increment")
        Range("StartDay") = r
        DoEvents
    Next r
    ' Після завершення анімації, починаємо знову
    StartVal = Range("StartDay") ' Скидаємо стартову
точку анімації
Loop While AnimationInProgress ' Продовжуємо
анімацію, якщо вона не була зупинена
End Sub

```

Пояснення змін:

1. Цикл Do While: Цей цикл дозволяє запускати анімацію безперервно, поки змінна AnimationInProgress залишатиметься True. Після досягнення кінця діапазону (встановленого в коді через 5219), програма продовжить виконання знову, використовуючи нові значення.
2. Оновлення StartVal: Кожного разу після завершення прокручування анімації значення StartVal оновлюється і встановлюється знову, щоб анімація починалася з актуальної точки.

3. Постійна перевірка умови AnimationInProgress: Це дозволяє зупинити анімацію вручну за допомогою зміни цієї змінної на False, якщо потрібно припинити анімацію в будь-який момент.

Окремо можна додати кнопку для зупинки анімації:

Sub StopAnimation()

AnimationInProgress = False

End Sub

Тепер ваша анімація буде повторюватися, поки ви не зупините її вручну.

3.4. Індивідуальне завдання №3.

Вивчення меню Розробник

Завдання: Вивчити операції зі встановленням інтерактивних елементів Excel.

Порядок виконання: використовуються пункт меню Excel «Розробник».

Методичні вказівки: 1) Відкрити пункт меню «Розробник».

- 2) Для наступного переліку списків створіть кнопки, прапорці та перемикачі:
 - о Анкета для співробітників: Створіть список посад: Менеджер, Аналітик, Програміст.
 - о У таблиці проекту створіть список статусів: "На розгляді", "В роботі", "Завершено".
 - о Створіть список категорій товарів: Електроніка, Одяг, Канцелярія.
 - о Створіть список типів витрат: Оренда, Зарплата, Маркетинг.
- 3) Додайте графічні елементи для яких створіть макроси переходу на іншу сторінку та на іншу комірку цього ж листа.
- 4) З сайту Держстату візьміть довільне значення якогось параметру за весь період по місяцях і створіть рухомий графік, так, щоб у вікні графіку водночас було видно 12 точок даних.
- 5) У звіті наведіть весь порядок роботи з демонстрацією активних вікон і поясненнями до них.

Контрольні запитання

1. Як створити список, що фільтрується, у Excel?
 2. Які можливості надає фільтрація даних у Excel?
 3. Які типи фільтрів можна застосовувати до таблиці?
 4. Як застосувати фільтр за умовами, щоб вибрати лише певні значення з великого списку?
 5. Як зберегти налаштування фільтрації в Excel після перезавантаження файлу?
 6. Як використовувати складні фільтри для числових та текстових значень?
 7. Як створити кнопку для запуску макросу в Excel, і які можливості вона дає?
 8. У чому відмінність між кнопкою і кнопкою-перемикачем?
 9. Як використовувати прапорець для вибору опцій в Excel? Як це може бути корисно в практичних завданнях?
 10. Як налаштувати лічильник в Excel, що збільшується або зменшується при натисканні кнопки?
 11. Що таке вікно групи, і як воно може бути корисним для організації елементів керування?
 12. Як створити список із декількома опціями для вибору в Excel?
 13. Як змінюється поведінка Excel при використанні кнопки-перемикача для вибору між кількома варіантами?
3. Графічні елементи:
14. Як додати SmartArt у Excel і які типи схем доступні для використання?
 15. Як використовувати фігури та картинки як елементи інтерактивного вікна для покращення презентації даних?

16. Як створити рухомий графік в Excel і які можливості для цього надає VBA?
17. Які типи графічних елементів можна вставити в Excel для покращення візуалізації даних?
18. Як змінюються функції Excel при додаванні графічних елементів (фігур, картинок, SmartArt) до листа?
19. Як створити інтерактивний діаграму, де дані автоматично оновлюються при зміні значень?
20. Як можна використовувати картинку чи фігуру для відображення динамічних даних в Excel?
21. Як змінити стиль та колір елементів SmartArt для кращої візуалізації?
22. Як налаштувати фільтрацію в списках за допомогою елементів керування, таких як прапорці та випадючі списки?
23. Як використовувати кнопки та перемикачі для автоматизації простих завдань у Excel?

Вивчивши матеріали цього розділу студенти навчилися встановлювати кнопки, смуги прокрутки, використовувати зображення як інтерактивні елементи, створювати рухомі графіки.

Розділ 4.

АВТОМАТИЗИЦІЯ З VBA

*Вивчивши матеріали цього студенти
узнають основні принципи мови VBA.*

VBA (Visual Basic for Applications) – це мова програмування, що була створена для автоматизації завдань і розширення функціональності в програмних продуктах Microsoft, таких як Excel, Word, Access та інших. VBA дозволяє користувачам писати скрипти та макроси для автоматизації рутинних завдань, налаштування інтерфейсу користувача та створення більш складних функцій у програмах Microsoft.

VBA була створена як зручна, доступна для користувачів мова програмування для автоматизації та розширення функцій додатків Microsoft Office. Вона дозволяє користувачам Excel автоматизувати рутинні задачі, створювати власні функції та інтерфейси, а також інтегрувати дані між різними програмами в межах Microsoft Office. VBA стала стандартним інструментом для користувачів Excel, що забезпечує їм гнучкість і потужність у роботі з таблицями та іншими даними.

Visual Basic – це мова програмування, створена Біллом Гейтсом і Полом Алленом на основі мови BASIC, яка була популярною в 1960-х роках. Вона дозволяла програмістам створювати прості додатки для операційних систем. У середині 1980-х компанія Microsoft почала розробляти розширену версію цієї мови під назвою Visual Basic (VB).

Чому VBA використовується в Excel?

1. Автоматизація рутинних завдань:
 - о Однією з основних причин використання VBA в Excel є можливість автоматизувати повторювані або складні завдання. Завдяки макросам і скриптам, створеним на VBA, можна зекономити час на виконання

простих або багаторазових операцій, таких як обчислення, фільтрація, форматування даних тощо.

2. Розширення функціональності Excel:

- о Excel має багатий набір стандартних функцій, але завдяки VBA можна створювати додаткові функції (UDF – User Defined Functions), що дозволяють реалізовувати складні розрахунки або специфічні бізнес-логіки, які не можна виконати стандартними інструментами Excel.

3. Інтерфейс користувача:

- о VBA дає можливість створювати інтерфейси користувача для взаємодії з користувачами, використовуючи форми (UserForms), кнопки, поля введення та інші елементи керування. Це робить роботу з Excel більш інтерактивною і зручною.

4. Інтеграція з іншими додатками Microsoft:

- о VBA дозволяє інтегрувати Excel з іншими додатками Microsoft, такими як Word, Outlook, Access та PowerPoint. Наприклад, можна автоматизувати створення звітів в Word на основі даних з Excel або автоматично надсилати електронні листи через Outlook.

5. Легка доступність для користувачів:

- о VBA дозволяє навіть недосвідченим користувачам, які не мають глибоких знань у програмуванні, створювати прості макроси для автоматизації задач. Це робить VBA доступним інструментом для широкого кола користувачів.

6. Гнучкість та потужність:

- о VBA підтримує об'єктно-орієнтоване програмування і дозволяє працювати з багатьма складними структурами даних, зокрема масивами та об'єктами. Це робить її потужним інструментом для вирішення навіть складних завдань в Excel.

7. Сумісність з іншими мовами програмування:

- о VBA має можливість викликати функції з інших мов програмування (наприклад, C++, Python) або навіть працювати з бібліотеками через COM-об'єкти, що робить її потужним інструментом для інтеграції з іншими технологіями.

4.1. Введення у VBA: основні принципи

VBA (Visual Basic for Applications) – це мова програмування, що використовується для автоматизації задач, створення макросів і розширення функціональності в додатках Microsoft Office, таких як **Excel**, **Word**, **Access** тощо. VBA дозволяє програмістам і користувачам створювати спеціалізовані інструменти та виконувати рутинні завдання за допомогою скриптів.

Основні принципи роботи з VBA в Excel:

VBA базується на мові **Visual Basic**, але орієнтована на автоматизацію та інтеграцію з додатками Office. Вона дозволяє виконувати завдання, які складно чи неможливо реалізувати стандартними засобами Excel.

Основні елементи VBA:

- **Модулі** — це контейнери для коду. У VBA код пишеться в **модулях**, які можна створювати для зручності.
- **Процедури (Sub)** — це окремі частини коду, що виконуються при їх виклику. Вони починаються з **Sub** і закінчуються **End Sub**.
- **Функції (Function)** — це блоки коду, які повертають значення. Функція визначається через **Function** і **End Function**.

Приклад процедури:

```
Sub HelloWorld()  
    MsgBox "Привіт, світ!"  
End Sub
```

Основні принципи:

1. Об'єктно-орієнтований підхід:

- o VBA працює з об'єктами та їх властивостями. Наприклад, в Excel об'єктами є **клітинки, діапазони, аркуші** тощо.
- o Кожен об'єкт має **властивості** (наприклад, значення клітинки, форматування) та **методи** (наприклад, методи для зміни значення або форматування клітинки).

2. Перемінні та типи даних:

- o У VBA можна оголошувати змінні для зберігання значень. Змінні можуть мати різні типи даних, наприклад:
 - **Integer** – ціле число.
 - **String** – текст.
 - **Double** – дійсні числа.
 - **Boolean** – логічні значення **True/False**.
- o Змінні оголошуються через ключове слово **Dim**:

Dim x As Integer

x = 5

3. Контроль виконання (умови та цикли):

- o Для управління виконанням коду використовуються конструкції умов і циклів:

If...Then для умов:

If x > 10 Then

MsgBox "Більше ніж 10"

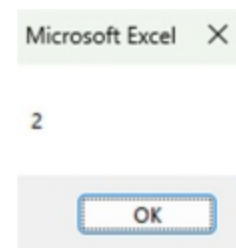
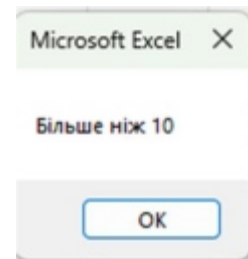
End If

For для циклів:

For i = 1 To 5

MsgBox i

Next i



4. Обробка помилок:

- o Для керування помилками застосовуються конструкції **On Error**, що дозволяє управляти сценаріями, коли в коді виникають помилки:
- o **On Error Resume Next**

2. Макроси і автоматизація

- **Макроси** – це записи послідовностей дій користувача, які можна автоматизувати за допомогою VBA. Вони зазвичай використовуються для виконання простих повторюваних операцій.

Щоб створити макрос:

- Перейдіть на вкладку **Розробник (Developer)**.
 - Натисніть **Записати макрос**.
 - Виконуйте дії, які хочете автоматизувати, а потім зупиніть запис.
 - Після цього можна редагувати записаний код у редакторі VBA, щоб змінити чи доповнити його.
- **Використання об'єктів Excel:**

У VBA для доступу до елементів Excel, таких як клітинки, діапазони, аркуші, використовуються об'єкти, наприклад:

Sheets("Лист1").Range("A1").Value = 10

У цьому прикладі у комірку записується число 10.

У Excel, кожен елемент (наприклад, клітинка, рядок, стовпець, аркуш) є об'єктом з певними властивостями і методами. Ось основні об'єкти:

- **Workbook** – книга Excel.
- **Worksheet** – аркуш Excel.
- **Range** – діапазон клітинок.
- **Cell** – окрема клітинка.

Взаємодія з об'єктами виконується через їхні властивості і методи:

Range("A1").Value = "Текст" ' Встановлює значення в клітинку A1

Sheets("Лист1").Activate ' Активує аркуш "Лист1"

VBA дозволяє створювати користувацькі форми та елементи управління (кнопки, текстові поля, прапорці і т. д.). Це дає змогу створювати більш інтуїтивно зрозумілі інтерфейси для взаємодії з користувачем.

- **UserForms** – форми для створення інтерфейсу користувача, в яких можна додавати кнопки, поля введення, списки, тощо.

5. Події

VBA може реагувати на події, які відбуваються в Excel. Наприклад, можна налаштувати події для обробки змін у клітинках або при натисканні кнопок.

- Подія зміни значення клітинки:

```
Private Sub Worksheet_Change(ByVal Target As Range)
```

```
MsgBox "Значення змінилося!"
```

```
End Sub
```

- Подія натискання кнопки:

```
Private Sub CommandButton1_Click()
```

```
MsgBox "Кнопка натиснута!"
```

```
End Sub
```

6. Основні об'єкти та їх властивості

- **Workbooks** – колекція всіх відкритих книг.
- **Worksheets** – колекція всіх аркушів у книзі.
- **Range** – об'єкт, що представляє діапазон клітинок.

Приклад доступу до аркуша та клітинки, в яку записується текст:

```
Dim ws As Worksheet
```

```
Set ws = ThisWorkbook.Sheets("Лист1")
```

```
ws.Range("A1").Value = "Привіт!"
```

У VBA (Visual Basic for Applications) відступи не є обов'язковими для правильного виконання коду. Однак, використання відступів є важливим для покращення читабельності коду і для того, щоб інші програмісти (або ви самі пізніше) могли легше розуміти структуру вашої програми.

Чому використовувати відступи:

1. **Читабельність:** Відступи допомагають зрозуміти, які команди входять до складу певного блоку, наприклад, до умовного оператора If...Then...Else або циклу For...Next. Це полегшує розуміння структури і логіки коду.
2. **Організація:** Відступи дозволяють чітко бачити, де починається і закінчується певний блок коду. Це особливо корисно в складних програмах із багатьма вкладеними умовами або циклами.

3. Стандарт у програмуванні: Використання відступів є хорошою практикою в програмуванні, і більшість стилів кодування рекомендують їх.

Приклад:

```
Sub Example()  
    Dim x As Integer  
    x = 10  
    If x > 5 Then  
        MsgBox "X більше за 5"  
    Else  
        MsgBox "X не більше за 5"  
    End If  
End Sub
```

У цьому прикладі:

- Відступи допомагають зрозуміти, що MsgBox знаходиться всередині блоку If...Else.
- Використання відступів робить код більш структурованим і легким для сприйняття.

Важливо:

- VBA не вимагає відступів для коректного виконання коду. Тобто код працюватиме навіть без них:

```
Sub Example()  
    Dim x As Integer  
    x = 10  
    If x > 5 Then MsgBox "X більше за 5" Else MsgBox "X не  
    більше за 5" End If  
End Sub
```

Такий код теж буде працювати, однак він буде важчим для читання і підтримки.

У VBA можна записати кілька команд в один рядок, якщо вони розділені : (крапка з комою). Це дозволяє зменшити кількість рядків коду, хоча не є

рекомендованою практикою для складних операцій, оскільки це знижує читабельність.

Приклад:

```
Sub Example()
```

```
Dim x As Integer, y As Integer
```

```
x = 5: y = 10
```

```
MsgBox "X: " & x & " Y: " & y
```

```
End Sub
```

У цьому прикладі:

- Дві команди – **x = 5** та **y = 10** – записані в одному рядку, розділені двокрапкою **:**.

Важливі моменти:

1. **Читабельність:** Використання кількох команд в одному рядку може зробити код менш зрозумілим, тому краще використовувати цей підхід лише в простих випадках, де команди є очевидними.
2. **Типи команд:** Зазвичай це підходить для простих команд присвоєння або для коротких умовних перевірок, але не рекомендується для складних логічних конструкцій чи викликів функцій, оскільки це може зробити код важким для розуміння.

Не рекомендовано:

```
If x > 5 Then MsgBox "X більше 5": y = 100
```

У цьому прикладі краще використовувати окремі рядки для кожної команди, щоб код був більш структурованим і зрозумілим.

- Ви можете записати кілька команд в одному рядку, використовуючи двокрапку **:**.
- Це дозволяє зекономити простір, але зазвичай не рекомендується, оскільки погіршується читабельність коду.

4.2. Запис та редагування макросів

Запис та редагування макросів у Excel

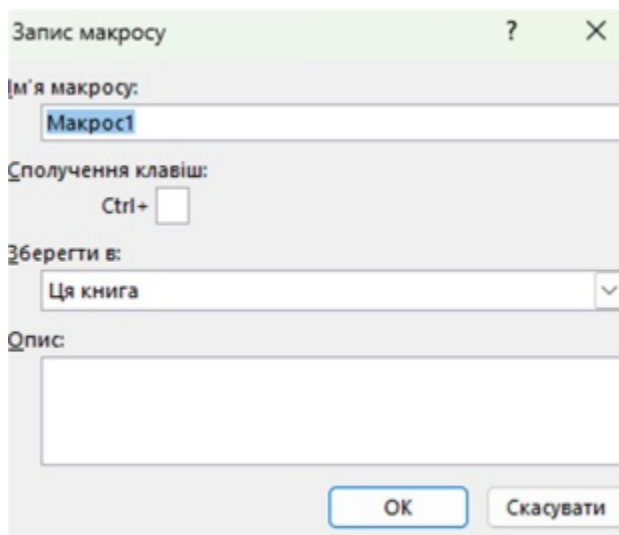
Макроси в Excel – це послідовності команд, які можна записати для автоматизації повторюваних завдань. Макроси в Excel записуються на мові програмування VBA (Visual Basic for Applications), але ви також можете використовувати вбудовану функцію для запису макросів без необхідності писати код вручну.

1. Запис макросу

Запис макросу дозволяє автоматично фіксувати ваші дії в Excel, перетворюючи їх у код VBA. Це зручний спосіб автоматизації завдань без потреби знати мову програмування.

Кроки для запису макросу:

1. **Відкрийте Excel.**
2. Перейдіть до вкладки **Розробник (Developer)** на стрічці і обрати пункт «Записати макрос».
3. Виберіть **Записати макрос (Record Macro)**. З'явиться діалогове вікно.
4. У діалоговому вікні задайте:
 - о **Ім'я макросу** – надайте макросу ім'я (наприклад, MyMacro).
 - о **Сполучення клавіш (Keyboard Shortcut)** – при необхідності призначте сполучення клавіш для запуску макросу. Наприклад
 - о **Місце збереження:**
 - **Ця книга (This Workbook)** — макрос буде збережений тільки в поточній книзі.



- **Нова книга (New Workbook)** — макрос буде збережений в новому файлі.
 - **Особистий файл макросів (Personal Macro Workbook)** — макрос буде доступний у всіх файлах Excel.
5. Натисніть **ОК**, і Excel почне записувати ваші дії.
 6. Виконайте дії, які ви хочете автоматизувати (наприклад, форматування клітинок, створення діаграм або сортування даних).
 7. Коли всі дії будуть виконані, натисніть **Зупинити запис (Stop Recording)** на вкладці **Розробник (Developer)**.

Тепер ваш макрос збережений, і ви можете його використовувати в заданих межах (аркуш або книга).

2. Редагування макросу (VBA)

Макроси, які ви записуєте, автоматично генерують код на VBA, який можна редагувати для удосконалення або зміни. Щоб редагувати макроси, потрібно використовувати редактор VBA.

Кроки для редагування макросу:

1. Перейдіть на вкладку **Розробник (Developer)**.
2. Натисніть **Visual Basic** або натисніть **Alt + F11**, щоб відкрити редактор VBA.
3. У редакторі VBA у лівій частині вікна (вікно **Project Explorer**) знайдіть проект вашої книги Excel (наприклад, **VBAProject (YourWorkbookName)**).
4. Розгорніть розділ **Modules**, щоб побачити всі модулі з макросами.
5. Двічі клацніть на потрібний модуль (наприклад, **Module1**), і з'явиться код записаного макросу.
6. Тепер ви можете змінити код на ваш розсуд. Наприклад, ви можете додавати умовні оператори, цикли, або змінювати параметри макросу.

Приклад простого записаного макросу:

Sub MyMacro()

Range("A1").Value = "Hello, World!" ' Записуємо текст у клітинку A1

Range("A1").Font.Bold = True ' Зробити текст жирним

End Sub

7. Внесіть зміни, якщо необхідно, і закрийте редактор VBA, натиснувши **Alt + Q** або **Закрити**.

3. Виконання макросу

Для виконання записаного макросу є кілька способів:

1. На вкладці **Розробник (Developer)** натискайте **Макроси (Macros)**, виберіть потрібний макрос зі списку та натисніть **Запустити (Run)**.
2. Якщо ви призначили комбінацію клавіш для макросу, ви можете просто натискати ці клавіші.
3. Якщо макрос прив'язаний до кнопки або елемента керування на аркуші (наприклад, кнопка або форма), ви можете натискати на цей елемент для запуску макросу.

4. Управління макросами

Щоб краще організувати макроси, ви можете:

1. **Змінити макрос** в редакторі VBA.
2. **Видалити макроси** через діалогове вікно **Макроси** на вкладці **Розробник** (натиснувши **Видалити**).
3. Зберігати макроси в **Особистому файлі макросів** (Personal Macro Workbook), щоб вони були доступні у всіх книгах Excel.

5. Основні операції з VBA в Excel

- **Запуск макросів:** Як згадувалося, це можна робити через меню **Макроси** або через кнопки на листах Excel.
- **Використання змінних:** У VBA ви можете використовувати змінні для зберігання даних, що дозволяє робити макроси більш гнучкими.
- **Умовні оператори (If, Else, Select Case):** Ці оператори дозволяють робити макроси більш динамічними, реагуючи на зміни даних.

- **Цикли (For, While):** Цикли дозволяють повторювати одну й ту саму дію для різних клітинок або діапазонів даних.
- **Обробка помилок:** За допомогою конструкцій **On Error** можна обробляти можливі помилки при виконанні макросів.

Підсумок

- **Запис макросу** – це простий спосіб автоматизувати завдання без необхідності писати код вручну. Ви просто записуєте дії, і Excel перетворює їх у VBA.
- **Редагування макросу** дозволяє налаштовувати і удосконалювати записаний код.
- Використовуйте **редактор VBA** для гнучкого управління та редагування макросів.

Макроси в Excel дозволяють значно пришвидшити роботу з даними та автоматизувати багато повторюваних завдань, що робить роботу в Excel набагато ефективнішою.

4.3. Використання кнопок для запуску макросів

В Excel можна створити кнопки на аркуші, за допомогою яких можна запускати макроси. Це зручний спосіб взаємодії з макросами, оскільки користувач може просто натискати кнопку для виконання певних дій, замість того, щоб йти до редактора макросів або натискати комбінацію клавіш.

Ось кроки для створення кнопки та прив'язки її до макросу:

1. Додавання кнопки на аркуш

Кроки:

1. **Відкрийте вкладку "Розробник" (Developer):**
2. **Вибір елемента керування "Кнопка":**
 - о Перейдіть на вкладку **Розробник** і натисніть на кнопку **Вставити (Insert)**.

- о В розділі **Форми** виберіть кнопку **Кнопка (Button)**.

3. Розташування кнопки на аркуші:

- о Після вибору кнопки курсор зміниться на хрестик. Клацніть лівою кнопкою миші на місце на аркуші, де ви хочете розмістити кнопку.
- о Після цього на аркуші з'явиться вікно, яке запросить вас вибрати макрос для запуску.

4. Прив'язка макросу до кнопки:

- о У діалоговому вікні **Призначити макрос (Assign Macro)** виберіть макрос, який ви хочете прив'язати до кнопки.
- о Натисніть **ОК**.

2. Налаштування кнопки

Кнопка автоматично додається на аркуш, і ви можете виконати додаткові налаштування, щоб змінити її вигляд:

- **Зміна тексту на кнопці:**
 - о Клацніть правою кнопкою на кнопці та виберіть **Редагувати текст (Edit Text)**.
 - о Напишіть бажаний текст на кнопці (наприклад, "Запустити макрос").
- **Зміна розміру або стилю кнопки:**
 - о Ви можете змінити розмір кнопки, потягнувши за краї або кути.
 - о Для зміни стилю кнопки натисніть правою кнопкою миші на кнопці та виберіть **Формат елемента керування (Format Control)**, де можна налаштувати кольори, шрифт та інші параметри.

3. Редагування макросу

Щоб змінити макрос, який виконується при натисканні кнопки, вам потрібно зайти в редактор VBA:

1. Натисніть **Alt + F11**, щоб відкрити редактор VBA.
2. У вікні **Project Explorer** знайдіть книгу Excel і розгорніть її.
3. Виберіть **Module**, де зберігається ваш макрос, і редагуйте його відповідно до ваших потреб.
4. Після редагування збережіть зміни і поверніться до Excel.

4. Типи кнопок для запуску макросів

В Excel є два основних типи кнопок, які можна використовувати для запуску макросів:

1. Кнопки форми (Form Controls)

- Це кнопки, які ви додаєте через вкладку **Розробник** → **Вставити** → **Кнопка**.
- Вони прості у використанні і дозволяють прив'язати макрос без написання коду.
- Підходять для простих задач.

2. Активні елементи керування (ActiveX Controls)

- Це більш складні елементи, які також дозволяють прив'язати макроси, але вони надають більше можливостей для налаштування.
- Для додавання таких кнопок вам потрібно вибрати **ActiveX Controls** замість **Form Controls** в діалоговому вікні **Вставити**.
- Ви можете налаштовувати ці кнопки на більш детальному рівні, використовуючи VBA код (наприклад, змінювати їх вигляд, розмір, дії тощо).

5. Запуск макросу через кнопку: приклад коду

Приклад простого макросу, який можна прив'язати до кнопки:

```
Sub MyMacro()
```

```
    MsgBox "Це повідомлення, яке з'являється при натисканні  
кнопки."
```

```
End Sub
```

Після того, як ви створите кнопку і прив'яжете цей макрос до неї, при натисканні на кнопку на аркуші з'явиться повідомлення **"Це повідомлення, яке з'являється при натисканні кнопки."**

6. Використання кнопок для більш складних завдань

Кнопки можна використовувати не тільки для простих макросів, але й для більш складних завдань, таких як:

- Автоматизація звітів.

- Створення діаграм.
- Введення та перевірка даних.
- Інтерактивні форми та панелі управління.

Підсумок

- Кнопки в Excel — це зручний спосіб запускати макроси.
- Вони дозволяють користувачам виконувати складні операції одним натисканням.
- Ви можете використовувати **кнопки форми** для простих задач або **активні елементи керування** для складніших сценаріїв.
- Кнопки з макросами дозволяють автоматизувати повторювані дії та підвищують зручність користування Excel.

Використання кнопок є потужним інструментом для покращення ефективності роботи в Excel, дозволяючи значно спростити виконання складних або повторюваних операцій.

4.4. Робота з діалоговими вікнами (MsgBox, InputBox, UserForm)

В Excel VBA можна використовувати різні діалогові вікна для взаємодії з користувачем. Це дозволяє створювати більш інтерактивні макроси та автоматизувати роботу з даними. Існують три основні типи діалогових вікон: **MsgBox**, **InputBox** та **UserForm**.

1. MsgBox (Message Box)

MsgBox — це вбудоване діалогове вікно для виведення повідомлень. Ви можете використовувати його для виведення інформативних повідомлень, попереджень або помилок, а також для отримання від користувача підтвердження або відповіді.

Синтаксис:

MsgBox prompt, [buttons], [title], [helpfile], [context]

- prompt: Текст, який буде відображатися в вікні повідомлення.

- **buttons** (необов'язково): Параметр, який визначає, які кнопки буде показано (наприклад, **ОК**, **Yes/No**). За замовчуванням відображається лише одна кнопка **ОК**.
- **title** (необов'язково): Заголовок вікна повідомлення.
- **helpfile** і **context**: Параметри для підключення до системи довідки (зазвичай не використовуються).

Приклад:

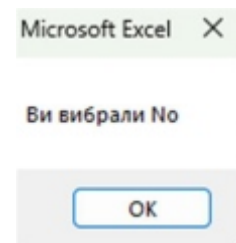
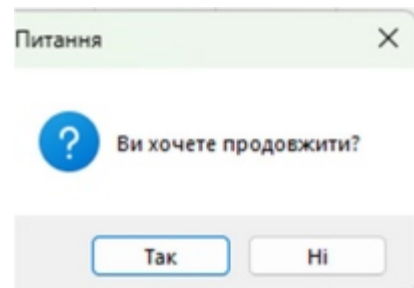
```
Sub ShowMessage()
    MsgBox "Це інформативне повідомлення", vbInformation,
        "Повідомлення"
End Sub
```

Кнопки MsgBox:

- **vbOKOnly** – лише кнопка **ОК** (за замовчуванням).
- **vbYesNo** – кнопки **Yes** і **No**.
- **vbRetryCancel** – кнопки **Retry** і **Cancel**.
- **vbCritical** – вікно з помилкою.
- **vbInformation** – інформативне повідомлення.

Приклад з варіантами кнопок:

```
Sub MsgBoxExample()
    Dim result As Integer
    result = MsgBox("Ви хочете продовжити?", vbYesNo +
        vbQuestion, "Питання")
    If result = vbYes Then
        MsgBox "Ви вибрали Yes"
    Else
        MsgBox "Ви вибрали No"
    End If
End Sub
```



Пояснення:

- **vbYesNo** – показує кнопки **Yes** та **No**.
- **vbQuestion** – додає іконку запитання.
- Після натискання кнопки відповідь зберігається у змінній `result`, і можна виконати різні дії в залежності від вибору користувача.

2. InputBox

InputBox – це вбудоване діалогове вікно, яке дозволяє користувачеві вводити текст або числа. Ви можете використовувати його для запиту у користувача якихось значень, наприклад, для введення імені, числа, дати і т.д.

Синтаксис:

InputBox(prompt, [title], [default], [xpos], [ypos], [helpfile], [context])

- `prompt`: Текст, який з'являється в полі вводу.
- `title` (необов'язково): Заголовок вікна вводу.
- `default` (необов'язково): Текст, який буде вказаний у полі вводу за замовчуванням.
- `xpos` і `ypos` (необов'язково): Позиція вікна на екрані.
- `helpfile` і `context`: Параметри для підключення до системи довідки (зазвичай не використовуються).

Приклад:

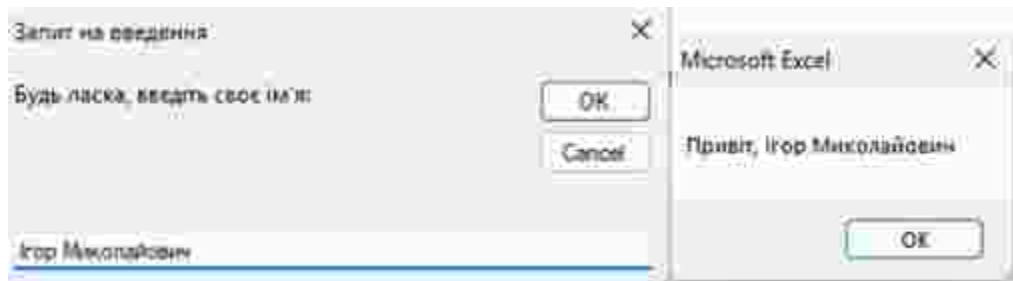
```
Sub GetUserInput()
```

```
    Dim userName As String
```

```
    userName = InputBox("Будь ласка, введіть своє ім'я:",  
        "Запит на введення", "Ім'я")
```

```
    MsgBox "Привіт, " & userName
```

```
End Sub
```

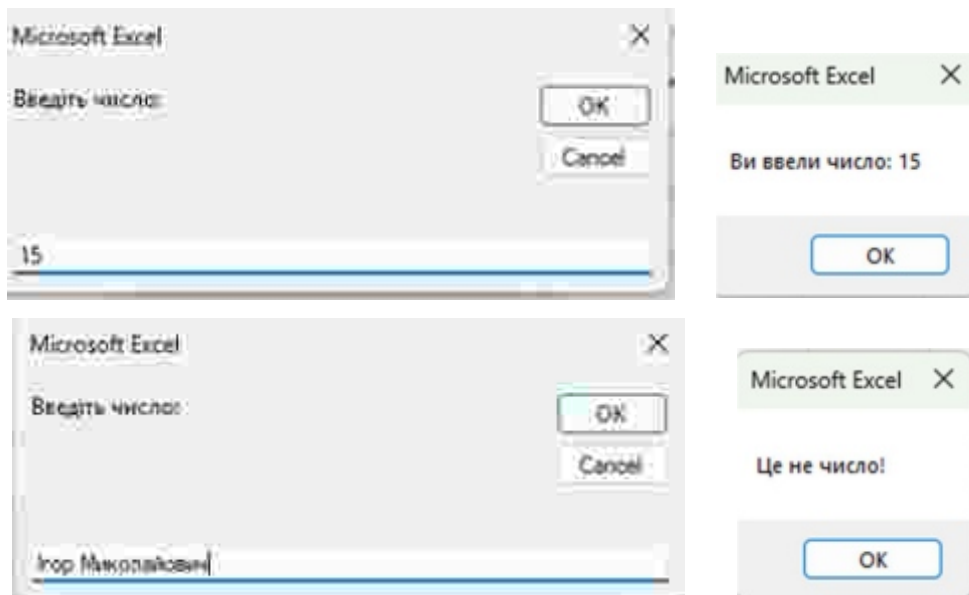


Приклад з перевіркою введених даних:

```

Sub GetValidNumber()
    Dim userInput As String
    userInput = InputBox("Введіть число:")
    If IsNumeric(userInput) Then
        MsgBox "Ви ввели число: " & userInput
    Else
        MsgBox "Це не число!"
    End If
End Sub

```



3. UserForm

UserForm – це користувацька форма, яку можна створювати для збору даних, введення інформації або створення складних інтерактивних інтерфейсів. UserForm дозволяє створювати вікна з різними елементами керування, такими як текстові поля, списки, кнопки, прапорці, комбо-бокси та ін.

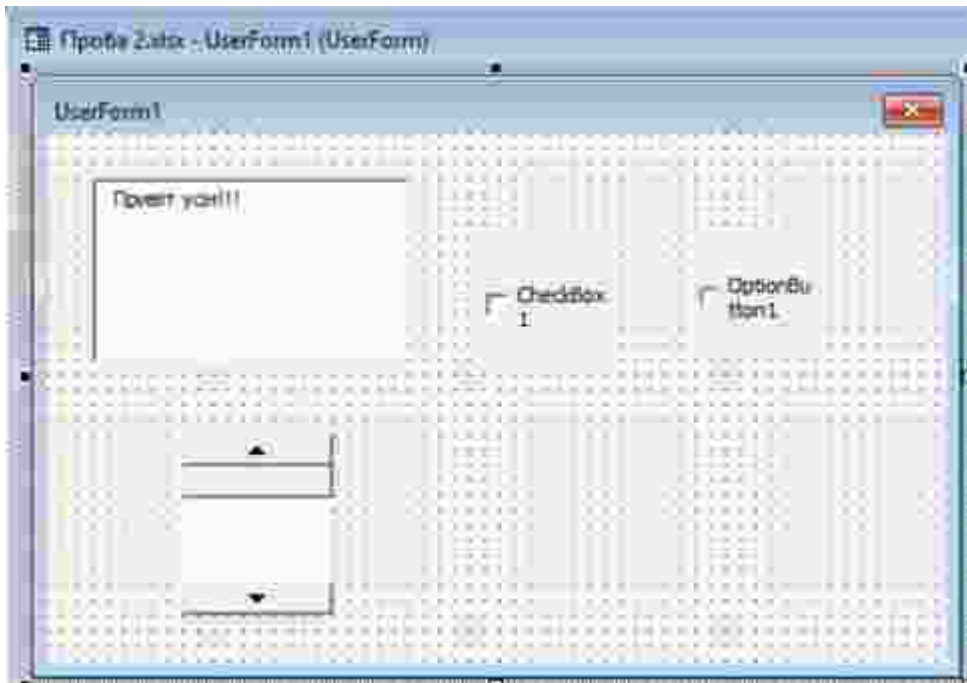
Кроки для створення UserForm:

1. Відкрийте редактор VBA за допомогою **Alt + F11**.
2. Виберіть **Insert** → **UserForm** для створення нової форми.
3. Додайте елементи керування (наприклад, кнопки, текстові поля, мітки) з панелі інструментів на формі.

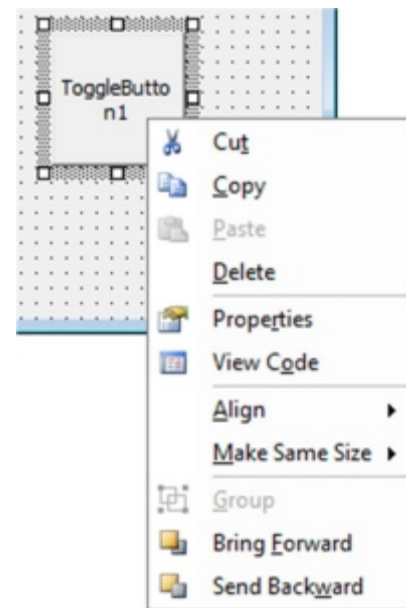
4. Для кожного елемента можна написати відповідний код для обробки подій.

Приклад:

1. Створіть **UserForm** з кнопкою **OK** і текстовим полем.



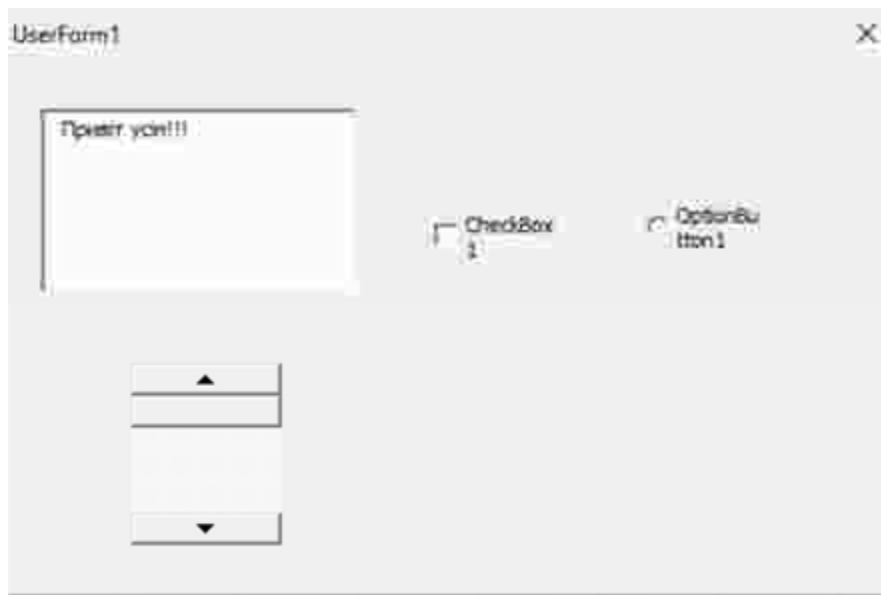
2. Призначте код для кнопки, натиснувши праву кнопку мишки. У відкритому контекстному меню оберіть View Code:



```
Private Sub  
    CommandButton1_Click()  
        MsgBox "Ви ввели: " &  
            TextBox1.Text  
        Unload Me ' Закриває форму  
        після натискання  
    End Sub
```

3. Відкрийте UserForm через макрос:
4.

```
Sub ShowUserForm()  
    UserForm1.Show  
End Sub
```



Основні елементи UserForm:

- TextBox – текстове поле для вводу даних.
- CommandButton – кнопка для виконання дій (наприклад, для підтвердження або закриття форми).
- Label – мітка для виведення тексту або інструкцій.
- ComboBox – випадаючий список для вибору значень.
- ListBox – список для вибору одного або кількох елементів.
- CheckBox – прапорець для вибору/відміни опцій.
- OptionButton (RadioButton) – перемикач для вибору одного з кількох варіантів.

Підсумок:

1. **MsgBox** – використовується для виведення повідомлень або отримання підтвердження від користувача. Підходить для простих інформативних повідомлень.

2. **InputBox** – дозволяє користувачеві ввести значення, яке буде використано в макросі. Ідеальний для введення тексту або чисел.

3. **UserForm** – надає більше можливостей для створення інтерактивних форм, де користувач може вводити дані через різні елементи керування (кнопки, текстові поля, списки тощо).

Ці інструменти допомагають створювати більш інтерактивні та зручні макроси для роботи в Excel, дозволяючи легко збирати дані від користувача або виводити йому необхідну інформацію.

4.5. Індивідуальне завдання №3.

Вивчення елементів VBA

Завдання: Вивчити деякі операції з можливостями мови VBA.

Порядок виконання: використовуються пункт меню Excel «Розробник» та редактор VBA.

Методичні вказівки: 1) Відкрити новий аркуш Excel, в у ньому відкрити пункт меню «Розробник».

- 2) Створіть кнопку1 по центру листа.
- 3) Для цієї кнопки створіть форму користувача, на якій мають бути вікна з кнопками та полями, що містять текст.
- 4) Перевірте, як працює ця кнопка.
- 5) Створіть кнопку2 нижче кнопки 1.
- 6) Створіть код для цієї кнопки, що працює з усіма варіантами MsgBox.
- 7) Перевірте, як працює ця кнопка.
- 8) Створіть кнопку3 нижче кнопки 2.
- 9) Створіть код для цієї кнопки, що працює з усіма варіантами InputBox для введення чисел (не менше трьох) і тексту окремо.
- 10) Вставте перевірку на допустимість типу даних.

11) На тому ж аркуші Excel виконайте розрахунки зі введеними числами і виведіть повідомлення з текстом і результатами розрахунку.

12) З віті помістіть зображення активних вікон з поясненнями.

Контрольні запитання

1. Що таке VBA в Excel, і для чого вона використовується?
2. Яка основна відмінність між Excel і VBA?
3. Що таке редактор VBA, і як до нього потрапити в Excel?
4. Що таке процедура Sub в VBA?
5. Яка різниця між процедурами Sub та Function в VBA?
6. Що таке змінні в VBA, і чому важливо їх правильно оголошувати?
7. Як оголошується змінна в VBA?
8. Як працює оператор If у VBA, і який його синтаксис?
9. Як працює цикл For...Next в VBA? Наведіть приклад використання.
10. Що таке масиви в VBA, і як їх використовувати?
11. Як записати макрос в Excel без написання коду?
12. Які обмеження існують при записі макросу за допомогою засобу запису макросів в Excel?
13. Як редагувати записаний макрос у редакторі VBA?
14. Як можна змінити код записаного макросу для виконання складніших операцій?
15. Які типові помилки можуть виникнути при запису макросу і як їх виправити?
16. Як вставити кнопку на аркуш в Excel для запуску макросу?
17. Як прив'язати макрос до кнопки на аркуші Excel?
18. Як змінити вигляд кнопки, що запускає макрос?
19. Як можна використовувати кнопки для запуску декількох макросів одночасно?

20. Як можна додавати додаткові елементи керування на форму, щоб покращити взаємодію з користувачем?
21. Як створити просте повідомлення за допомогою MsgBox в VBA?
22. Які типи кнопок можуть бути відображені у вікні MsgBox?
23. Як отримати від користувача ввід даних за допомогою InputBox в VBA?
24. Як перевірити, чи введені користувачем дані є числом за допомогою InputBox?
25. Як створити користувацьку форму (UserForm) для введення даних у VBA?
26. Як додавати елементи керування (кнопки, текстові поля) на форму в UserForm?
27. Як обробляти події, які відбуваються при натисканні кнопки на UserForm?
28. Як закрити UserForm після виконання певних дій?
29. Як передати значення з одного UserForm до іншого в VBA?
30. Як додати обробку помилок для введення даних у формах через VBA?
31. Як працює оператор DoEvents, і коли його варто використовувати в макросах?
32. Як працює оператор Select Case, і чому він може бути корисним у VBA?
33. Що таке модуль у VBA, і як з ним працювати?
34. Як працювати з діапазонами (Range) у VBA?
35. Як використовувати макроси для автоматизації форматування даних у Excel?

*Вивчивши матеріали цього студенти
опанували оснвні принципи мови VBA,
навчилися створювати цикли,
повідомлення, форми.*

Розділ 5.

ДИНАМІЧНА ВІЗУАЛІЗАЦІЯ ДАНИХ

*Вивчивши матеріали цього студенти
опанують, як керувати формами
представлення графіків і таблиць.*

Динамічна візуалізація даних в Excel – це спосіб представлення даних у вигляді графіків, діаграм і таблиць, які автоматично оновлюються при зміні вихідних даних. Це дозволяє користувачам швидко аналізувати інформацію та знаходити тенденції.

Основні методи динамічної візуалізації в Excel:

1. Динамічні діаграми – оновлюються при зміні даних (наприклад, за допомогою розумних таблиць або функцій іменованих діапазонів).
2. Зведені таблиці та зведені діаграми – автоматично перебудовуються при оновленні даних.
3. Фільтри та сегменти (Slicers) – дозволяють інтерактивно змінювати відображення даних.
4. Умовне форматування – змінює вигляд комірок залежно від значень (наприклад, кольорові шкали, гістограми).
5. Формули та динамічні масиви – використовуються для автоматичного перерахунку та оновлення даних.
6. Макроси та VBA – можуть створювати складніші інтерактивні візуалізації.
7. Power Query та Power Pivot – використовуються для обробки великих обсягів даних та створення розширених дашбордів.

Дашборд (від англ. **dashboard**) — це візуальний інтерфейс для відображення даних, який дозволяє користувачам швидко оцінити поточний стан різних показників або процесів. Зазвичай він включає в себе графіки, діаграми,

таблиці та інші візуальні елементи, що допомагають зрозуміти важливу інформацію на одному екрані.

Дашборди широко використовуються в бізнесі, аналізі даних, маркетингу, фінансах, медицині та в багатьох інших сферах. Вони дозволяють отримувати інформацію в реальному часі, приймати рішення та відстежувати зміни в процесах.

Основні характеристики дашборду:

1. **Візуалізація даних:** графіки, діаграми, карти, таблиці.
2. **Інтерактивність:** користувачі можуть взаємодіяти з даними, фільтрувати або деталізувати інформацію.
3. **Оперативність:** відображає дані в реальному часі або з мінімальною затримкою.

Приклад: корпоративний дашборд може показувати фінансові показники компанії, продажі, складські запаси та інші важливі метрики на одному екрані для швидкої оцінки ситуації.

Динамічна візуалізація в Excel корисна для аналізу фінансових показників, маркетингових досліджень, прогнозування тенденцій та інших бізнес-аналітичних завдань.

5.1. Використання умовного форматування

Умове форматування – це інструмент, який змінює вигляд комірок (колір фону, тексту, рамки) залежно від їхнього значення. Це допомагає швидко знаходити тенденції, виділяти важливі дані та покращувати аналіз інформації.

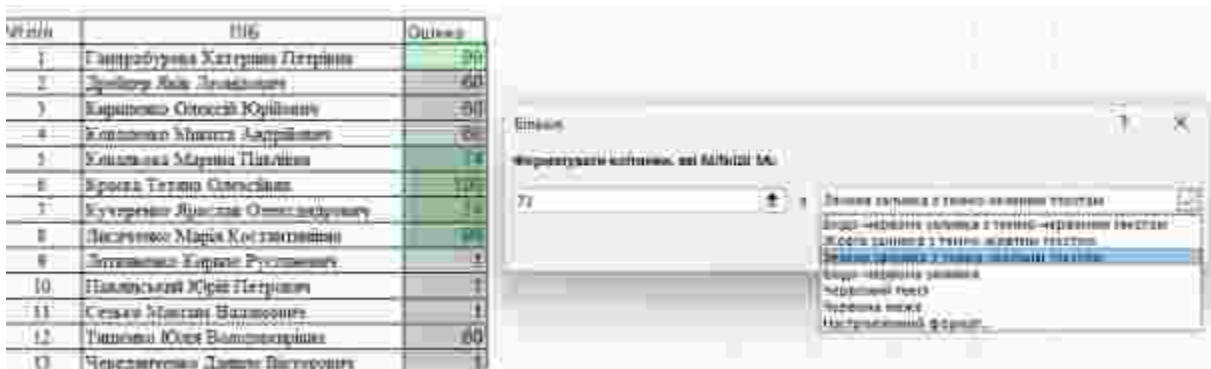
Розглянемо приклади використання умовного форматування.

1. Виділення найбільших і найменших значень

 **Завдання:** У вас є список оцінок студентів, і потрібно підсвітити найвищу та найнижчу оцінки.

✓ **Рішення:**

1. Виділіть діапазон з оцінками.
2. Відкрийте "Головна" → "Умовне форматування" → "Правила виділення комірок" → "Більше ніж / Менше ніж".
3. Введіть значення для виділення, наприклад, "Найбільше значення".
4. Виберіть колір для підсвічування та натисніть "ОК".



2. Автоматичне виділення прострочених дат

✚ **Завдання:** Є список завдань із дедлайнами, і вам потрібно виділити червоним прострочені дати.

✓ **Рішення:**

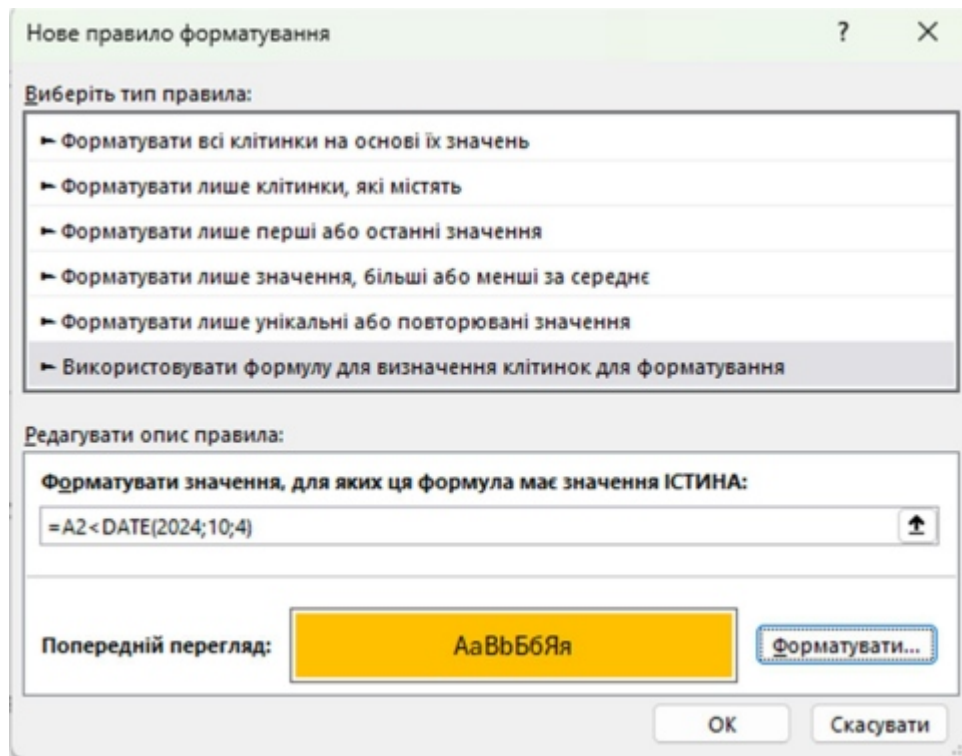
1. Виділіть діапазон дат.
2. Перейдіть до "Умовне форматування" → "Нові правила" → "Форматувати комірки на основі формули".

3. Введіть формулу:

=A2<DATE(2024;10;4)

Це означає, що комірки з датами, раніше за визначену дату будуть форматуватися.

4. Виберіть потрібний колір заливки → **ОК**. І ті комірки, що містять дату пізніше заданої, будуть відмічені заданим кольором.

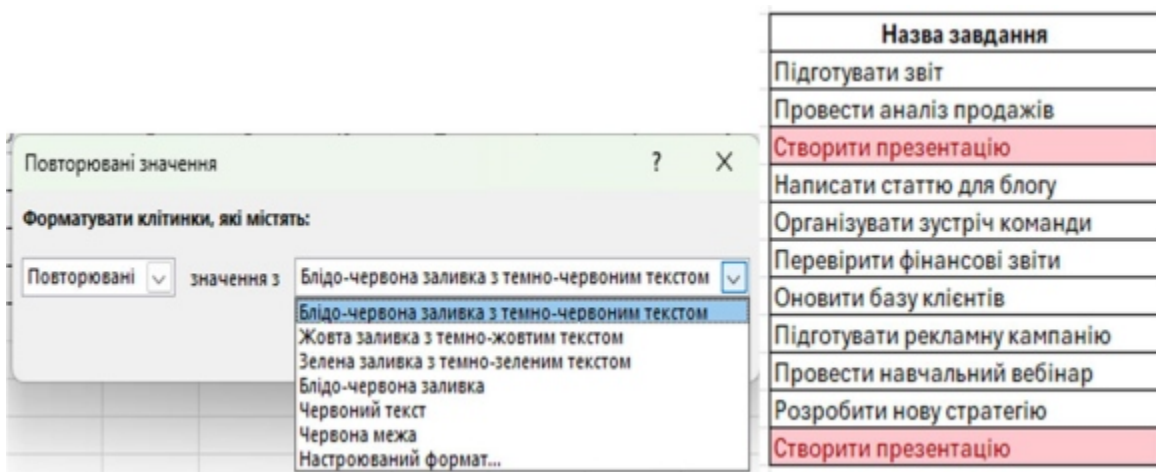
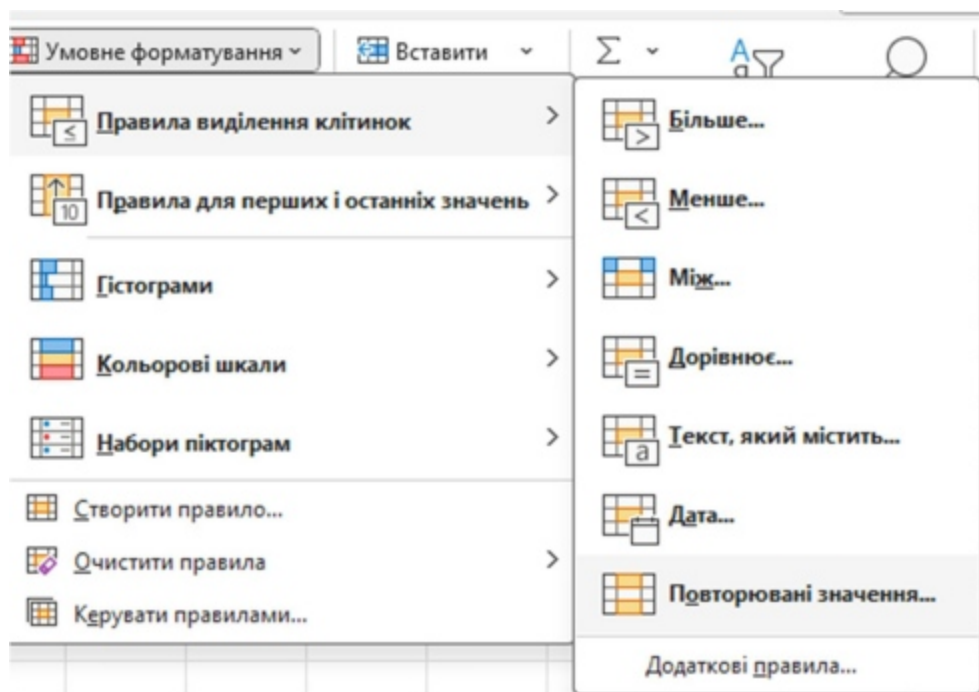


3. Виділення дублікатів у списку

✦ **Завдання:** У вас є список імен, і потрібно знайти повторювані записи.

✓ **Рішення:**

1. Виділіть діапазон даних.
2. Перейдіть до "Умовне форматування" → "Правила виділення комірок" → "Повторювані значення".
3. Виберіть колір для дублікатів та натисніть "ОК".



4. Гістограми в комірках

🚩 **Завдання:** Візуалізувати рівень продажів у таблиці.

✅ **Рішення:**

1. Виділіть діапазон даних.
2. Виберіть "Умовне форматування" → "Гістограми".
3. Виберіть стиль гістограми та натисніть "ОК".

Правила виділення клітинок >

Правила для перших і останніх значень >

Гістограми >

Кольорові шкали >

Набори піктограм >

Створити правило...

Очистити правила >

Керувати правилами...

Сортувати й
фільтрувати >

Знайти
виділіти

Редагування

Назва товару	Рівень продажів (шт.)
Смартфон	150
Ноутбук	85
Планшет	60
Телевізор	40
Навушники	120
Смарт-годинник	95
Ігрова приставка	55
Пилосос	30
Холодильник	25
Мікрохвильова піч	45

5. Колірна шкала для порівняння значень

✦ **Завдання:** В таблиці з фінансовими показниками потрібно виділити великі числа зеленим, а малі – червоним.

✓ **Рішення:**

1. Виділіть діапазон чисел.
2. Перейдіть до "Умове форматування" → "Колірні шкали".
3. Виберіть градієнт (наприклад, від червоного до зеленого).

Гістограми

Кольорові шкали >

Набори піктограм >

Створити правило...

Очистити правила >

Керувати правилами...

M N O

Додаткові правила...

Назва товару	Рівень продажів (шт.)
Смартфон	150
Ноутбук	85
Планшет	60
Телевізор	40
Навушники	120
Смарт-годинник	95
Ігрова приставка	55
Пилосос	30
Холодильник	25
Мікрохвильова піч	45

Умовне форматування можна автоматично задавати за допомогою VBA в Excel. Ось кілька прикладів, як це зробити.

1. Виділення комірок із продажами нижче 50 (червона заливка)

Цей макрос перевіряє значення у стовпці **B2:B11**. Якщо продажі **менше 50**, комірки забарвляться в червоний.

```
Sub FormatLowSales()  
    Dim ws As Worksheet  
    Dim rng As Range  
    Dim cond As FormatCondition  
    ' Встановлюємо робочий лист і діапазон  
    Set ws = ThisWorkbook.Sheets("Лист1") ' Замініть на свій лист  
    Set rng = ws.Range("B2:B11") ' Діапазон із рівнями продажів  
    ' Видаляємо попереднє умовне форматування  
    rng.FormatConditions.Delete  
    ' Додаємо нове правило  
    Set cond = rng.FormatConditions.Add(Type:=xlCellValue,  
Operator:=xlLess, Formula1:="50")  
    ' Встановлюємо формат (червоний фон)  
    With cond.Interior  
        .Color = RGB(255, 0, 0) ' Червоний колір  
    End With  
End Sub
```

2. Виділення дат, які менші за поточну дату (прострочені)

Цей код перевіряє **стовпець B2:B11**. Якщо в комірці дата **менша за сьогоднішню**, вона стане жовтою.

```
Sub HighlightExpiredDates()  
    Dim ws As Worksheet
```

```

Dim rng As Range
Dim cond As FormatCondition
' Вибір аркуша та діапазону
Set ws = ThisWorkbook.Sheets("Лист1") ' Замініть на свій лист
Set rng = ws.Range("B2:B11") ' Діапазон із датами
' Очищення попереднього умовного форматування
rng.FormatConditions.Delete
' Додаємо правило: дата менша за сьогодні
Set cond = rng.FormatConditions.Add(Type:=xlExpression,
Formula1:="=B2<TODAY()")
' Змінюємо формат (жовтий фон)
With cond.Interior
    .Color = RGB(255, 255, 0) ' Жовтий
End With
End Sub

```

3. Використання умовного форматування для порівняння з іншою коміркою.

Цей макрос перевіряє, чи значення в B2:B11 менше, ніж у C1. Якщо так – комірки будуть синіми.

Якщо потрібно виділити всі значення, які менші за певну цифру в комірці C1, можна використати таку формулу:

```

Sub CompareWithCell()
    Dim ws As Worksheet
    Dim rng As Range
    Dim cond As FormatCondition
    Set ws = ThisWorkbook.Sheets("Лист1")
    Set rng = ws.Range("B2:B11") ' Діапазон із продажами
    ' Видалення старого форматування
    rng.FormatConditions.Delete

```

```

' Додаємо правило: значення менше, ніж у комірії C1
Set cond = rng.FormatConditions.Add(Type:=xlExpression,
Formula1:="=B2<$C$1")
' Формат - синій фон
With cond.Interior
.Color = RGB(0, 0, 255) ' Синій колір
End With
End Sub

```

Як запустити макрос у VBA?

- Відкрийте Excel
- Натисніть ALT + F11, щоб відкрити редактор VBA
- Вставте код у **модуль** (Insert → Module)
- Запустіть макрос (Run або F5)

5.2. Динамічні діаграми (включення/виключення серій даних)

Динамічні діаграми дозволяють керувати відображенням серій даних, даючи можливість вмикати або вимикати певні дані без редагування самої таблиці. Це зручно, якщо потрібно аналізувати різні набори даних без створення нових діаграм.

1. Динамічна діаграма за допомогою прапорців (CheckBox)

Використовується **CheckBox** (прапорці) для керування відображенням серій даних у діаграмі.

Приклад: Продажі різних товарів по місяцях

Місяць	Смартфони	Ноутбуки	Телевізори
Січень	120	90	45
Лютий	150	85	50
Березень	130	88	47

Кроки для створення динамічної діаграми

Створіть стандартну діаграму:

- Виділіть тільки дані без заголовків рядків і стовпців, наприклад A1:D4.
- Перейдіть у "Вставка" → "Гістограма" або "Лінійчата діаграма".

Додайте CheckBox:

- Перейдіть у "Розробник" → "Вставити" → "Прапорець (CheckBox)".
- Вставте три прапорці (для Смартфонів, Ноутбуків, Телевізорів).

Прив'яжіть CheckBox до комірок:

- Натисніть правою кнопкою на CheckBox → "Формат елемента керування".
- У полі "Зв'язати з коміркою" виберіть, наприклад, F1 (для смартфонів), F2 (для ноутбуків), F3 (для телевізорів).
- Тепер у комірках F1, F2, F3 буде TRUE/FALSE залежно від стану прапорця.

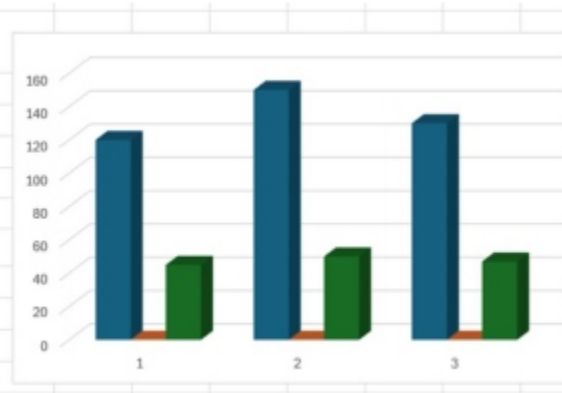
Створіть допоміжну таблицю з умовами:

Місяць	Смартфони	Ноутбуки	Телевізори
Січень	=IF(F1=TRUE;B2;"")	=IF(F2=TRUE;C2;"")	=IF(F3=TRUE;D2;"")
Лютий	=IF(F1=TRUE;B3;"")	=IF(F2=TRUE;C3;"")	=IF(F3=TRUE;D3;"")
Березень	=IF(F1=TRUE;B4;"")	=IF(F2=TRUE;C4;"")	=IF(F3=TRUE;D4;"")

Оновіть діаграму, використовуючи цю таблицю (коли CheckBox вимкнений, відповідні дані будуть порожніми, і діаграма автоматично їх приховає).

◆ **Результат:** Користувач може вмикати/вимикати серії даних на діаграмі, використовуючи прапорці. На рисунку вимкнено дані для ноутбуків.

Місяць	Смартфони	Ноутбуки	Телевізори
Січень	120	90	45
Лютий	150	85	50
Березень	130	88	47
Прапорці	☒	☐	☒
Зв'язок	TRUE	FALSE	TRUE
Допоміжна таблиця			
Місяць	Смартфони	Ноутбуки	Телевізори
Січень	120		45
Лютий	150		50
Березень	130		47



2. Динамічна діаграма через список (Drop-down List)

Якщо потрібно показувати **лише одну серію даних**, можна використати випадаючий список.

Приклад: Вибір товару для аналізу

Створіть вихідну таблицю:

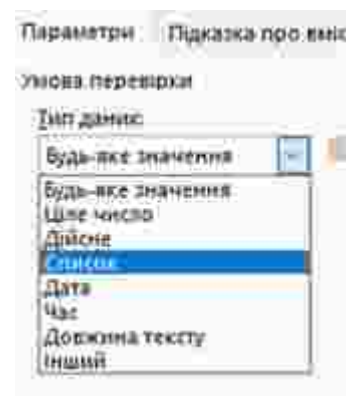
Місяць	Смартфони	Ноутбуки	Телевізори
Січень	120	90	45
Лютий	150	85	50
Березень	130	88	47

Додайте випадаючий список (Data Validation):

- Виділіть комірку F1,
- Перейдіть в "Дані" → «Знаряддя даних»

→ "Перевірка даних" ,

- Виберіть "Список" і введіть посилання на список товарів.
- Тепер у комірці F1 можна вибирати товар.

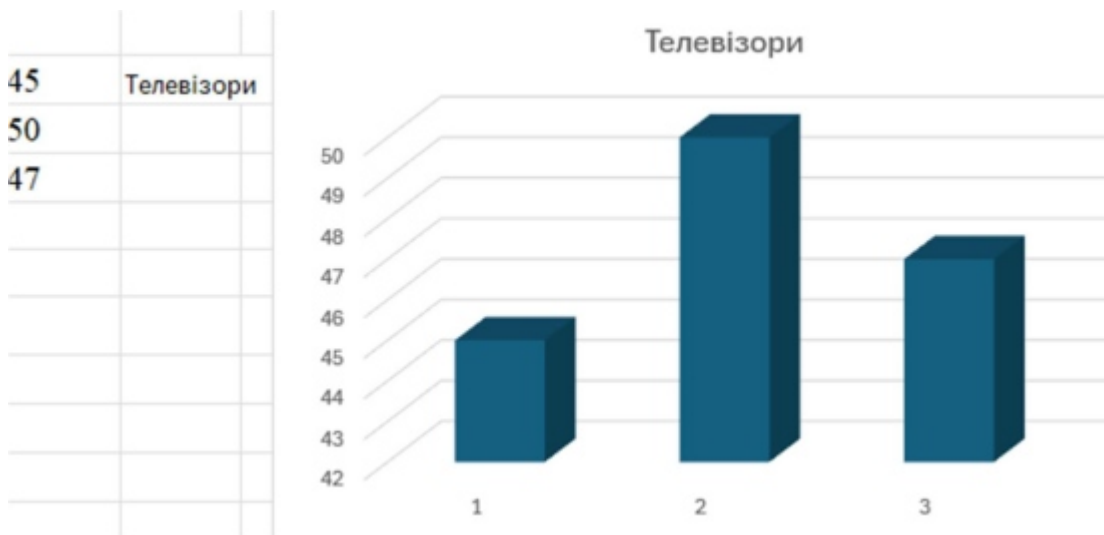


Створіть допоміжну колонку для вибору серії:

- У комірці G2 введіть:
- =IF(\$F\$1="Смартфони";B2;IF(\$F\$1="Ноутбуки";C2;
IF(\$F\$1="Телевізори";D2;"")))
- Протягніть формулу вниз.

Створіть діаграму, використовуючи новий діапазон (A2:A4 та G2:G4).

Тепер, коли користувач вибирає товар у F1, діаграма автоматично оновлюється!



3. VBA-метод для динамічного керування серіями

Якщо потрібно автоматично приховувати серії без допоміжної таблиці, можна використати VBA.

Приклад VBA-коду для приховування серій на діаграмі

```
Sub ToggleSeries()  
    Dim ch As Chart  
    Dim seriesIndex As Integer  
    Dim seriesExists As Boolean  
    ' Перевіряємо, чи існує діаграма  
    On Error Resume Next  
    Set ch = ActiveSheet.ChartObjects("Діаграма 1").Chart  
    On Error GoTo 0
```

```

If ch Is Nothing Then
    MsgBox "Діаграма не знайдена!"
Exit Sub
End If
' Перевірка на існування серії
seriesExists = False
For seriesIndex = 1 To ch.SeriesCollection.Count
    If Not ch.SeriesCollection(seriesIndex) Is Nothing Then
        seriesExists = True
    Exit For
End If
Next seriesIndex
If Not seriesExists Then
    MsgBox "Серії не знайдено!"
Exit Sub
End If
' Перемикаємо відображення серії
If Range("F1").Value = False Then
    ch.SeriesCollection(1).Format.Line.Visible = msoFalse
Else
    ch.SeriesCollection(1).Format.Line.Visible = msoTrue
End If
End Sub

```

Пояснення коду:

1. **Set ch = ActiveSheet.ChartObjects("Діаграма 1").Chart:**
 - о Ця частина коду вибирає діаграму на активному аркуші за її ім'ям "Діаграма 1".
 - о Якщо діаграма має інше ім'я, замініть "Діаграма 1" на актуальне ім'я діаграми.
2. **If Range("F1").Value = False Then:**

- о Перевіряється значення в клітинці F1. Якщо там значення FALSE, то перша серія діаграми буде прихована, інакше — показана.

3. **ch.SeriesCollection(1).Format.Line.Visible = msoFalse:**

- о Відповідає за приховування лінії першої серії на діаграмі.

4. **ch.SeriesCollection(1).Format.Line.Visible = msoTrue:**

- о Відповідає за відображення лінії першої серії.

Якщо на вашій діаграмі більше ніж одна серія і ви хочете перемикаати видимість кожної серії, потрібно додати цикл, який дозволить перемикаати видимість для кожної серії окремо.

Ось як можна модифікувати код для перемикання видимості всіх серій:

Sub ToggleSeries()

Dim ch As Chart

Dim i As Integer

Set ch = ActiveSheet.ChartObjects("Діаграма 1").Chart

' Перевіряємо, скільки серій є на діаграмі

For i = 1 To ch.SeriesCollection.Count

' Перемикаємо видимість серії

If Range("F1").Value = False Then

ch.SeriesCollection(i).Format.Line.Visible = msoFalse

Else

ch.SeriesCollection(i).Format.Line.Visible = msoTrue

End If

Next i

End Sub

Опис змін:

- **Цикл For:** Тепер код проходить через всі серії, які є на діаграмі, і перемикає їх видимість.
- **ch.SeriesCollection(i):** Змінено з 1 на i, щоб перебирати всі серії на діаграмі.
- Якщо значення в клітинці F1 – **False**, всі серії на діаграмі будуть приховані. Якщо значення **True**, всі серії будуть відображені.

5.3. Графіки, що змінюються за допомогою кнопок

5.3.1. Вимикання сітки на діаграмах та на аркушах Excel

Щоб включати або вимикати розграфку в Excel за допомогою VBA, ви можете використовувати властивість Gridlines для активного аркуша або діаграм.

Ось два приклади кодів для цього:

1. Вмикання та вимикання розграфки на аркуші

```
Sub ToggleGridlines()  
    ActiveWindow.DisplayGridlines      =      Not  
    ActiveWindow.DisplayGridlines  
End Sub
```

Пояснення:

- **ActiveSheet.ShowGridlines:** Це властивість контролює видимість розграфки на аркуші.
- Якщо розграфка включена, вона вимикається, і навпаки.

2. Вимкнення розграфки для конкретного діапазону клітинок

Якщо вам потрібно вимкнути відображення ліній тільки для певного діапазону, ось як це зробити:

```
Sub HideGridlinesForRange()  
    With ActiveSheet.Range("A1:C10")  
        .Borders(xlEdgeBottom).LineStyle = xlNone  
        .Borders(xlEdgeRight).LineStyle = xlNone  
    End With  
End Sub
```

Пояснення:

- Ці команди вимикають сітку (відображення ліній) тільки для діапазону клітинок (наприклад, з A1 по C10), не зачіпаючи решту аркуша.

3. Вимкнення розграфки для діаграм (не для аркуша)

Якщо потрібно вимкнути сітку для діаграм, ось код:

```
Sub ToggleChartGridlines()  
    Dim ch As Chart  
    Set ch = ActiveSheet.ChartObjects("Діаграма 1").Chart  
  
    ' Перемикає видимість розграфки для діаграми  
    If ch.HasAxis(xlCategory) Then  
        If ch.Axes(xlCategory).HasMajorGridlines Then  
            ch.Axes(xlCategory).HasMajorGridlines = False  
        Else  
            ch.Axes(xlCategory).HasMajorGridlines = True  
        End If  
    End If  
End Sub
```

Пояснення:

- HasMajorGridlines: Це властивість діаграми, яка дозволяє включати або вимикати основну сітку для категорійної осі.
- Ви можете застосувати цей код до різних осей, залежно від потреб (наприклад, для осі Y або X).

4. Вмикання розграфки на аркуші та діаграмі разом

```
Sub ToggleGridlinesAndChartGridlines()  
    ' Перемикаємо розграфку на аркуші  
    ActiveSheet.ShowGridlines = Not  
    ActiveSheet.ShowGridlines  
  
    ' Перемикаємо сітку на діаграмі  
    Dim ch As Chart  
    Set ch = ActiveSheet.ChartObjects("Діаграма 1").Chart
```

```

If ch.HasAxis(xlCategory) Then
    ch.Axes(xlCategory).HasMajorGridlines = Not
ch.Axes(xlCategory).HasMajorGridlines
End If
End Sub

```

Пояснення:

Цей код спочатку перемикає розграфку на аркуші, а потім перемикає сітку на діаграмі.

Підсумок:

- Для вмикання та вимикання розграфки на аркуші можна використовувати властивість ShowGridlines.
- Для вимикання/вмикання сітки на діаграмі використовується властивість HasMajorGridlines для кожної осі діаграми.

5.3.2. Інші методи для динамічних графіків

Окрім розграфки, є багато інших цікавих способів для створення динамічних графіків, що змінюються за допомогою кнопок у Excel. Ось кілька варіантів, які можна розглянути для теми графіки, що змінюються за допомогою кнопок:

1. Перемикання серій даних на діаграмі

За допомогою кнопок ви можете створювати механізм, який дозволяє вмикати або вимикати певні серії на діаграмі (як ми розглядали раніше).

Код для перемикання видимості серій:

```

Sub ToggleSeries()
    Dim ch As Chart
    Set ch = ActiveSheet.ChartObjects("Діаграма 1").Chart
    ' Перемикаємо видимість першої серії
    If ch.SeriesCollection(1).Format.Line.Visible = msoTrue Then

```

```

        ch.SeriesCollection(1).Format.Line.Visible = msoFalse
    Else
        ch.SeriesCollection(1).Format.Line.Visible = msoTrue
    End If
End Sub

```

Це дозволяє вам змінювати, які серії даних на діаграмі будуть видимі, натискаючи кнопку.

2. Зміна типу діаграми

За допомогою кнопок можна створити механізм для зміни типу діаграми (наприклад, з гістограми на лінійний графік або на кругову діаграму).

Код для зміни типу діаграми:

```

Sub ChangeChartType()
    Dim ch As Chart
    Set ch = ActiveSheet.ChartObjects("Діаграма 1").Chart
    ' Перемикаємо тип діаграми на лінійний
    ch.ChartType = xlLine
End Sub

```

Також можна створити різні кнопки для кожного типу діаграм, наприклад, для графіка типу лінія, діаграми стовпчиків, кругової діаграми і т.д.

Для цього в команді **ch.ChartType = xlLine** потрібно замінити тип діаграми **xlLine** на інший.

Ось повний список типів діаграм, які можна використовувати в Excel VBA через ch.ChartType:

Стандартні типи діаграм

◆ Колонкові (Column Charts)

```

xlColumnClustered    ' Згруповані стовпчасті
xlColumnStacked      ' Накопичені стовпчасті
xlColumnStacked100   ' 100% накопичені стовпчасті

```

◆ Гістограми (Bar Charts)

xlBarClustered ' Згруповані горизонтальні стовпчасті
xlBarStacked ' Накопичені горизонтальні стовпчасті
xlBarStacked100 ' 100% накопичені горизонтальні стовпчасті

◆ Лінійні (Line Charts)

xlLine ' Звичайна лінійна діаграма
xlLineStacked ' Накопичена лінійна діаграма
xlLineStacked100 ' 100% накопичена лінійна
xlLineMarkers ' Лінійна з маркерами
xlLineMarkersStacked ' Накопичена лінійна з маркерами
xlLineMarkersStacked100 ' 100% накопичена лінійна з маркерами

◆ Площинні (Area Charts)

xlArea ' Звичайна площинна діаграма
xlAreaStacked ' Накопичена площинна
xlAreaStacked100 ' 100% накопичена площинна

◆ Кругові (Pie Charts)

xlPie ' Звичайна кругова
xlPieExploded ' Вибухова кругова (відокремлені сегменти)

◆ Кільцеві (Doughnut Charts)

xlDoughnut ' Кільцева діаграма
xlDoughnutExploded ' Вибухова кільцева

◆ Точкові (Scatter Charts)

xlXYScatter ' Точкова без ліній
xlXYScatterLines ' Точкова з лініями
xlXYScatterLinesNoMarkers ' Точкова з лініями без маркерів
xlXYScatterSmooth ' Точкова зі згладженими лініями
xlXYScatterSmoothNoMarkers ' Точкова зі згладженими лініями без маркерів

◆ Бульбашкові (Bubble Charts)

xlBubble ' Бульбашкова діаграма
xlBubble3DEffect ' Бульбашкова з 3D ефектом

◆ **Фінансові (Stock Charts)**

xlStockHLC ' Біржова (High-Low-Close)
xlStockOHLC ' Біржова (Open-High-Low-Close)
xlStockVHLC ' Біржова (Volume-High-Low-Close)
xlStockVOHLC ' Біржова (Volume-Open-High-Low-Close)

◆ **Поверхневі (Surface Charts)**

xlSurface ' 3D поверхнева діаграма
xlSurfaceTopView ' Поверхнева, вид зверху
xlSurfaceTopViewWireframe ' Поверхнева каркасна, вид зверху
xlSurfaceWireframe ' Поверхнева каркасна 3D

◆ **Радарні (Radar Charts)**

xlRadar ' Радіальна (радарна) діаграма
xlRadarFilled ' Заповнена радарна
xlRadarMarkers ' Радарна з маркерами

◆ **Комбіновані (Mixed Charts)**

xlColumnLineCombo ' Колонкова + Лінійна
xlCustomCombo ' Користувачька комбінована

Приклад коду VBA для зміни типу діаграми

Sub ChangeChartType()

Dim ch As Chart

Set ch = ActiveSheet.ChartObjects("Діаграма 1").Chart

**ch.ChartType = xlColumnClustered ' Зміна на стовпчасту
згруповану**

End Sub

Щоб змінювати тип діаграми за допомогою випадаючого списку (Drop-down List) в Excel, потрібно:

1. Створити випадаючий список із типами діаграм.
2. Написати **VBA-код**, який змінюватиме тип діаграми відповідно до вибору користувача.

КРОК 1: Створення випадаючого списку

1. Виділіть комірку (наприклад, E1), де буде список.
2. Перейдіть у Дані → Перевірка даних.
3. Виберіть **Список** як тип перевірки.
4. У полі "Джерело" введіть або вставте:
Column Clustered, Column Stacked, Bar Clustered, Line, Pie, Scatter
5. Натисніть **ОК**.

Тепер у комірці E1 з'явиться випадаючий список.

КРОК 2: VBA-код для зміни типу діаграми

Вставте цей код у модуль VBA:

```
Sub ChangeChartType()  
    Dim ch As Chart  
    Dim chartType As String  
    ' Отримуємо вибраний тип діаграми  
    chartType = Range("E1").Value  
    ' Посилання на діаграму (змінити на вашу)  
    Set ch = ActiveSheet.ChartObjects("Діаграма 1").Chart  
    ' Змінюємо тип діаграми  
    Select Case chartType  
        Case "Column Clustered"  
            ch.ChartType = xlColumnClustered  
        Case "Column Stacked"  
            ch.ChartType = xlColumnStacked  
        Case "Bar Clustered"  
            ch.ChartType = xlBarClustered
```

```

Case "Line"
    ch.ChartType = xlLine
Case "Pie"
    ch.ChartType = xlPie
Case "Scatter"
    ch.ChartType = xlXYScatter
Case Else
    MsgBox "Невідомий тип діаграми!", vbExclamation
End Select
End Sub

```

КРОК 3: Автоматична зміна при виборі в списку

Щоб макрос запускався автоматично при зміні вибору, додайте код у модуль **Worksheet**:

```

Private Sub Worksheet_Change(ByVal Target As Range)
    If Not Intersect(Target, Me.Range("E1")) Is Nothing Then
        Call ChangeChartType
    End If
End Sub

```

ВАЖЛИВО:

- Код потрібно вставити в **аркуш**, на якому знаходиться випадаючий список (Sheet1, Sheet2, тощо).
 - Тепер, коли ви змінюєте значення в E1, діаграма миттєво змінює тип.
-

3. Фільтрація даних для діаграм

Використовуючи кнопки, ви можете створювати фільтри для даних, які використовуються на діаграмі, тим самим динамічно змінюючи візуалізацію залежно від фільтра.

Код для фільтрації даних:

```
Sub FilterData()  
    Dim dataRange As Range  
    Set dataRange = Range("A1:B10") ' Вказати діапазон,  
    який фільтруємо  
    ' Приклад: показати тільки значення більші за 50  
    dataRange.AutoFilter Field:=2, Criteria1:=">50"  
End Sub
```

Цей код може змінювати, які дані відображаються на діаграмі, фільтруючи певні рядки або стовпці.

4. Зміна кольору або стилю серії на діаграмі

Ще один варіант – змінювати колір або стиль серії на діаграмі за допомогою кнопки.

Код для зміни кольору серії:

```
Sub ChangeSeriesColor()  
    Dim ch As Chart  
    Set ch = ActiveSheet.ChartObjects("Діаграма 1").Chart  
    ' Змінюємо колір першої серії на червоний  
    ch.SeriesCollection(1).Format.Fill.ForeColor.RGB =  
    RGB(255, 0, 0)  
End Sub
```

Це дозволяє вам за допомогою кнопки змінювати вигляд серії на діаграмі (наприклад, колір або стиль).

5. Додавання або видалення даних з діаграми

Ви також можете створити кнопку для додавання або видалення даних з діаграми, щоб користувач міг вибірково змінювати діаграму.

Код для додавання нової серії:

```
Sub AddNewSeries()  
    Dim ch As Chart  
    Set ch = ActiveSheet.ChartObjects("Діаграма 1").Chart  
    ' Додаємо нову серію до діаграми  
    ch.SeriesCollection.NewSeries  
    ch.SeriesCollection(ch.SeriesCollection.Count).XValues =  
Range("A1:A10")  
    ch.SeriesCollection(ch.SeriesCollection.Count).Values =  
Range("B1:B10")  
End Sub
```

Код для видалення серії:

```
Sub RemoveSeries()  
    Dim ch As Chart  
    Set ch = ActiveSheet.ChartObjects("Діаграма  
1").Chart  
    ' Видаляємо першу серію  
    ch.SeriesCollection(1).Delete  
End Sub
```

6. Зміна діапазону даних для діаграми

Ви можете створювати механізм для зміни діапазону даних, що використовується на діаграмі, в залежності від того, який користувач вибирає.

Код для зміни діапазону даних:

```
Sub ChangeDataRange()  
    Dim ch As Chart  
    Set ch = ActiveSheet.ChartObjects("Діаграма 1").Chart  
    ' Зміна діапазону даних для діаграми  
    ch.SetSourceData Source:=Range("A1:C10")
```

End Sub

7. Додавання або видалення елементів діаграми (наприклад, заголовків, легенд)

Використовуючи кнопки, можна додавати або прибирати певні елементи діаграми, наприклад, заголовки осей або легенду.

Код для вимкнення легенди:

Sub ToggleLegend()

Dim ch As Chart

Set ch = ActiveSheet.ChartObjects("Діаграма 1").Chart

If ch.HasLegend Then

ch.HasLegend = False

Else

ch.HasLegend = True

End If

End Sub

5.4. Індивідуальне завдання №4.

Візуальні ефекти на графіках

Завдання: Вивчити деякі операції з можливостями зміни графіків.

Порядок виконання: використовуються пункт меню Excel «Розробник» та редактор VBA.

Методичні вказівки: 1) Відкрити новий аркуш Excel, в у ньому відкрити пункт меню «Розробник».

2) Створити таблицю даних, що має не менше 10 рядків, а стовпців, не менше двох.

3) Перший стовпець – дата, другий – назва об'єкту фірми, люди, товари, на ваш розсуд), третій – кількість.

- 4) Виконати дії з розмітки елементів таблиці, описані у пп. 5.1-5.2.
- 5) Створити кнопки, що робитимуть те саме, що і у попередньому завданні, але автоматично перемикають представлення діаграм і таблиць.
- 8) У звіті подавати опис ваших дій з прикладами у вигляді активних вікон з виносками, що пояснюють ваші дії.

Контрольні запитання

1. Що таке умовне форматування в Excel?
2. Як створити правило умовного форматування для підсвічування значень, більших за певне число?
3. Яка формула умовного форматування виділить всі дати, що вже минули?
4. Як використовувати Data Bars в умовному форматуванні?
5. Чи можна застосовувати умовне форматування до всього рядка на основі значення в одній комірці? Як?
6. Що таке динамічна діаграма в Excel?
7. Як можна включати та вимикати серії даних у діаграмі?
8. Яка функція Excel допомагає створювати діапазони для динамічних діаграм?
9. Як можна використовувати іменовані діапазони для оновлення діаграми?
10. Чи можна змінювати тип діаграми залежно від вибору користувача? Як?
11. Які основні методи створення динамічних графіків у Excel?
12. Як можна використовувати кнопки форми для зміни даних на графіку?
13. Що таке Scroll Bar і як він допомагає візуалізації даних?
14. Як використати VBA для зміни типу графіка через кнопку?
15. Які переваги використання кнопок та випадаючих списків у графіках Excel?
16. Як можна вимкнути сітку на діаграмі вручну?

17. Який VBA-код дозволяє приховати або показати сітку на аркуші?
18. Чи впливає приховування сітки на друк документа? Як це виправити?
19. Як використовувати змінні осі у динамічних графіках?
20. Які ще методи, крім кнопок, можна використовувати для оновлення графіків у реальному часі?

Вивчивши матеріали цього студенти зможуть використовувати умовне форматування, оперувати з динамічними діаграмами, створювати графіки, що змінюються за допомогою кнопок.

.

Розділ 6.

ПІДКЛЮЧЕННЯ ДО ЗОВНІШНІХ ДАНИХ

*У цьому розділі студенти вивчать
принципи роботи з інтеграцією даних
та зв'язок з зовнішніми даними.*

6.1. Використання Power Query для інтеграції даних

Power Query є потужним інструментом, і його просунуті функції дозволяють ефективно обробляти великі обсяги даних. Ось деякі з функцій, які можна використовувати для більш складних завдань:

1. Створення параметрів

- Ви можете створити параметри, щоб автоматизувати та повторно використовувати запити. Наприклад, налаштування параметра для динамічної зміни джерела даних чи значення фільтра.

2. Злиття (Merge) та додавання (Append) таблиць

- Об'єднуйте кілька таблиць у одну, зберігаючи логічну структуру даних. Наприклад, злиття даних з різних джерел із використанням загальних ключів.

3. Використання М-коду

- Мова М дозволяє створювати складніші трансформації даних, які недоступні через графічний інтерфейс. Ви можете написати власні функції, оптимізувати обробку даних тощо.

4. Розширений аналіз даних

- **Групування даних (Group By):** дозволяє сегментувати дані на основі ключових полів і застосовувати обчислення, такі як суми чи середні.

- **Визначення користувацьких колонок:** додавайте обчислювані поля з використанням виразів і формул.

5. Очищення та трансформація

- Видалення дублікатів, заповнення пропущених значень, зміна форматів дат чи чисел, розділення чи об'єднання колонок.

6. Автоматизація оновлення даних

- Налаштуйте оновлення запитів, щоб дані автоматично завантажувались із нових джерел або оновлювались відповідно до змін.

7. Підключення до веб-API

- Ви можете підключатися до REST API, щоб отримувати дані з веб-ресурсів і інтегрувати їх у вашу модель.

8. Робота з великими наборами даних

- Ви можете застосовувати фільтри, щоб обробляти лише необхідні дані, знижуючи навантаження на ресурси.

9. Використання шаблонів

- Збережіть запит як шаблон, щоб використовувати його в інших проектах, заощаджуючи час.

Ось приклад, як злиття таблиць у Power Query може бути виконано:

Завдання:

У вас є дві таблиці. Одна містить інформацію про клієнтів, а інша – про їхні замовлення. Ви хочете об'єднати ці таблиці, щоб створити повну картину.

1. Вихідні таблиці: Таблиця Клієнти

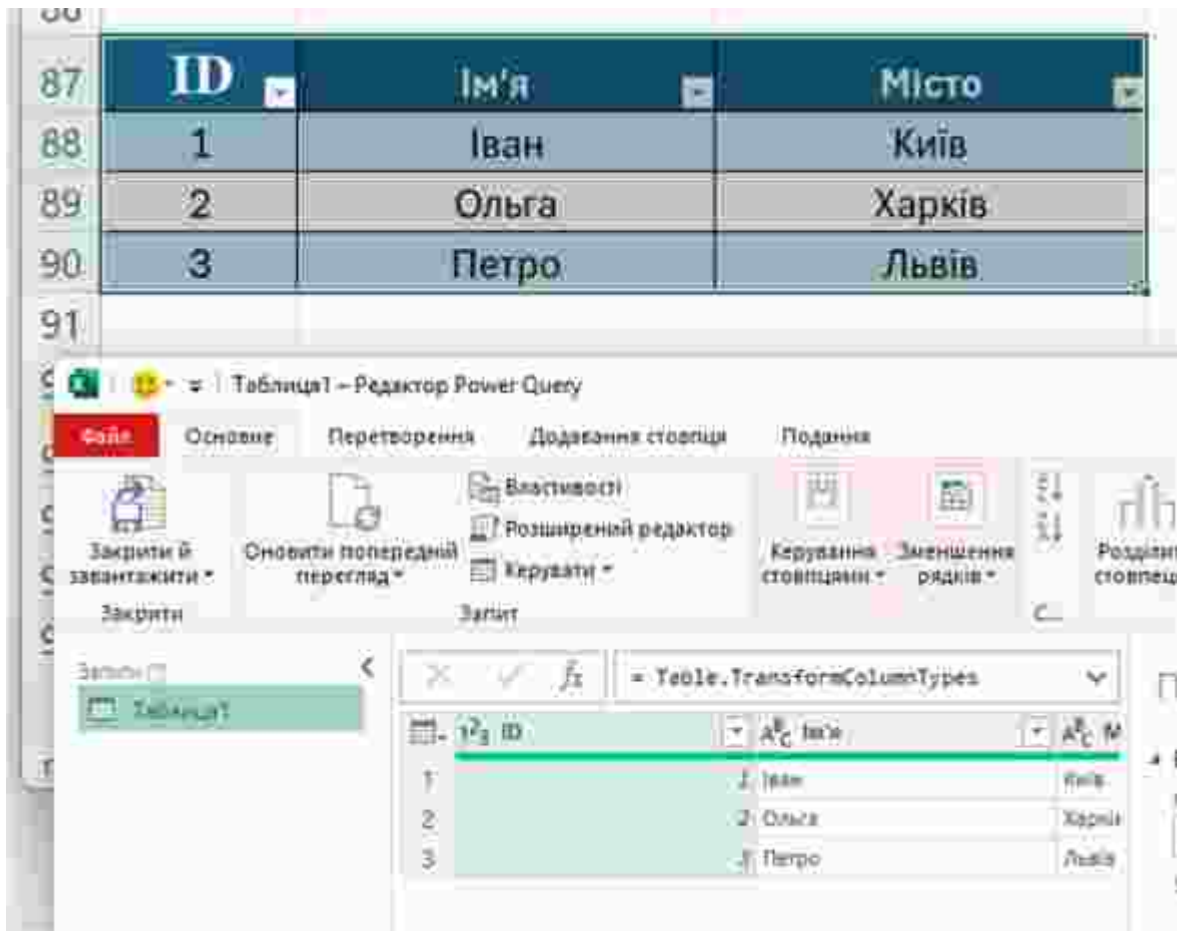
ID	Ім'я	Місто
1	Іван	Київ
2	Ольга	Харків
3	Петро	Львів

Таблиця Замовлення

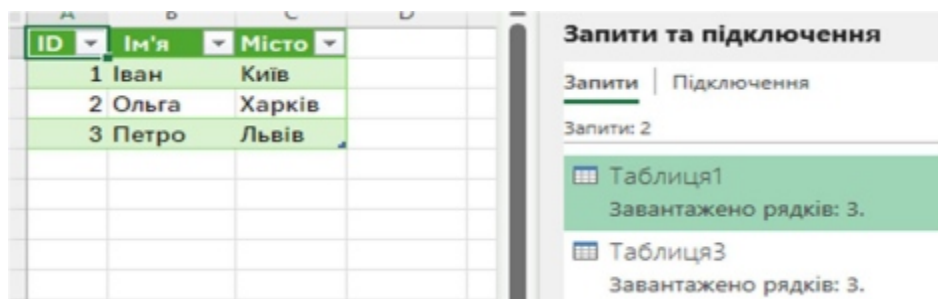
ID	Товар	Сума
1	Ноутбук	1500
2	Телефон	800
3	Мишка	25

Інструкція злиття:

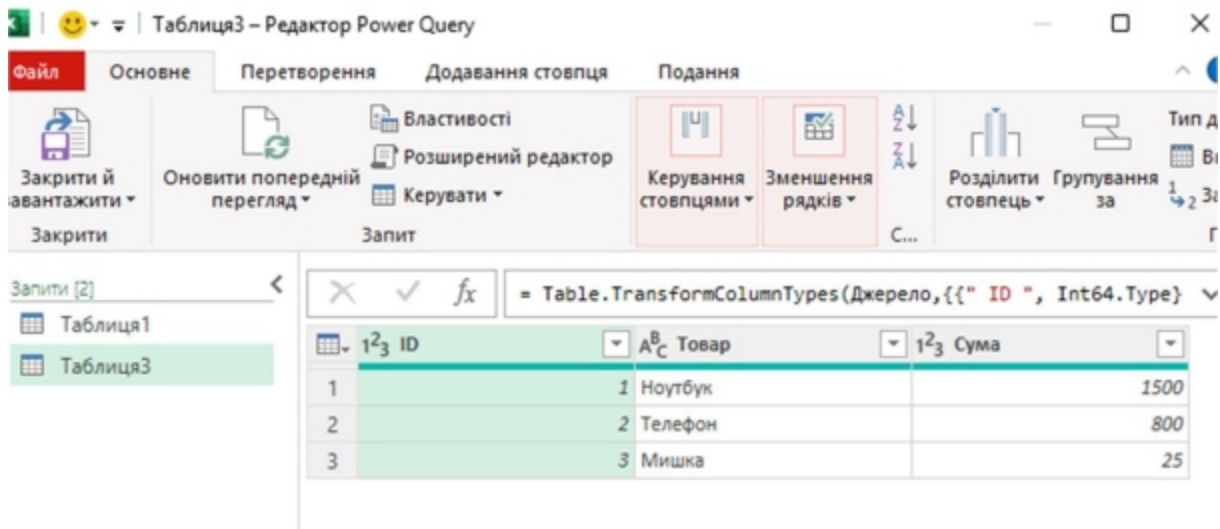
1. **Завантажте дані у Power Query.** Виділіть таблицю, перейдіть на вкладку "Дані" → "Отримати та трансформувати" → "З Power Table/Range". У вас з'явиться нове вікно Power Query із завантаженою таблицею, а у вікні Excel, ваша таблиця стане розграфленою кольорами.



Натисніть кнопку «Закрити і завантажити». Повторіть цю процедуру для другої таблиці. Тепер у вас в Excel утвориться два нових аркуші, що міститимуть ці самі таблиці.

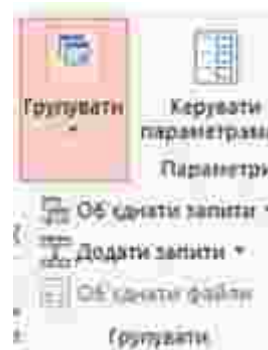


- Відкрийте редактор Power Query, натиснувши у Excel пункт «Отримання та перетворення» у головному меню.



- Запустіть процес злиття (Merge) у Power Query :

- У верхній частині вікна редактора ви побачите вкладку "Головна" (Home).
- На цій вкладці знайдіть кнопку "Групувати" (Merge Queries), яка розташована в групі "Запити" (Query).



У вікні злиття виберіть **Таблицю Клієнти** як основну та **Таблицю Замовлення** як додаткову. Для цього



натисніть кнопку на пустому вікні і виберіть її.

- З'єднайте таблиці за стовпцем **ID** (він є спільним ключем), відмітивши їх мишкою.

- Налаштуйте вигляд результатів:

- У результаті Power Query створить нову колонку, яка буде містити вкладену таблицю із Замовленнями.

ID	Ім'я	Місто
1	Іван	Київ
2	Ольга	Харків
3	Петро	Львів

ID	Товар	Сума
1	Ноутбук	1500
2	Телефон	800
3	Мишка	25

- о Натисніть на значок розгортання (▼) у вкладеній колонці, щоб вибрати поля, які необхідно показати.

5. Зберігайте результат:

- о Натисніть "Закрити та Завантажити", щоб імпортувати об'єднані дані у ваш Excel.

Результат:

Після об'єднання ви отримаєте таблицю:

1 ² ID	A ^B Ім'я	A ^B Місто	A ^B Таблица3.Товар	1 ² Таблица3.Сума
1	1 Іван	Київ	Ноутбук	1500
2	2 Ольга	Харків	Телефон	800
3	3 Петро	Львів	Мишка	25

Це злиття дозволяє об'єднати дані з різних джерел, створивши єдине представлення для подальшого аналізу.

Різниця між злиттям (**Merge**) і додаванням (**Append**) таблиць у Power Query полягає в способі їх об'єднання та кінцевому результаті:

1. Злиття (**Merge**):

- **Мета:** Об'єднання даних з різних таблиць на основі спільного стовпця (ключа).
- **Як працює:** Створюється нова таблиця, яка комбінує інформацію з обох джерел, підбираючи відповідні записи з таблиці за ключем. Це аналог функції VLOOKUP або SUMIF в Excel.
- **Приклад використання:** Якщо є таблиця з клієнтами та таблиця із замовленнями, ви можете об'єднати їх за спільним ключем (наприклад, **ID клієнта**), щоб побачити, які замовлення належать кожному клієнту.
- **Результат:** Кожен запис основної таблиці збагачується відповідними даними з додаткової таблиці.

2. Додавання (**Append**):

- **Мета:** З'єднання кількох таблиць одна за одною, щоб створити одну велику таблицю.

- **Як працює:** Дані просто додаються "вниз", створюючи нові рядки. Тут не потрібен спільний ключ.

- **Приклад використання:** Якщо є дві таблиці з продажами за різні місяці, ви можете додати їх разом, щоб створити одну таблицю з усіма продажами.

- **Результат:** Зростає кількість рядків у таблиці, але структура колонок залишається однаковою.

Порівняння у таблиці:

Критерій	Злиття (Merge)	Додавання (Append)
Мета	Об'єднання даних за ключовим полем	Об'єднання таблиць одна під одною
Ключові поля	Необхідні для поєднання	Не потрібні
Результат	Розширена таблиця	Довша таблиця
Приклад	Взаємозв'язок між таблицями клієнтів і замовлень	Комбінація даних за різні періоди

Вибір між злиттям і додаванням залежить від вашої задачі. Якщо потрібен зв'язок між даними, використовуйте **злиття**; якщо об'єднуєте дані із загальною структурою, обирайте **додавання**.

Power Query існує багато методів для обробки даних, які дозволяють ефективно трансформувати, очищати та аналізувати інформацію. Ось деякі з них:

Очищення даних

1. **Видалення дублікатів:** Легко знаходьте та видаляйте дублікати, щоб отримати унікальні значення.
2. **Заповнення порожніх значень:** Можете замінити порожні клітинки на задане значення чи попередній рядок.

3. **Очищення тексту:** Видаляйте пробіли, змінюйте регістр, замінійте або видаляйте небажані символи.

Трансформація даних

1. Розділення та об'єднання колонок:

- о Розділення тексту за роздільником (наприклад, імені та прізвища).
- о Об'єднання кількох колонок в одну (наприклад, створення повного імені).

2. Групування даних:

- о Створення агрегатних значень (сума, середнє, максимум) для груп рядків за спільним значенням.

3. Pivot і Unpivot:

- о **Pivot:** Перетворення рядків у стовпці.
- о **Unpivot:** Перетворення стовпців назад у рядки.

Робота з текстом

1. **Замінювання:** Автоматично змінюйте одні значення на інші.
2. **Витяг тексту:** Зберігайте певні символи або фрагменти тексту з рядка, наприклад, перші кілька літер.
3. **Форматування тексту:** Перетворюйте текст на великий або малий регістр, видаляйте зайві пробіли.

Дата та час

1. **Розділення дати та часу:** Виділення дня, місяця, року або години, хвилини.
2. **Розрахунки з датою:** Обчислення різниці між датами чи додавання кількості днів.

Робота з кількома джерелами даних

1. **Об'єднання кількох джерел:** З'єднання даних із баз даних, таблиць Excel, веб-сайтів, текстових файлів тощо.
2. **Динамічне оновлення джерел:** Дані автоматично оновлюються з останніх доступних даних.

Користувацькі обчислення

1. **Вирази в користувацьких стовпцях:** Використовуйте формули для створення нових значень.
2. **Використання мови М:** Створюйте власні алгоритми обробки даних для унікальних потреб.

Фільтрування

1. **Фільтрація за умовами:** Залишайте тільки рядки, які відповідають певним умовам (наприклад, значення більше 100).
2. **Видалення порожніх рядків:** Автоматично фільтруйте всі порожні рядки.

6.2. Мова М

Мова М у Power Query дозволяє створювати та налаштовувати запити за допомогою коду. Вона забезпечує більш гнучкий підхід до трансформації даних і корисна для складних операцій, які не можна виконати через інтерфейс Power Query.

М-код розширює можливості Power Query і дозволяє виконувати більш складні трансформації даних.

М-код використовується безпосередньо у Power Query, а не у вікні VBA. Щоб переглядати або редагувати М-код у Power Query, потрібно виконати наступне:

1. **Як потрапити до редактора М-коду:**
 - Завантажте дані у Power Query.
 - У вікні Power Query натисніть на вкладку **Додатковий редактор** (Advanced Editor), яка знаходиться на стрічці меню.
 - У цьому редакторі ви побачите М-код, який відповідає вашому запиту.
2. **Що робити в редакторі:**
 - Ви можете переглядати автоматично згенерований код.

- Редагувати його, щоб внести зміни, які недоступні через графічний інтерфейс.
- Додавати нові кроки чи писати користувацькі вирази.

Вигляд редактора М-коду:

Він нагадує простий текстовий редактор і виглядає, наприклад, ось так для таблиці 1 з попереднього прикладу.



- **Не плутайте VBA та М-код:** VBA (Visual Basic for Applications) використовується для автоматизації процесів в Excel, а М-код – для трансформації даних у Power Query.

- **Автоматизація:** Все, що ви налаштовуєте у Power Query, можна перезапустити, адже воно інтегрується з оновленням даних.

Ось базові приклади використання:

1. Основи М-коду

- М-код використовується у кожному Power Query запиті. Наприклад, якщо ви завантажуєте дані з файлу Excel, М-код виглядатиме так:

let

Source = Excel.Workbook(File.Contents("C:\Path\File.xlsx"), null, true),

Data = Source[[Name="Sheet1"]][Content]

in

Data

- **let** визначає послідовність операцій.
- **in** повертає кінцевий результат.

2. Додавання обчислюваних колонок

Ви можете створювати нові обчислення прямо в коді. Наприклад:

```
let
  Source = Table.FromRecords({
    [Імя]="Іван", [Вік]=30
    [Імя]="Ольга", [Вік]=25
    [Імя]="Петро", [Вік]=48
  }),
  NewColumn = Table.AddColumn(Source, "Категорія", each if [Вік] >
28 then "Старший" else "Молодший")
in
  NewColumn
```



У цьому прикладі новий стовпець **Категорія** визначається на основі віку.

3. Фільтрування даних

Ви можете відфільтрувати таблицю за певними критеріями:

```
let
  Source = Table.FromRows({{"Іван", 30}, {"Ольга", 25}, {"Петро",
35}}, {"Ім'я", "Вік"}),
  Filtered = Table.SelectRows(Source, each [Вік] > 30)
in
  Filtered
```

Це залишить лише записи, де вік більше 30.

4. Об'єднання тексту

Використовуйте функцію для об'єднання текстових значень:

```
let
    Source = Table.FromRows({"Іван", "Київ"}, {"Ольга", "Харків"}),
    {"Ім'я", "Місто"},
    Combined = Table.AddColumn(Source, "Повна назва", each [Ім'я] & "
із " & [Місто])
in
    Combined
```

У результаті отримаєте колонку з повними описами, наприклад, "Іван із Києва".

5. Створення користувацьких функцій

Ви можете створювати функції для повторного використання:

```
Multiply = (x, y) => x * y,
```

```
Result = Multiply(5, 3)
```

Ця функція множить два числа.

6.3. Підключення до баз даних

Підключення до баз даних з Excel дозволяє вам автоматично імпортувати, оновлювати та аналізувати дані, зберігаючи актуальність та точність інформації без необхідності вручну оновлювати дані. Excel підтримує кілька способів підключення до різних типів баз даних, таких як SQL Server, Access, MySQL, Oracle та інші.

Excel дозволяє підключатися до різних типів баз даних через функціонал Power Query або вбудовані підключення. Це забезпечує потужний інструмент для аналізу та обробки даних з баз даних. Залежно від типу бази даних, підключення може відрізнятися, але загальний процес полягає в наступному:

1. Вибір типу джерела даних.
2. Введення параметрів підключення.

3. Вибір таблиці або запиту для імпорту.
4. Завантаження даних у Excel для подальшого аналізу.

Ось покрокова інструкція, як підключитися до баз даних у Excel:

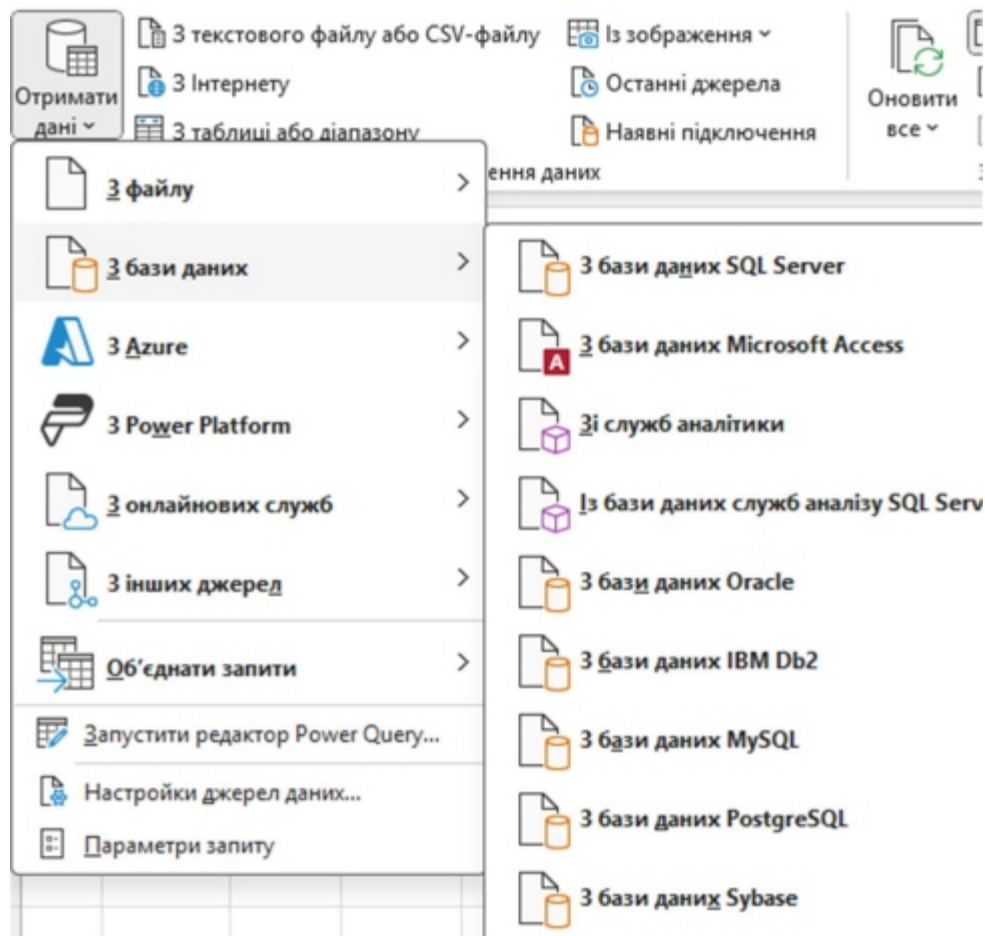
1. Підключення до бази даних SQL Server

Крок 1: Відкриття Excel

1. Відкрийте Excel.
2. Перейдіть на вкладку "Дані" (Data) на стрічці.

Крок 2: Вибір джерела даних

1. Натисніть "Отримати дані" (Get Data).
2. Виберіть "З бази даних" (From Database) → "З SQL Server" (From SQL Server Database).



Крок 3: Введення параметрів підключення

1. У діалоговому вікні введіть адресу сервера в поле "Сервер" (Server). Це може бути як IP-адреса сервера, так і доменне ім'я.

2. Якщо база даних потребує автентифікації, введіть **логін** і **пароль** в поля відповідно.
3. Натисніть **"Ок"**.

Крок 4: Вибір бази даних та таблиці

1. Після підключення Excel відобразить доступні бази даних. Виберіть потрібну базу даних.
2. Виберіть таблицю або вигляд запиту, з яким ви хочете працювати.
3. Натисніть **"Завантажити"** (Load), щоб імпортувати дані в Excel.

Крок 5: Оновлення даних

1. Для оновлення даних просто натискайте **"Оновити все"** на вкладці **"Дані"** (Data) в Excel. Це оновить всі з'єднання з базами даних.

2. Підключення до бази даних

Access

Крок 1: Відкриття Excel

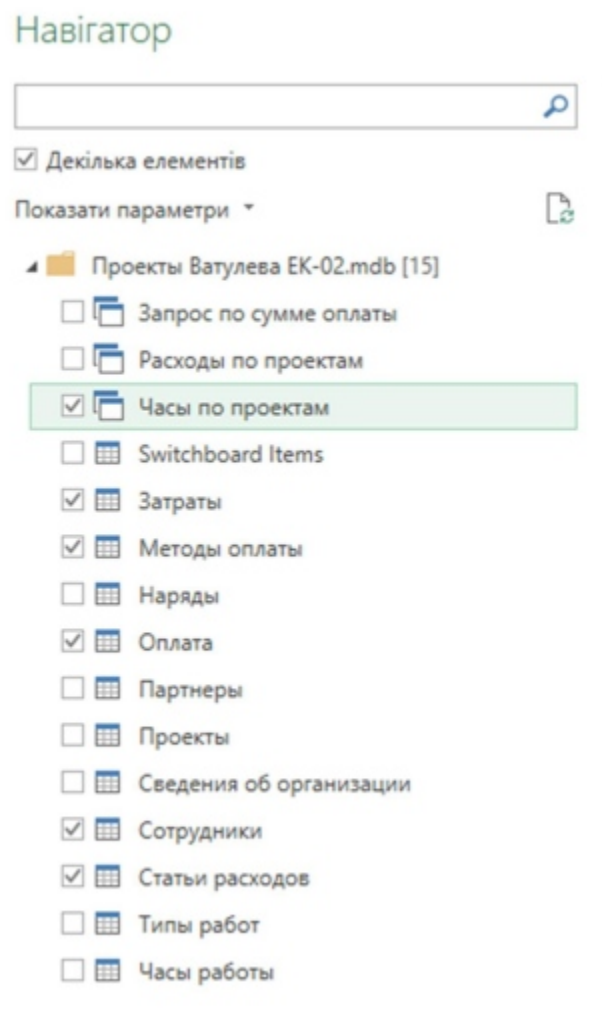
1. Відкрийте Excel.
2. Перейдіть на вкладку **"Дані"** (Data).

Крок 2: Вибір джерела даних

1. Натисніть **"Отримати дані"** (Get Data).
2. Виберіть **"З бази даних"** (From Database) → **"З Microsoft Access"** (From Microsoft Access Database).

Крок 3: Вибір файлу бази даних

1. У діалоговому вікні виберіть файл бази даних Access, до якого хочете підключитися.
2. Натисніть **"Відкрити"** (Open).

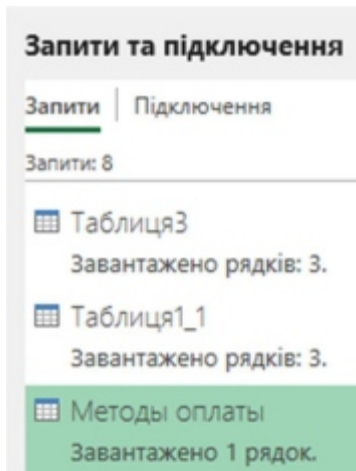


Крок 4: Вибір таблиці та завантаження

1. Excel покаже всі таблиці та запити в базі даних. Виберіть потрібну таблицю.
2. Натисніть "Завантажити" (Load), щоб імпортувати дані.

3. Тепер у правій частині вікна Excel з'явився список вибраних таблиць з бази даних. Клацнувши по будь якій із них, ви отримаєте доступ до цієї таблиці через редактор Power Query.

4. Редагування таблиці, її доповнення виконується за описаними раніше правилами через управління кнопками головного меню.



3. Підключення до бази даних MySQL

Для підключення до бази даних MySQL, вам потрібно буде встановити **MySQL ODBC Driver**, якщо він ще не встановлений.

Крок 1: Встановлення MySQL ODBC Driver

1. Перейдіть на сайт [MySQL ODBC Driver](#) і завантажте драйвер для вашої операційної системи.
2. Встановіть драйвер, слідуючи інструкціям.

Крок 2: Відкриття Excel

1. Відкрийте Excel.
2. Перейдіть на вкладку "Дані" (Data).

Крок 3: Вибір джерела даних

1. Натисніть "Отримати дані" (Get Data).
2. Виберіть "З іншого джерела" (From Other Sources) → "З ODBC" (From ODBC).

Крок 4: Налаштування підключення

1. У діалоговому вікні виберіть підключення до MySQL через встановлений ODBC драйвер.
2. Введіть потрібні параметри підключення: сервер, порт, база даних, ім'я користувача та пароль.

3. Натисніть "Ок".

Крок 5: Вибір таблиці та завантаження

1. Після підключення виберіть таблицю або вигляд запиту.
2. Натисніть "Завантажити" (Load), щоб імпортувати дані.

4. Підключення до бази даних Oracle

Крок 1: Встановлення Oracle Data Provider для .NET

Щоб підключитися до бази даних Oracle, вам потрібно мати встановлений **Oracle Data Provider for .NET (ODP.NET)**.

Крок 2: Відкриття Excel

1. Відкрийте Excel.
2. Перейдіть на вкладку "Дані" (Data).

Крок 3: Вибір джерела даних

1. Натисніть "Отримати дані" (Get Data).
2. Виберіть "З іншого джерела" (From Other Sources) → "З Oracle" (From Oracle Database).

Крок 4: Введення параметрів підключення

1. Введіть параметри підключення до Oracle бази даних (сервера, ім'я користувача, пароль).
2. Натисніть "Ок".

Крок 5: Вибір таблиці та завантаження

1. Виберіть таблицю або запит, який ви хочете імпортувати.
2. Натисніть "Завантажити" (Load).

5. Підключення через Power Query (для інших джерел)

Якщо ви хочете підключити інші бази даних або джерела, які не мають прямої інтеграції з Excel, ви можете скористатися **Power Query** для налаштування підключення:

1. Відкрийте Excel.
2. Перейдіть на вкладку "Дані" (Data).
3. Натисніть "Отримати дані" (Get Data).
4. Виберіть "З інших джерел" (From Other Sources).

5. Виберіть тип джерела даних, наприклад "З ODBC", "З API", або інші доступні варіанти.
6. Дотримуйтесь інструкцій для налаштування підключення та завантаження даних.

6.4. Робота з веб-запитами

Microsoft Excel дозволяє працювати з веб-запитами для отримання та обробки даних з інтернету. Основні способи роботи з веб-запитами в Excel:

-
1. Імпорт даних з веб-сторінки через Power Query.
 2. Використання функції WEBSERVICE (тільки для простих API-запитів).
 3. VBA для автоматизації роботи з веб-запитами.

Розглянемо, як це робиться кожним методом.

1. Імпорт даних з веб-сторінки через Power Query

Power Query дозволяє імпортувати та оновлювати дані з веб-сторінок.

Приклад: Отримання курсів валют із сайту НБУ

1. Відкрийте Excel та перейдіть до "Дані" → "Отримати дані" → "З інших джерел" → "З Інтернету".
2. Введіть URL-адресу, наприклад:
3. <https://bank.gov.ua/NBUStatService/v1/statdirectory/exchange?json>
4. Натисніть "ОК".

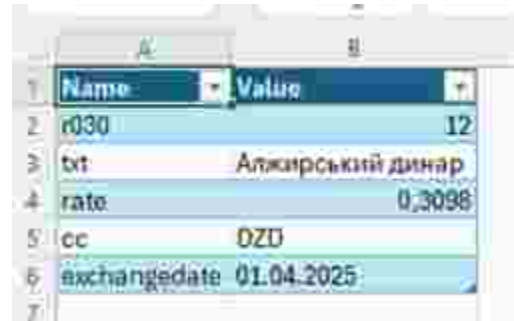


OK

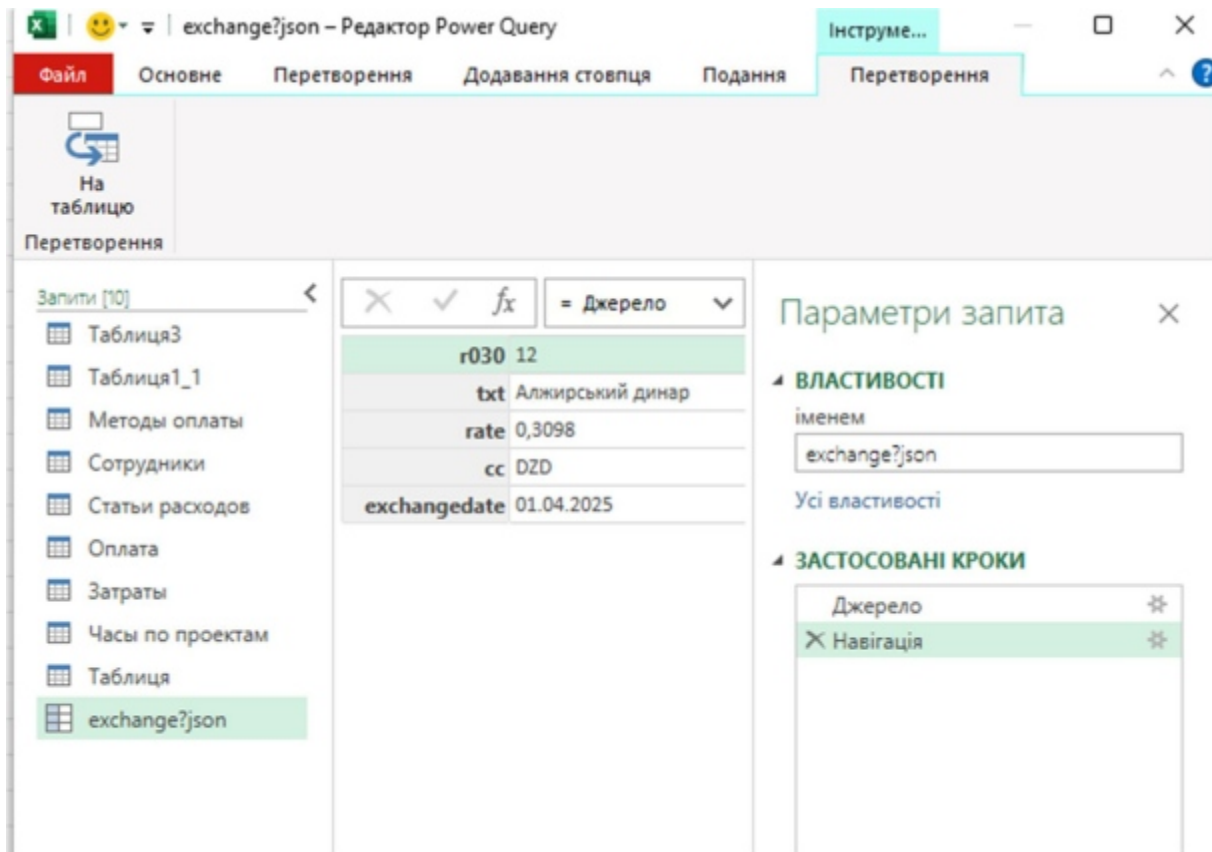
5. Excel відкриває Power Query Editor, де можна перетворювати та фільтрувати дані.

6. Натисніть "Закрити та завантажити", щоб перенести дані у таблицю Excel.

Це зручно для автоматичного оновлення даних про курси валют без ручного копіювання.



Name	Value
r030	12
txt	Алжирський динар
rate	0,3098
cc	DZD
exchangedate	01.04.2025



2. Використання функції WEBSERVICE

Якщо потрібно отримати текстові або JSON-дані з API, можна скористатися функцією WEBSERVICE(). API Exchangerate-API використовує середньозважений курс валют, який формується на основі даних із банківських

та фінансових установ по всьому світу. Це не дані конкретної біржі (наприклад, Форекс або Нью-Йоркської біржі), а агреговані середні курси.

Приклад: Отримання курсу долара через API

=WEBSERVICE("<https://api.exchangerate-api.com/v4/latest/USD>")

Тоді в клітинці з цією функцією з'являється наступне повідомлення:

```
{"provider": "https://www.exchangerate-api.com", "WARNING_UPGRADE_TO_V6": "https://www.exchangerate-api.com/docs/free", "terms": "https://www.exchangerate-api.com/terms", "base": "USD", "date": "2025-04-01", "time_last_updated": 1743465601, "rates": {"USD": 1, "AED": 3.67, "AFN": 70.58, "ALL": 91.65, "AMD": 391.7, "ANG": 1.79, "AOA": 918.57, "ARS": 1073.88, "AUD": 1.6, "AWG": 1.79, "AZN": 1.7, "BAM": 1.81, "BBD": 2, "BDT": 121.42, "BGN": 1.81, "BHD": 0.376, "BIF": 2965.85, "BMD": 1, "BND": 1.34, "BOB": 6.88, "BRL": 5.75, "BSD": 1, "BTN": 85.5, "BWP": 13.74, "BYN": 3.27, "BZD": 2, "CAD": 1.44, "CDF": 2859.43, "CHF": 0.883, "CLP": 947.87, "CNY": 7.26, "COP": 4153.18, "CRC": 496.76, "CUP": 24, "CVE": 101.95, "CZK": 23.06, "DJF": 177.72, "DKK": 6.9, "DOP": 62.64, "DZD": 133.81, "EGP": 50.57, "ERN": 15, "ETB": 129.58, "EUR": 0.925, "FJD": 2.33, "FKP": 0.774, "FOK": 6.9, "GBP": 0.774, "GEL": 2.77, "GGP": 0.774, "GHS": 15.55, "GIP": 0.774, "GMD": 72.63, "GNF": 8651.58, "GTQ": 7.72, "GYD": 209.54, "HKD": 7.78, "HNL": 25.51, "HRK": 6.97, "HTG": 131.03, "HUF": 372.12, "IDR": 16604.18, "ILS": 3.72, "IMP": 0.774, "INR": 85.5, "IQD": 1311.41, "IRR": 41999.16, "ISK": 131.93, "JEP": 0.774, "JMD": 156.41, "JOD": 0.709, "JPY": 149.8, "KES": 129.04, "KGS": 87.2, "KHR": 4005.78, "KID": 1.6, "KMF": 454.87, "KRW": 1472.88, "KWD": 0.308, "KYD": 0.833, "KZT": 504.62, "LAK": 21522.74, "LBP": 89500, "LKR": 295.53, "LRD": 199.36, "LSL": 18.33, "LYD": 4.84, "MAD": 9.63, "MDL": 17.99, "MGA": 4674.24, "MKD": 57.11, "MMK": 2099.93, "MNT": 3509.27, "MOP": 8.01, "MRU": 39.95, "MUR": 45.54, "MVR": 15.45, "MWK": 1749.73, "MXN": 20.45, "MYR": 4.44, "MZN": 63.92, "NAD": 18.33, "NGN": 1532.35, "NIO": 36.67, "NOK": 10.52, "NPR": 136.8, "NZD": 1.76, "OMR": 0.384, "PAB": 1, "PEN": 3.67, "PGK": 4.09, "PHP": 57.28, "PKR": 280.23, "PLN": 3.87, "PYG": 7954.39, "QAR": 3.64, "RON": 4.6, "RSD": 108.49, "RUB": 84.03, "RWF": 1421.46, "SAR": 3.75, "SBD": 8.13, "SCR": 14.32, "SDG": 525.86, "SEK": 10.04, "SGD": 1.34, "SHP": 0.774, "SLE": 22.8, "SLL": 22798.78, "SOS": 571.48, "SRD": 36.81, "SSP": 44.51, "STN": 22.65, "SYP": 12996.67, "SZL": 18.33, "THB": 33.95, "TJS": 10.91, "TMT": 3.5, "TND": 3.11, "TOP": 2.38, "TRY": 37.96, "TTD": 6.77, "TVD": 1.6, "TWD": 33.22, "TZS": 2641.33, "UAH": 41.42, "UGX": 3661.05, "UYU": 41.87, "UZS": 12937.1, "VES": 69.78, "VND": 25592.92, "VUV": 122.08, "WST": 2.83, "XAF": 606.5, "XCD": 2.7, "XCG": 1.79, "XDR": 0.753, "XOF": 606.5, "XPF": 110.33, "YER": 246.07, "ZAR": 18.33, "ZMW": 28.28, "ZWL": 26.77}}
```

Тут наведені курси валют відносно долара США. Кожна валюта позначена своїм міжнародно затвердженим шифром.

Але ця функція не дозволяє працювати з даними у форматі JSON або XML без додаткової обробки. Для парсингу можна використовувати FILTERXML().

Оскільки функція WEBSERVICE() повертає JSON-форматований рядок, нам потрібен спосіб дістати конкретний курс валюти.

Використання Power Query

1. Перейдіть в Excel у "Дані" → "Отримати дані" → "З інших джерел" → "З Інтернету".
2. Введіть URL: <https://api.exchangerate-api.com/v4/latest/USD>
3. Натисніть "ОК", зачекайте завантаження даних.

4. Знову Перейдіть в Excel у "Дані" → "Отримати дані" → "З таблиць або діапазону".

5. Відкриється Power Query. В цьому Виділіть клітинку з JSON-текстом.

6. Перейдіть у "Перетворення" → "Розширений редактор" (Advanced Editor).

7. Замініть Source на цей код:

let

Source = **Json.Document(Web.Contents("https://api.exchangerate-api.com/v4/latest/USD"))**

in

Source

8. Натисніть ОК.

Тепер дані повинні відобразитися у вигляді структурованих записів, і ви зможете їх розгорнути. І дивитися конкретний кус конкретної валюти

✕ ✓ fx = Json.Document(Web.Contents("https://api.exchangerate-api.com/ ▼

provider	https://www.exchangerate-api.com
WARNING_UPGRADE_TO	https://www.exchangerate-api.com/docs/free
terms	https://www.exchangerate-api.com/terms
base	USD
date	2025-04-01
time_last_updated	1743465601
rates	Record

USD	1
AED	3,67
AFN	70,58
ALL	91,65
AMD	391,7
ANG	1,79
AOA	918,57
ARS	1073,88
AUD	1,6
AWG	1,79
AZN	1,7

3. Використання VBA для автоматизації роботи з веб-запитами

VBA дозволяє робити складні запити, зокрема авторизацію та парсинг даних. Парсинг даних (від англ. *parsing* – розбір) – це процес обробки, вилучення та структурування інформації з неструктурованих або слабо структурованих джерел (текст, HTML, JSON, XML тощо).

Простіше кажучи, це коли ми отримуємо "сиру" інформацію (наприклад, текстовий JSON у одній клітинці Excel) і перетворюємо її у зручний вигляд (таблицю, список тощо).

Приклад: Отримання погоди через API OpenWeather

1. Потрібно використовувати зовнішню бібліотеку JsonConverter.bas.
2. Завантажити JsonConverter можна за посиланням: [GitHub: VBA-JSON](#)
3. Перейдіть за цим посиланням.
4. Натисніть "Download" і отримайте файл *.ZIP зі свіжою версією програми.
5. Розпакуйте цей файл, внаслідок чого утвориться папка, і знайдіть у ній файл JsonConverter.bas.
6. Відкрийте Редактор VBA (Alt + F11)
7. Перейдіть у "File" → Import File".
8. Відкриється вікно файл-менеджера в якому ви вкажете на файл JsonConverter.bas.
9. Тепер створіть кнопку з назвою «Температура у Києві» на аркуші з назвою "Weather".
10. Вставте код у модуль цієї кнопки:

Sub GetWeather()

Dim http As Object

Dim JSON As Object

Dim url As String

```

Dim city As String
Dim apiKey As String
' Встановлюємо значення для міста і API-ключа
city = "Kyiv"
apiKey = "YOUR_API_KEY" ' Вставте свій API-ключ
' Формуємо URL запиту до OpenWeather API
url = "https://api.openweathermap.org/data/2.5/weather?q=" & city &
"&appid=" & apiKey & "&units=metric"

' Створюємо об'єкт для HTTP-запиту
Set http = CreateObject("MSXML2.XMLHTTP")
On Error GoTo ErrorHandler ' Додаємо обробку помилок
' Відправляємо запит на сервер
http.Open "GET", url, False
http.Send
' Перевіряємо, чи успішно виконано запит (HTTP статус 200)
If http.Status = 200 Then
    ' Парсимо JSON відповідь
    Set JSON = JsonConverter.ParseJson(http.responseText)
    ' Перевіряємо, чи є дані для температури
    If Not JSON("main") Is Nothing And Not JSON("main")("temp") Is
Nothing Then
        ' Виводимо температуру в клітинку J11
        Sheets("Weather").Range("J11").Value = JSON("main")("temp") &
"°C"
    Else
        MsgBox "Температура не знайдена у відповіді API"
    End If
Else
    MsgBox "Помилка при отриманні даних: " & http.Status

```

End If

' Завершуємо роботу з об'єктами

Set http = Nothing

Set JSON = Nothing

Exit Sub ' Переміщаємося до кінця, щоб уникнути виклику

ErrorHandler

ErrorHandler:

' Обробка помилок

MsgBox "Виникла помилка: " & Err.Description

Set http = Nothing

Set JSON = Nothing

End Sub

Натисніть кнопку і в Excel у клітинці J11 з'явиться температура в Києві.

6.5. Індивідуальне завдання №5.

Отримання даних із зовнішніх носіїв

Завдання: Вивчити деякі операції з можливостями зміни графіків.

Порядок виконання: використовуються пункт меню Excel «Розробник» та редактор VBA.

Методичні вказівки: 1) Відкрити новий аркуш Excel, в у ньому відкрити пункт меню «Розробник».

2) Створити 2 таблиці за прикладом завдання у розділі 5.

3) Виконати їх інтеграцію через Power Query.

4) У викладача отримати файл бази даних та отримати доступ до цієї бази через Power Query.

5) Провести дії з видалення або додавання елементів до цієї бази.

6) Створити нову таблицю для цієї бази.

- 7) Виконати дії із веб-запитами щодо курсів валют через функцію WEBSERVICE, а потім через Power Query розділити JSON дані для їх перегляду.
- 8) У звіті подавати опис ваших дій з прикладами у вигляді активних вікон з виносками, що пояснюють ваші дії.

Контрольні запитання

1. Що таке Power Query і як він допомагає в інтеграції даних?
2. Які основні етапи обробки даних у Power Query?
3. Як можна імпортувати дані з різних джерел за допомогою Power Query?
4. Як створюється зв'язок між різними джерелами даних у Power Query?
5. Які типи джерел даних підтримує Power Query для імпорту?
6. Як обробляти помилки та неповні дані в Power Query?
7. Як за допомогою Power Query можна об'єднати дані з кількох таблиць?
8. Що таке мова M в Power Query і для чого вона використовується?
9. Як за допомогою мови M можна налаштувати складні трансформації даних?
10. Назвіть і поясніть основні функції мови M.
11. Як можна фільтрувати дані за допомогою мови M?
12. Як здійснити обчислення та маніпуляції з даними в мові M?
13. Як налаштувати підключення до бази даних за допомогою Power Query?
14. Які типи баз даних підтримуються для підключення через Power Query?

15. Як можна здійснити SQL-запити через Power Query для отримання даних із бази даних?
16. Як виконати підключення до бази даних через Windows Authentication і через SQL Server Authentication?
17. Як працювати з таблицями та представленнями бази даних у Power Query?
18. Як здійснити підключення до веб-сторінки або API для отримання даних в Power Query?
19. Як у Power Query працювати з JSON або XML відповідями від веб-запитів?
20. Як обробляти дані, отримані через веб-запити, в Power Query, щоб вони були готові до аналізу?
21. Як можна імпортувати дані з CSV файлів у Power Query?
22. Які переваги має використання Power Query для імпорту даних із зовнішніх носіїв порівняно з традиційними методами в Excel?
23. Як у Power Query здійснити очищення і перетворення даних після імпорту з зовнішніх файлів?
24. Як інтегрувати дані з зовнішніх джерел (наприклад, файлів Excel або CSV) для побудови єдиної таблиці в Power Query?
25. Як працювати з великими об'ємами даних, що імпортуються із зовнішніх носіїв, з метою оптимізації процесу?

У цьому розділі студенти узнали, як в Excel обробляти дані з баз даних, отримувати дані з Інтернету за допомогою Power Query та VBA.

Розділ 7.

ПРАКТИЧНІ ПРИКЛАДИ ТА КЕЙСИ

Цей розділ призначений для повторення описаних тут прийомів для створення готової програми на Excel з використанням інтерактивних методів та мови програмування VBA.

Більше прикладів ви можете отримати за наступним посиланням на «хмару», де покладені файли типу *.xlsm. І ви, зберігаючи ваш файл, маєте обирати тип .

https://myngu-my.sharepoint.com/:f:/g/personal/pistunov_i_m_nmu_one/Eru-a1J0P29JpkB2GjPItfwBqFkq9RfoCPGqtrAfH5clzQ?e=2wg8ny

7.1. Розрахунок кредитного платежу з регулюванням параметрів

Порядок роботи алгоритму:

1. Користувач вводить параметри кредиту:
 - о Сума кредиту (₴)
 - о Процентна ставка (% річних)
 - о Кількість місяців погашення
2. Натисканням кнопки "Розрахувати платіж" відбувається обчислення щомісячного платежу.
3. Використовується аннуїтетна формула для розрахунку рівномірних платежів.

Аннуїтетна формула використовується для розрахунку рівномірних платежів, які здійснюються протягом певного періоду часу, коли відома сума позики (або капіталу) і процентна ставка. Формула для розрахунку рівномірного аннуїтету виглядає так:

$$A = \frac{P \cdot i}{1 - (1 + i)^{-n}}$$

де:

- A – сума рівномірного платежу (аннуїтету) на кожен період.
- P – початкова сума позики або капіталу (основний борг).
- i – процентна ставка за період (якщо процентна ставка річна, то для щомісячних платежів її потрібно поділити на 12).
- n – кількість періодів (наприклад, кількість місяців або років, в залежності від того, як визначені періоди).

Ця формула дозволяє обчислити регулярні рівні платежі, які потрібно здійснювати для погашення боргу або інвестиції.

4. Результат виводиться у спеціальну клітинку.

Покрокова інструкція створення програми:

Крок 1: Створення форми в Excel

1. Увімкніть вкладку "Розробник":
 - Файл → Параметри → Налаштувати стрічку → Вкладка "Розробник" (поставте галочку).
2. Додайте елементи керування на аркуш:
 - Текстові поля (TextBox) для введення параметрів:
 - txtAmount (Сума кредиту), тут поле F4
 - txtRate (Процентна ставка), тут поле F5
 - txtMonths (Кількість місяців), тут поле F6
 - Кнопку (Button) "Розрахувати" для запуску макросу:
 - Призначте їй макрос CalculateLoanPayment
 - Поле виводу (Label або TextBox):
 - txtPayment (щомісячний платіж), тут поле F10

Крок 2: VBA-код для обчислення кредитного платежу

Ось VBA-код, який виконує розрахунок платежу при натисканні кнопки:

```
Sub CalculateLoanPayment()  
    Dim loanAmount As Double  
    Dim interestRate As Double  
    Dim loanTerm As Integer  
    Dim monthlyPayment As Double  
    Dim monthlyRate As Double  
    ' Отримуємо введені користувачем дані  
    loanAmount =  
CDBl(ThisWorkbook.Sheets("Розрахунки").Range("F4").Value) ' Сума  
кредиту  
    interestRate =  
CDBl(ThisWorkbook.Sheets("Розрахунки").Range("F5").Value) / 100 ' Річна  
ставка у відсотках  
    loanTerm =  
CInt(ThisWorkbook.Sheets("Розрахунки").Range("F6").Value) ' Термін у  
місяцях  
    ' Обчислюємо місячну процентну ставку  
    monthlyRate = interestRate / 12  
    ' Формула ануїтетного платежу  
    If monthlyRate > 0 Then  
        monthlyPayment = loanAmount * (monthlyRate / (1 - (1 +  
monthlyRate) ^ -loanTerm))  
    Else  
        monthlyPayment = loanAmount / loanTerm ' Якщо ставка 0%, рівні  
платежі  
    End If  
    ' Виводимо результат
```

```
ThisWorkbook.Sheets("Розрахунки").Range("F10").Value =  
Format(monthlyPayment, "#,##0.00")  
End Sub
```

Крок 3: Призначення коду елементам керування

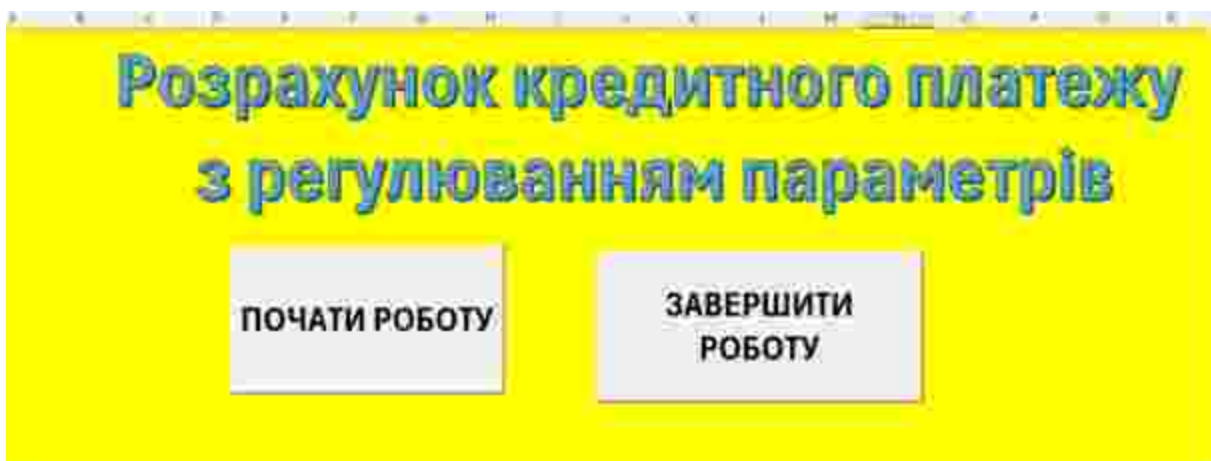
1. Відкрийте Редактор VBA (Alt + F11).
2. В Project Explorer знайдіть Worksheet (Аркуш), де розташовані кнопки.
3. Вставте код у модуль Sheet1 (або створіть новий модуль Module1).
4. Призначте макрос CalculateLoanPayment кнопці "Розрахувати".
5. Переконайтеся, що імена полів у коді збігаються з назвами в Properties.

Зауваження:

1. Іменувати поля потрібно тільки, якщо ці поля стоять у формі користувача. Якщо ні, то іменувати не потрібно.
2. Всі поля для введення даних мають бути типу «Текст».

Крок 4: Тестування

1. Введіть:
 - о Суму кредиту: 100 000 ₴
 - о Ставку: 12% річних
 - о Термін: 24 місяці
2. Натисніть кнопку "Розрахувати".
3. Отримайте щомісячний платіж у txtPayment (наприклад, 4716,36 ₴).



Уведіть у виділені клітинки наступну інформацію

○ Сума кредиту (€)	258000,15 грн
○ Процентна ставка (% річних)	8.5 %
○ Кількість місяців погашення	36,4 міс

Сума регулярного платежу становить

**РОЗРАХУВАТИ
ПЛАТІЖ**

8 144,43 грн

← Сума кредиту: 258 000,15 грн | Ставка: 8,5% річних | Термін: 36,4 міс

```

Microsoft Visual Basic for Applications - [Module1] (Code)
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Ln 2, Col 1

VBAProject (Rozrachunki.XLS)
  Microsoft Excel Objects
    Аркуш1 (Головна сторінка)
    Аркуш2 (Розрахунки)
    ЦейЛист
  Modules
    Module1

Module1
  Sub Кнопка1_Клікніть()
    Worksheets("Розрахунки").Activate
  End Sub
  Sub Кнопка2_Клікніть()
    Application.Quit
  End Sub
  Sub Стрілкавправо7_Клікніть()
    Worksheets("Головна сторінка").Activate
  End Sub
  Sub CalculateLoanPayment()
    Dim loanAmount As Double
    Dim interestRate As Double
    Dim loanTerm As Integer
    Dim monthlyPayment As Double
    Dim monthlyRate As Double
    ' Оприлюднено введені користувачем дані
    loanAmount = CDbl(ThisWorkbook.Sheets("Розрахунки").Range("F4").Value) ' Сума кредиту
    interestRate = CDbl(ThisWorkbook.Sheets("Розрахунки").Range("F5").Value) / 100 ' Річна ставка у відсотках
    loanTerm = CInt(ThisWorkbook.Sheets("Розрахунки").Range("F6").Value) ' Термін у місяцях
    ' Обчислюємо місячну процентну ставку
    monthlyRate = interestRate / 12
    ' Формула ануїтетного платежу
    If monthlyRate > 0 Then
      monthlyPayment = loanAmount * (monthlyRate / (1 - (1 + monthlyRate) ^ -loanTerm))
    Else
      monthlyPayment = loanAmount / loanTerm ' Якщо ставка 0%, рівні платежі
    End If
    ' Виводимо результат
    ThisWorkbook.Sheets("Розрахунки").Range("F10").Value = Format(monthlyPayment, "#,##0.00")
  End Sub
  
```

7.2. Автоматизація обліку витрат

Для створення автоматичного обліку витрат за допомогою Excel, VBA та елементів управління, таких як кнопки, списки і графіки, що змінюються при додаванні нових даних, можна розробити інтерактивний додаток, який дозволить вам вводити, зберігати і аналізувати дані про витрати. Ось покрокова інструкція та опис алгоритму.

Алгоритм:

1. Створення структури таблиць:

- На аркуші Excel створюється таблиця для введення витрат. Таблиця повинна містити такі стовпці:
 - Дата витрати
 - Категорія витрати
 - Сума
 - Коментар (необов'язково)

2. Додавання елементів управління (кнопок, списків):

- Додати елементи управління (кнопки для додавання нових витрат та фільтрації, списки для вибору категорій витрат).
- Додати графік для візуалізації витрат.

3. Розробка макросів для автоматизації:

- Написати VBA код для автоматичного додавання нових витрат, оновлення списків категорій та змінюваних графіків.

4. Оновлення графіків при додаванні нових даних:

- Графік повинен оновлюватися автоматично після введення нових даних.

Покрокова інструкція:

Крок 1: Створення структури таблиць

1. На аркуші Excel створіть таблицю з такими стовпцями:

- **A1:** "Дата витрати"
- **B1:** "Категорія"

- o **C1:** "Сума"
- o **D1:** "Коментар"

Це буде ваша основна таблиця для введення даних.

Крок 2: Додавання елементів управління

1. Додайте кнопки для додавання даних:

- o Виберіть вкладку "**Розробник**" (**Developer**) на панелі інструментів Excel.
- o Виберіть "**Кнопка**" (**Button**) і розмістіть її на аркуші. Назвіть кнопку "Додати витрату".
- o Додайте ще одну кнопку для оновлення графіка, якщо потрібно.

2. Додайте список для вибору категорій витрат:

- o Виберіть "**Список**" (**ComboBox**) на вкладці "**Розробник**".
- o Додайте його на аркуш для вибору категорії витрат.
- o Заповніть список категоріями витрат, наприклад, "Продукти", "Транспорт", "Розваги", "Комунальні послуги" тощо.

3. Додайте графік:

- o Виберіть діапазон даних, які ви хочете використовувати для графіка.
- o Вставте графік (наприклад, стовпчастий або круговий) для візуалізації витрат по категоріях.
- o Для автоматичного оновлення графіка використовуйте динамічні діапазони.

Якщо ви створили **поіменований діапазон** за допомогою OFFSET, вам потрібно зробити те саме для осі **X** та **Y**, а потім використовувати ці імена у графіку.

1. Створення поіменованих діапазонів для X та Y

Припустимо, у вас:

- Дати (X) у стовпці A
- Значення (Y) у стовпці C

Іменований діапазон для X-осі (дати), який назвемо DataIn

=Облік!\$A\$4:INDEX(Облік!\$A\$4:\$A\$520; COUNTA(Облік!\$A\$4:\$A\$520))

Повертає діапазон від A4 до останнього заповненого рядка.

Іменований діапазон для Y-осі (значення), який назвемо CostIn

=Облік!\$C\$4:INDEX(Облік!\$C\$4:\$C\$520; COUNTA(Облік!\$C\$4:\$C\$520))

Аналогічно, повертає діапазон від B4 до останнього заповненого рядка.

2. Як застосувати в графіку

1. **Виділіть графік**, потім перейдіть у **Вибір джерела даних** (Select Data).

2. Для **X-осі (категорії)** введіть:

3. =НазваАркушу!Ім'я_поіменованого_діапазону_X

Наприклад:

=Облік!**DataIn**

4. Для **Y-осі (ряд даних)** введіть аналогічно:

5. = НазваАркушу! Ім'я_поіменованого_діапазону_Y

6. Наприклад:

=Облік!**CostIn**

3. Динамічне оновлення графіка

Тепер, коли ви додаєте нові рядки, графік **автоматично оновлюватиметься**, оскільки поіменовані діапазони змінюються динамічно.

Крок 3: Написання макросів для автоматизації

1. **Додавання макросу для додавання витрат:**

- o Створіть макрос, який буде додавати нові витрати до таблиці.
- o Напишіть VBA код, щоб автоматично додавати дані з текстових полів у таблицю при натисканні кнопки.

```
Sub AddExpense()
```

```
Dim rowNum As Integer
```

```
Dim ws As Worksheet
```

```
Set ws = Sheets("Облік")
```

```
Dim lastRow As Long
```

```
lastRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).Row
```

```
If lastRow < 4 Then
```

```
rowNum = 4 ' Якщо немає даних або вони починаються з четвертого  
рядка
```

```
Else
```

```
rowNum = lastRow + 1 ' Знайти наступний доступний рядок
```

```
End If
```

```
' Отримуємо значення з елементів управління
```

```
Dim expenseDate As Date
```

```
Dim expenseCategory As String
```

```
Dim expenseAmount As Double
```

```
Dim expenseComment As String
```

```
expenseDate = Range("txtDate").Value ' Дата
```

```
expenseCategory = Range("cmbCategory").Value ' Категорія
```

```
expenseAmount = Range("txtAmount").Value ' Сума
```

```
expenseComment = Range("txtComment").Value ' Коментар
```

```
' Записуємо значення в таблицю
```

```
Sheets("Лист1").Cells(rowNum, 1).Value = expenseDate
```

```
Sheets("Лист1").Cells(rowNum, 2).Value = expenseCategory
```

```
Sheets("Лист1").Cells(rowNum, 3).Value = expenseAmount
```

```
Sheets("Лист1").Cells(rowNum, 4).Value = expenseComment
```

```
' Оновлюємо графік
```

```
Call UpdateChart
```

End Sub

Позначені синім рядки коду можуть не спрацювати.

Якщо VBA не розпізнає іменовані діапазони, ось кілька варіантів, як це можна виправити.

А. Використання іменованих діапазонів через Range()

Якщо у вас є іменовані діапазони (txtDate, cmbCategory тощо), але VBA їх не бачить, спробуйте:

```
expenseDate = ThisWorkbook.Names("txtDate").RefersToRange.Value
expenseCategory =
ThisWorkbook.Names("cmbCategory").RefersToRange.Value
expenseAmount =
ThisWorkbook.Names("txtAmount").RefersToRange.Value
expenseComment =
ThisWorkbook.Names("txtComment").RefersToRange.Value
```

Пояснення:

- ThisWorkbook.Names("txtDate") шукає іменований діапазон у всій книзі.
- .RefersToRange.Value отримує значення комірки.

Якщо так працює, то VBA просто не міг правильно знайти іменований діапазон стандартним способом Range("txtDate").Value.

Б. Використання абсолютних адрес (A1, B2, C3...)

Якщо VBA ігнорує іменовані діапазони, можна прямо використовувати посилання на комірки:

```
expenseDate = Sheets("Облік").Range("A1").Value
expenseCategory = Sheets("Облік").Range("B1").Value
expenseAmount = Sheets("Облік").Range("C1").Value
expenseComment = Sheets("Облік").Range("D1").Value
```

Тут важливо:

- Вказати назву аркуша, наприклад, "Облік".
- Прямо вказати комірки, якщо VBA не розпізнає іменовані діапазони.

В. Використання Evaluate() для отримання значення

Якщо іменований діапазон існує, але VBA не бачить його як Range, спробуйте Evaluate():

```
expenseDate = Evaluate("txtDate")
expenseCategory = Evaluate("cmbCategory")
expenseAmount = Evaluate("txtAmount")
expenseComment = Evaluate("txtComment")
```

Цей метод змушує Excel обчислити вираз як формулу.

2. Оновлення графіка:

Напишіть макрос для оновлення графіка, щоб він відображав нові дані після кожного додавання витрати.

```
Sub UpdateChart()
    Dim chartRange As Range
    Set chartRange = Sheets("Лист1").Range("A1:D" &
    Sheets("Лист1").Cells(Sheets("Лист1").Rows.Count, "A").End(xlUp).Row)

    ' Оновлюємо графік
    With Sheets("Лист1").ChartObjects("ExpenseChart").Chart
        .SetSourceData Source:=chartRange
    End With
End Sub
```

Крок 4: Оновлення графіка при додаванні нових даних

1. Оновлення діапазону даних для графіка:

- о Встановіть динамічний діапазон для графіка, щоб він автоматично включав нові рядки таблиці. Це можна зробити через **Іменовані діапазони**.

Наприклад, для категорії витрат можна використовувати іменований діапазон:

- о Перейдіть на вкладку **Формули** → **Менеджер імен** → **Новий**.
- о Назвіть діапазон (наприклад, "**ExpenseData**") і використовуйте динамічну формулу:

=OFFSET(Лист1!\$A\$1; 0; 0; COUNTA(Лист1!\$A:\$A); 4)

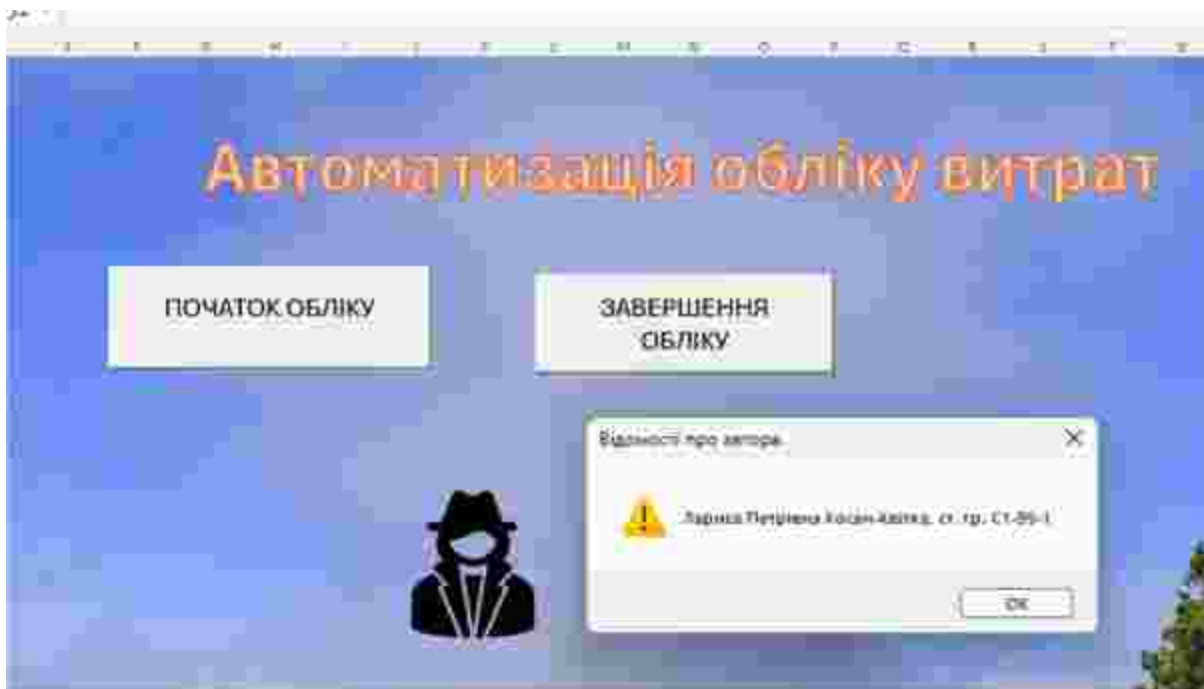
2. Оновлення графіка:

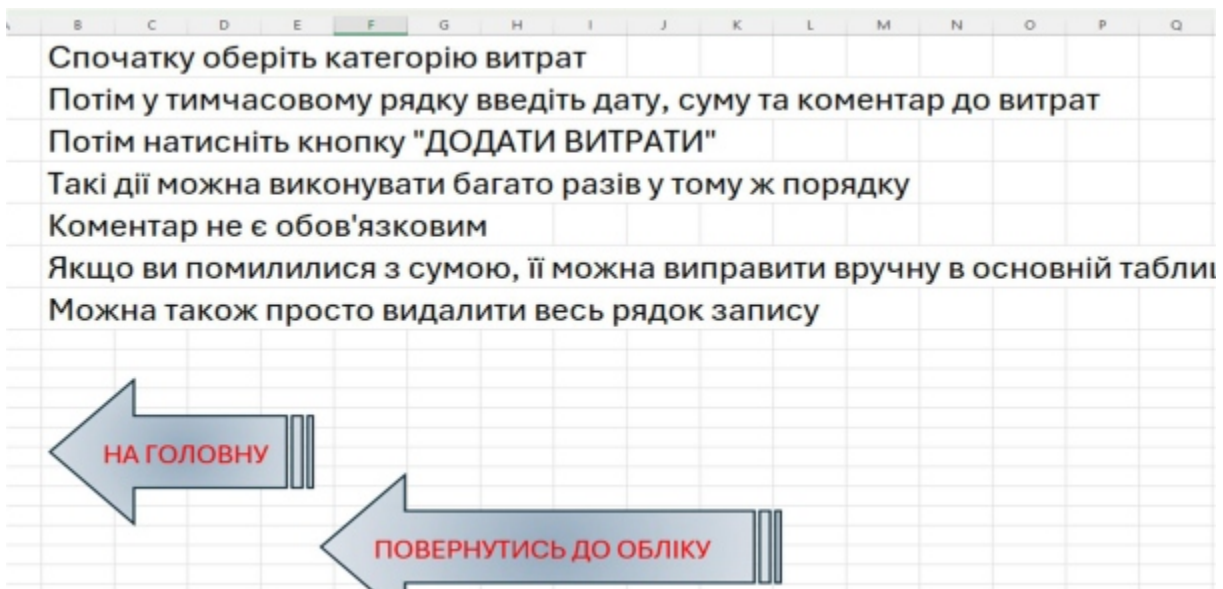
- о У властивостях графіка виберіть новий іменований діапазон як джерело даних для графіка.

Крок 5: Тестування та відлагодження

1. Переконайтеся, що кнопка "Додати витрату" працює правильно, і дані додаються в таблицю.
2. Перевірте, чи оновлюється графік після додавання нових витрат.
3. Тестуйте макроси на правильність введення даних та оновлення графіка.

Підсумок:





7.3. Динамічне планування ресурсів

Створення програми на Excel для динамічного планування ресурсів за допомогою кнопок, списків, прапорців та кодів VBA без використання форм користувача є складним, але цілком здійсненним завданням. Ось кроки для створення такої програми разом з прикладами розрахунків.

1. Планування таблиці

Перед тим як перейти до VBA і елементів управління, треба організувати дані на листі Excel. Для прикладу, у нас буде таблиця з даними про ресурси та їх використання:

Ресурс	Кількість	Вартість за одиницю	Всього використано	Вартість використаних ресурсів
Ресурс 1	100	20	50	(Розраховується)
Ресурс 2	200	15	120	(Розраховується)
Ресурс 3	150	30	90	(Розраховується)

- **Кількість** — наявні одиниці ресурсу.

- **Вартість за одиницю** — ціна одиниці ресурсу.
- **Всього використано** — кількість ресурсу, що використана.
- **Вартість використаних ресурсів** — загальна вартість використаних ресурсів (це буде розраховуватись автоматично).

2. Додавання елементів управління (форми на листі)

Крок 1: Додавання списків, що випадають

1. Виділіть клітинку, в яку хочете вставити випадаючий список (наприклад, клітинка для вибору ресурсу).
2. Перейдіть на вкладку "Дані" → "Перевірка даних".
3. Виберіть "Список" в меню **Тип даних** і вкажіть джерело списку (наприклад, діапазон з назвами ресурсів у таблиці).

Крок 2: Додавання прапорців

1. Перейдіть на вкладку "Розробник" → "Вставити" → "Прапорець".
2. Додайте прапорці на аркуш, наприклад, для включення або виключення певних ресурсів в обчисленнях.
3. Задайте відповідні клітинки, які будуть пов'язані з прапорцями для використання в формулах.

Крок 3: Додавання кнопок для запуску макросів

1. Перейдіть на вкладку "Розробник" → "Вставити" → "Кнопка".
2. Додайте кнопку і прив'яжіть її до макросу, який виконується при натисканні (наприклад, для запуску обчислень).

3. Написання макросу VBA

Приклад макросу для розрахунку вартості використаних ресурсів:

1. Натискаєте **Alt + F11** для відкриття редактора VBA.
2. У редакторі VBA створіть новий модуль та вставте наступний код:

Sub РозрахуватиВартість()

Dim ws As Worksheet

Set ws = ThisWorkbook.Sheets("Лист1") ' Замість "Лист1" вкажіть ім'я вашого листа

Dim i As Integer

```

Dim використано As Double
Dim вартість As Double
Dim загальнаВартість As Double
' Перебір усіх рядків з даними про ресурси
For i = 2 To 4 ' Замість 4 вкажіть останній рядок вашої таблиці
    If ws.Cells(i, 5).Value <> "" Then
        використано = ws.Cells(i, 4).Value ' Всього використано
        вартість = ws.Cells(i, 3).Value ' Вартість за одиницю
        загальнаВартість = використано * вартість
        ws.Cells(i, 5).Value = загальнаВартість ' Вартість використаних
ресурсів
    End If
Next i
End Sub

```

Цей макрос проходить по всіх рядках таблиці, обчислюючи вартість використаних ресурсів і записуючи результат у стовпець **"Вартість використаних ресурсів"**.

4. Розрахунки

Вам потрібно створити формули для динамічного підрахунку вартості використаних ресурсів. Для цього можна використовувати стандартні формули Excel. Наприклад:

- В клітинці E2 (вартість використаних ресурсів для Ресурсу 1):
=D2 * C2
- Далі копіюєте цю формулу вниз для всіх ресурсів.

5. Використання прапорців для вибору ресурсів

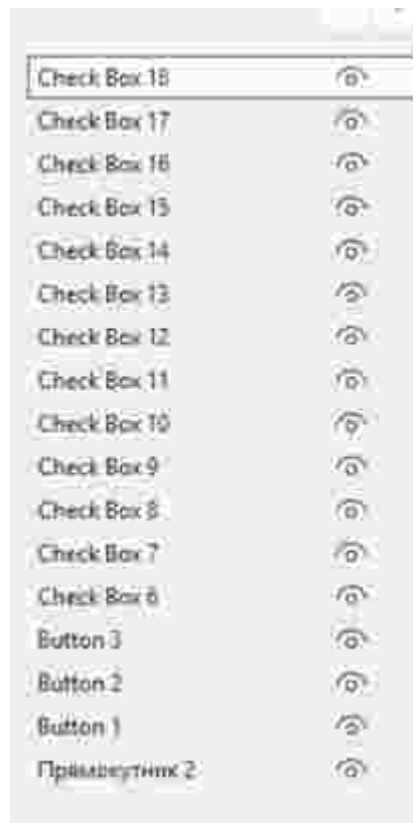
Якщо ви хочете використовувати прапорці для вибору, чи враховувати певний ресурс, ви можете налаштувати макрос, щоб він лише обчислював значення для тих ресурсів, для яких прапорці позначені.

Розставте прапорці навпроти кожного рядку таблиці. Це можна зробити шляхом копіювання одного із них. Перевірте, які номери мають імена прапорців, оскільки система їх автоматично нумерує по порядку копіювання. Всі імена починаються з CheckBox, а отім додається номер.

Перевірте, які номери мають прапорці на вашій таблиці. Це робиться так.

- ☐ Перейдіть на вкладку "Головна" (Home)
- ☐ Натисніть "Знайти й виділити" → "Область виділення" (Find & Select → Selection Pane)

☐ З'явиться список усіх об'єктів на аркуші, зокрема твоїх **Check Box'ів**



- ☐ Коли натискаєш на прапорець у списку — він підсвічується на аркуші

Далі створить кнопку, що матиме наступний макрос

Sub РозрахуватиВартістьЗПрапорцями()

Dim ws As Worksheet

Set ws = ThisWorkbook.Sheets("Аркуш1")

Dim i As Integer

Dim використано As Double

Dim вартість As Double

Dim загальнаВартість As Double

Dim прапорець As Boolean

' Перебір усіх рядків

For i = 10 To 22 ' У прикладі дані знаходяться в рядках 10-22

прапорець = ws.Shapes("CheckBox" & (i – 4)).ControlFormat.Value

' Перевірка прапорця, номери яких починаються з 6

If прапорець = 1 Then ' Якщо прапорець позначено

використано = ws.Cells(i, 4).Value

вартість = ws.Cells(i, 3).Value

загальнаВартість = використано * вартість

ws.Cells(i, 5).Value = загальнаВартість

Else

ws.Cells(i, 5).Value = 0 ' Якщо прапорець не позначено

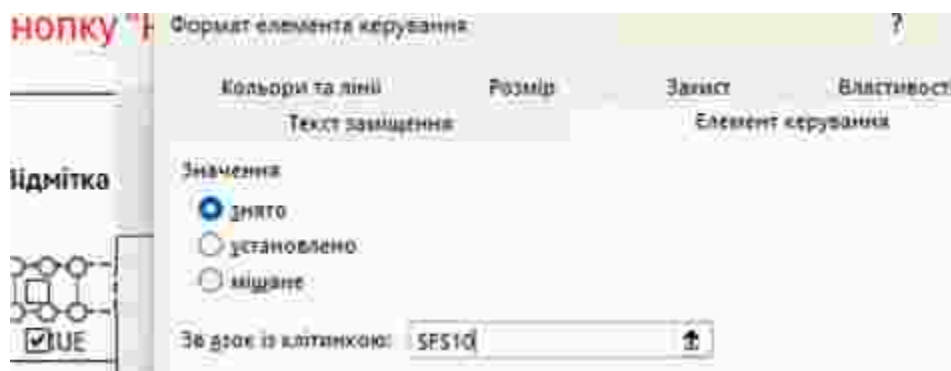
End If

Next i

End Sub

У деяких версіях Excel ця конструкція може не працювати. Тоді необхідно кожному прапорцю призначити клітинку, куди буде виводитися його стан. Для цього клацніть правою кнопкою по прапорцю, оберіть пункт «Формат елементів керування» і у вікні що відкритись оберіть закладку «Елемент керування».

Там, у вікні «Зв'язок з клітинкою» вкажіть адресу клітинки, сусідньої з прапорцем, або навіть клітинку, де стоїть прапорець.



Тоді у цій клітинці буду з'являтися стан прапорця TRUE, якщо він увімкнений, або FALSE, якщо вимкнений. Щоб їх не бачити, відмітьте всі ці клітинки та зробіть колі шрифту білий.

Тоді макрос кнопки має бути наступним.

```

Sub РозрахуватиВартістьЗПрапорцями()
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("Аркуш1")
    Dim i As Integer
    Dim використано As Double
    Dim вартість As Double
    Dim загальнаВартість As Double
    Dim прапорець As Boolean
    ' Перебір усіх рядків
    For i = 10 To 22 ' У прикладі дані знаходяться в рядках 10-22
        прапорець = ws.Range("F" & i).Value ' Перевірка стану прапорця,
номери яких починаються з 6
        If прапорець = True Then ' Якщо прапорець позначено
            використано = ws.Cells(i, 4).Value
            вартість = ws.Cells(i, 3).Value
            загальнаВартість = використано * вартість
            ws.Cells(i, 5).Value = загальнаВартість
        Else
            ws.Cells(i, 5).Value = " " ' Якщо прапорець не позначено, пробіл
        End If
    Next i
End Sub

```

6. Підсумки та використання кнопок для запуску макросів

Створіть кнопки на аркуші для запуску макросів. Наприклад, одну кнопку для розрахунку вартості ресурсів, іншу — для оновлення списків та результатів.

1. Після того, як ви додали кнопку на лист, натискаєте **Правою кнопкою миші** на кнопку → **Призначити макрос**.
2. Виберіть відповідний макрос (наприклад, **РозрахуватиВартість**).

3. Для відомостей про автора можна скористатися представленням через Web-сторінку. Ось найпростіший спосіб відкрити сторінку у браузері за

Sub OpenBrowserDefault()

Dim URL As String

URL = "<https://example.com>" ' Вкажіть потрібну сторінку

ActiveWorkbook.FollowHyperlink URL

End Sub

Просто вставте цей код у VBA-редактор і виконайте макрос. Сторінка відкриється в браузері, який встановлений за замовчуванням.

Підсумок

Завдяки використанню VBA, випадajuчих списків, прапорців і кнопок на аркуші Excel можна створити інтерактивну програму для динамічного планування ресурсів. Вона дозволяє користувачеві вибирати необхідні ресурси, вводити їх кількість та вартість, а також автоматично розраховувати вартість використаних ресурсів за допомогою кнопок для запуску макросів.

Ресурс	Кількість	Вартість за одиницю	Всього використано	Вартість використаних ресурсів	Відмітка
Матеріали	100	20	50	1000	☒
Сировина	200	15	120		☐
Обладнання	150	30	90	2700	☒
Електроенергія	214	25	162	4050	☒
Паливо	140	30	222		☐
Транспортні послуги	152	22	149	3278	☒
Інструменти	133	28	178		☐
Оренда автотранспорту	124	19	211		☐
Інформаційні ресурси	132	23	174		☐
Програми забезпечення	216	32	144	4608	☒
Технічна підтримка	185	28	227		☐
Фінансові кошти	186	33	195		☐
Комунікаційні послуги	155	18	172		☒

7.4. Інтерактивний дашборд для аналізу продажів

Задача створення **інтерактивного дашборду для аналізу продажів** — це потужний інструмент для візуалізації та управління даними, що дає змогу користувачу аналізувати важливі метрики та відслідковувати тенденції продажів у реальному часі. Створення такого дашборду в Excel з використанням **VBA, випадаючих списків, кнопок, прапорців і динамічних графіків** дозволить автоматизувати аналіз і надати інтуїтивно зрозумілий інтерфейс.

1. Опис структури даних

Приклад таблиці даних:

Дата	Продукт	Кількість	Ціна за одиницю	Сума продажу
01.01.2025	Продукт 1	10	50	500
02.01.2025	Продукт 2	5	100	500
...	...	...	...	...

Необхідні дані:

- **Дата** — для фільтрації за період.
- **Продукт** — для вибору конкретного продукту.
- **Кількість** — кількість проданих одиниць.
- **Ціна за одиницю** — ціна кожної одиниці продукту.
- **Сума продажу** — добуток кількості на ціну.

2. Основні елементи інтерфейсу дашборду

1. Випадаючі списки (ComboBox) для:

- Фільтрація даних за **періодом** (наприклад, місяць, квартал, рік).
- Вибір конкретного **продукту**.

2. Прапорці (CheckBox) для:

- Вибір метрик для відображення на графіках (наприклад, **Сума продажу** чи **Кількість**).

3. Кнопки (CommandButton) для:

- о Оновлення даних (кнопка для перерахунку даних або очищення фільтрів).

4. Лічильники для динамічної зміни значень (наприклад, введення мінімальної/максимальної ціни або кількості).

5. Динамічні графіки (Chart) для візуалізації даних у вигляді графіків, які змінюються в залежності від вибраних параметрів.

Для цього треба відмітити клітинку, де стоїть заголовок таблиці, наприклад, ДАТА, через меню Формули-Визначити ім'я надаємо йому ім'я DataIn.А у графі Посилання пишемо

= Розрахунки!\$A\$2:INDEX(Розрахунки!\$A\$2:\$A\$520;
COUNTA(Розрахунки!\$A\$2:\$A\$520))

Аналогічно, поіменуємо заголовки КІЛЬКІСТЬ та СУМА як QuoIn та SumIn, але в графах Посилання пишемо

=Розрахунки!\$C\$4:INDEX(Розрахунки!\$C\$2:\$C\$520;
COUNTA(Розрахунки!\$C\$2:\$C\$520))

=Розрахунки!\$E\$2:INDEX(Розрахунки!\$E\$2:\$E\$520;
COUNTA(Розрахунки!\$E\$2:\$E\$520))

Тепер створюємо графік і, обираючи координати для осі Х, замість того, щоб відмічати клітинки у таблиці, пишемо

= Розрахунки!DataIn ,

де «Розрахунки», це назва аркушу, де знаходиться таблиця.

Аналогічно, для суми та кількості для осі Y пишемо

= Розрахунки!QuoIn

= Розрахунки!SumIn

Оскільки числові значення кількості і суми відрізняються на декілька порядків, треба клацнути правою кнопкою мишки по одному із графіків, вибрати пункт «Формат ряду даних» і у вікні, що відкриється, вибрати «По допоміжній осі».

3. Алгоритм реалізації

Крок 1: Створення основного інтерфейсу

1. **Таблиця даних** повинна бути на одному листі (наприклад, "Дані").
2. Створюємо лист для дашборду (наприклад, "Дашборд").
3. Додати **випадаючі списки** для фільтрації за:
 - о Період (місяць, квартал, рік).
 - о Продукти.

Для цього перейдіть в "Дані" → «Знаряддя даних» → "Перевірка даних" → «Список» . Виберіть "Список" і введіть посилання на список періодів і товарів. Ці списки можуть бути навіть на іншому аркуші робочої книги. Також їх можна зробити кольору тла сторінки, щоб не заважали роботі.

Використовуємо **Data Validation** або **ComboBox** для динамічного вибору.

4. Додати **прапорці (CheckBox)** для вибору, яку метрику потрібно відобразити на графіку: "Сума продажу", "Кількість продажів" тощо.
5. Додати **кнопки (CommandButton)** для оновлення/очищення фільтрів.
6. Додати **графік** для візуалізації, який буде змінюватися залежно від вибраних фільтрів.

Крок 2: Написання коду VBA для взаємодії з елементами

1. **Очищення фільтрів** Кнопка для скидання фільтрів і оновлення дашборду.
Макроси для неї:

Sub ОчиститиФільтри()

Worksheets("Графік").Range("A8, B5").ClearContents ' Очищаємо вибір періоду та продукту, що знаходяться у клітинках A8 та B5

Call ФільтруватиДані

End Sub

2. **Фільтрація за періодом та продуктами.** Ця підпрограма пов'язана з попередньою. При виборі значення у випадаючому списку, дані мають фільтруватися відповідно:

Sub ФільтруватиДані()

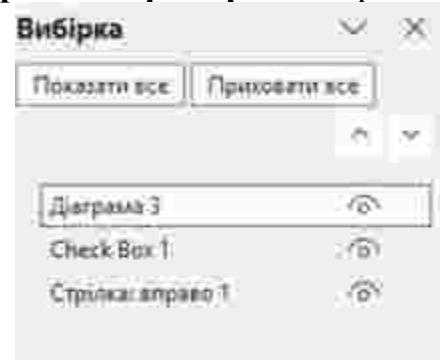
```

Dim wsData As Worksheet
Dim selectedPeriod As String
Dim selectedProduct As String
Set wsData = Worksheets("Розрахунки")
selectedPeriod = Worksheets("Графік").Range("B5").Value ' Період
вибирається у B5
selectedProduct = Worksheets("Графік").Range("A8").Value ' Назва
продукту в A8
wsData.Rows.Hidden = False ' Спочатку показуємо всі рядки
' Фільтруємо за періодом
If selectedPeriod <> "" Then
    For Each cell In wsData.Range("A2:A520")
        If Format(cell.Value, "yyyy-mm") <> selectedPeriod Then
            cell.EntireRow.Hidden = True
        End If
    Next cell
End If
' Фільтруємо за продуктом
If selectedProduct <> "" Then
    For Each cell In wsData.Range("B2:B520")
        If cell.Value <> selectedProduct Then
            cell.EntireRow.Hidden = True
        End If
    Next cell
End If

Call ОновитиГрафік ' Оновлюємо графік після фільтрації
End Sub

```

3. **Оновлення графіків на основі вибору метрики та фільтрів** Створення динамічного графіка, який змінюється в залежності від вибору метрики. Для цього потрібно узнати внутрішнє ім'я графіка, яке Excel присвоює всім об'єктам автоматично. Для цього натисніть Alt + F10. У вікні, що відкриється, знайдіть ім'я вашого графіка:



Sub ОновитиГрафік()

Dim wsData As Worksheet

Dim wsDashboard As Worksheet

Dim chart As ChartObject

Set wsData = Worksheets("Розрахунки")

Set wsDashboard = Worksheets("Графік")

Set chart = wsDashboard.ChartObjects("Діаграма3")

' Оновлюємо дані для графіка в залежності від вибору

chart.Chart.SetSourceData Source:=wsData.Range("A2:E520")

Вказуємо діапазон даних для графіка

' Додаємо залежність від прапорця

If Range("B5").Value = 1 Then

' Змінюємо назву графіка

chart.Chart.HasTitle = True

chart.Chart.ChartTitle.Text = "Кількість"

' Оновлюємо серію для "Кількості"

chart.Chart.SeriesCollection(1).Name = "Кількість"

chart.Chart.SeriesCollection(1).Values = wsData.Range("C2:C520")

chart.Chart.SeriesCollection(1).ChartType = xlColumnClustered

```
' Якщо потрібно, видаляємо інші серії
On Error Resume Next
chart.Chart.SeriesCollection(2).Delete
On Error GoTo 0

Else

' Змінюємо назву графіка
chart.Chart.HasTitle = True
chart.Chart.ChartTitle.Text = "Сума продажу"

' Оновлюємо серію для "Суми продажу"
chart.Chart.SeriesCollection(1).Name = "Сума продажу"
chart.Chart.SeriesCollection(1).Values = wsData.Range("E2:E520")
chart.Chart.SeriesCollection(1).ChartType = xlLine

' Якщо потрібно, видаляємо інші серії
On Error Resume Next
chart.Chart.SeriesCollection(2).Delete
On Error GoTo 0

End If
End Sub
```

5. Результат

Якщо потрібна кількість, поставте галочку ☒

, інакше буде сума продажів

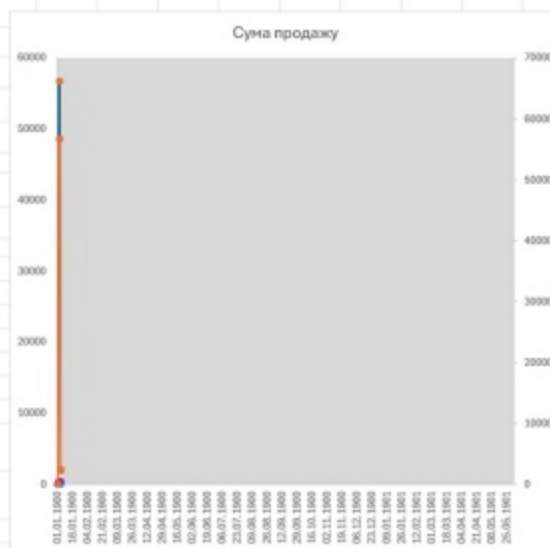
Виберіть період сортування

День

Виберіть тип товарів для сортування

Органічні продукти харчування

Оновити графік



ВИСНОВКИ

У сучасному інформаційному середовищі Excel уже давно вийшов за межі простої електронної таблиці, яка раніше асоціювалася винятково з арифметичними розрахунками та зведеними таблицями. Тепер це багатофункціональна платформа для аналізу, моделювання, автоматизації та візуалізації даних. Навчальний посібник, побудований на вивченні інтерактивних розрахунків в Excel, демонструє системний і цілеспрямований підхід до опанування цієї програми як універсального інструмента для вирішення практичних завдань будь-якої складності — від навчальних до професійно-аналітичних. Зміст посібника логічно структурований, охоплюючи як базові знання, так і складні прийоми роботи з формулами, макросами та зовнішніми джерелами даних. Від самого вступу читач знайомиться з метою вивчення інтерактивних розрахунків і отримує розуміння не лише технологічних можливостей Excel, а й того, як ці інструменти можуть використовуватись для оптимізації реальних бізнес-процесів або навчальних кейсів.

Перший розділ посібника зосереджений на підготовці середовища Excel до роботи — йдеться про налаштування доступу до додаткових функцій, зокрема підключення меню "Розробник", а також активацію Power Query та Power Pivot. Ці кроки мають ключове значення, адже саме вони відкривають користувачеві доступ до розширених можливостей, пов'язаних з обробкою великих масивів даних, створенням моделей даних і формуванням зведених таблиць на новому рівні. Такий підхід акцентує увагу не лише на навчальному аспекті, а й на практичному — на побудові робочого середовища, яке відповідає професійним вимогам.

Розділ про основи автоматизації в Excel демонструє, що справжня сила цього програмного забезпечення полягає у гнучкості формульного апарату. Тут детально розглядаються відносні, абсолютні та змішані посилання, логічні функції, робота з текстом, датами, часом, а також сучасні функції, як-от LET, LAMBDA, SEQUENCE, UNIQUE, FILTER, XLOOKUP та інші. Особливо цінним

є те, що кожна функція не просто подається як окрема команда, а аналізується її роль у вирішенні реальних задач — умовне фільтрування, сортування, динамічне обчислення та пошук значень. Це перетворює навчання на процес набуття інструментальних навичок, які одразу можна застосовувати у професійній діяльності. Таким чином, користувач не просто знайомиться з теорією, а відразу вчиться будувати функціональні моделі для аналізу.

Не менш важливою частиною є розділ про інтерактивні елементи Excel, де розглядаються засоби створення інтерфейсів із кнопками, прапорцями, перемикачами, смугами прокручування та іншими об'єктами керування. Це наближує Excel до середовища користувацького застосунку, де користувач взаємодіє не з формулами безпосередньо, а з наочними елементами, що значно підвищує ефективність та інтуїтивність взаємодії. Використання таких елементів дозволяє створювати інструменти, зрозумілі навіть тим, хто не має технічної підготовки, що є великою перевагою при побудові інтерактивних дашбордів або презентацій.

Окрему увагу у посібнику приділено мові програмування VBA, що відкриває безмежні можливості для автоматизації рутинних процесів. Завдяки вивченню синтаксису, принципів роботи з макросами, діалоговими вікнами та формами, користувач отримує змогу не просто автоматизувати задачі, а й створювати власні додатки, що реагують на дії користувача, обробляють великі обсяги даних і формують результати у зручному вигляді. Це вже не просто робота з Excel, а етап переходу до повноцінної розробки офісних рішень.

Важливою особливістю посібника є також акцент на динамічній візуалізації. Вивчаються методи створення діаграм, які змінюються залежно від вибраних параметрів, включаються або виключаються серії даних, надаючи змогу створити не просто графік, а повноцінний інструмент для візуального аналізу. Посадження кнопок керування з умовним форматуванням і смарт-графікою дозволяє зробити інтерфейс живим, адаптивним і привабливим. Це вчить не лише працювати з даними, а й доносити ці дані до аудиторії у зручному та зрозумілому форматі.

Ще однією безцінною частиною є розділ, присвячений підключенню до зовнішніх джерел даних, зокрема використанню Power Query, мови M та взаємодії з базами даних. Це дозволяє інтегрувати Excel з іншими системами, завантажувати, оновлювати та трансформувати інформацію у реальному часі. Це вже рівень BI-рішень, який є критично важливим у сучасному середовищі, де дані постійно змінюються, і потрібен інструмент, що вміє адаптуватися до цих змін без втрати гнучкості.

Фінальний розділ — це практичні кейси, які перетворюють отримані знання на навички. Тут наводяться приклади розрахунків кредитів, бюджетування, обліку витрат, створення інтерактивних дашбордів — завдань, з якими стикається практично кожен бізнес-аналітик, фінансист чи менеджер. Це дає можливість не лише перевірити себе, а й зрозуміти, як глибоке знання Excel дає конкурентну перевагу на ринку праці.

Таким чином, вивчення інтерактивних розрахунків в Excel за цим посібником не просто навчає користуватися програмою — воно формує цілісне розуміння цифрової грамотності в контексті аналітики, автоматизації та прийняття рішень. Такий підхід дозволяє збудувати міст між технічними інструментами та бізнес-задачами, роблячи Excel незамінним помічником у будь-якій сфері діяльності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. <https://support.microsoft.com/uk-ua/office/%d0%b4%d1%96%d0%b7%d0%bd%d0%b0%d0%b9%d1%82%d0%b5%d1%81%d1%8f-%d1%8f%d0%ba-%d0%b2%d0%b8%d0%ba%d0%be%d1%80%d0%b8%d1%81%d1%82%d0%be%d0%b2%d1%83%d0%b2%d0%b0%d1%82%d0%b8-power-query-%d1%82%d0%b0-power-pivot-%d1%83-excel-42d895c2-d1d7-41d0-88da-d1ed7ecc102d>
2. Walkenbach, J. (2018). *Excel 2019 Bible: The Comprehensive Tutorial Resource*. Wiley.
3. Gaskin, J. (2020). *Excel 2020 For Dummies*. For Dummies.
4. Peltier, R. (2017). *Excel Charts: The Complete Guide*. Peltier Tech.
5. Exceljet. (2023). *Excel Formulas and Functions*. Exceljet.
6. Jelen, D., & Alexander, M. (2019). *Excel 2019 Power Programming with VBA*. Wiley.
7. Lantz, B. (2016). *Excel Data Analysis: Your visual blueprint for analyzing data, charts, and PivotTables*. Wiley.
8. Cheng, S. (2018). *Master Microsoft Excel: The step-by-step guide to learning Excel for beginners*. Independently published.
9. Koller, D. (2017). *Excel Power Pivot & Power Query For Dummies*. For Dummies.
10. Bishop, C. (2016). *Excel 2016 for Business Statistics: A Guide to Solving Practical Business Problems*. McGraw-Hill Education.
11. Linton, R. (2021). *Microsoft Excel: A Comprehensive Guide to Data Analysis*. Independently published.
- 12.

ДОДАТКИ

ГАРЯЧІ КЛАВІШІ ДЛЯ EXCEL

Основні команди:

1. **Ctrl + N** – Створити нову книгу.
2. **Ctrl + O** – Відкрити існуючу книгу.
3. **Ctrl + S** – Зберегти книгу.
4. **Ctrl + P** – Друк.
5. **Ctrl + W** – Закрити книгу.
6. **Ctrl + Z** – Скасувати останню дію.
7. **Ctrl + Y** – Повторити скасовану дію.

Робота з текстом і клітинками:

8. **Ctrl + C** – Копіювати.
9. **Ctrl + X** – Вирізати.
10. **Ctrl + V** – Вставити.
11. **Ctrl + D** – Заповнити вниз.
12. **Ctrl + R** – Заповнити вправо.
13. **Ctrl + B** – Зробити текст жирним.
14. **Ctrl + I** – Зробити текст курсивом.
15. **Ctrl + U** – Підкреслити текст.

Навігація:

16. **Ctrl + стрілка** – Перехід до краю діапазону даних.
17. **Ctrl + Home** – Перейти до початку книги.
18. **Ctrl + End** – Перейти до останньої заповненої клітинки.
19. **Page Up/Page Down** – Перехід на сторінку вгору/вниз.

Виділення:

20. **Shift + стрілка** – Виділити одну клітинку.
21. **Ctrl + Shift + стрілка** – Виділити діапазон до краю даних.

22.**Ctrl + A** – Виділити всю таблицю або книгу.

Формули:

23.**Alt + =** – Автоматичне додавання формули суми.

24.**Ctrl + `** – Показати всі формули в таблиці.

25.**F2** – Редагувати активну клітинку.

Форматування:

26.**Ctrl + 1** – Відкрити вікно форматування клітинок.

27.**Ctrl + Shift + \$** – Форматувати як валюту.

28.**Ctrl + Shift + %** – Форматувати як відсотки.

29.**Ctrl + Shift + :** – Вставити поточний час.

30.**Ctrl + ;** – Вставити поточну дату.

Робота з рядками та стовпцями:

31.**Ctrl + "+"** – Вставити рядок або стовпець.

32.**Ctrl + "-"** – Видалити рядок або стовпець.

33.**Ctrl + 9** – Сховати рядки.

34.**Ctrl + 0** – Сховати стовпці.

Інші корисні команди:

35.**Alt + Enter** – Перенос тексту в клітинці на новий рядок.

36.**Ctrl + F** – Пошук.

37.**Ctrl + H** – Замінити текст.

38.**Ctrl + T** – Створити таблицю.

МЕТОДИ УСУНЕННЯ ПОМИЛОК В EXCEL

При створенні Excel-додатків, особливо складних, часто виникають типові помилки. Ось декілька з них разом з методами їх усунення:

1. Невірне використання формул

- **Помилка:** Часто виникають помилки типу #DIV/0!, #VALUE! або #REF!, коли неправильно введені формули або посилання на клітинки.
- **Метод усунення:**
 - Перевірити, чи правильні посилання на клітинки (особливо при копіюванні формул).
 - Використовувати функції обробки помилок, такі як IFERROR() або IFNA(), щоб приховати або обробити помилки.
 - Переконаватися, що формули використовують правильні типи даних.

2. Занадто складні або довгі формули

- **Помилка:** Великі, складні формули можуть бути важкими для відслідковування та налагодження, а також можуть призвести до уповільнення роботи Excel.
- **Метод усунення:**
 - Розбивати складні формули на декілька простих, щоб зробити їх більш зрозумілими.
 - Використовувати допоміжні стовпці або рядки для проміжних обчислень.
 - Використовувати функції, як-от SUMPRODUCT() або масивні формули, замість занадто складних.

3. Невірне форматування даних

- **Помилка:** Неправильне форматування чисел або дат може призвести до некоректних обчислень або відображення даних.
- **Метод усунення:**
 - Переконаватися, що всі клітинки мають правильний формат (числовий, текстовий, дата тощо).

- Використовувати стандартні формати чисел та дат.
- Для великих наборів даних використовувати функцію TEXT() для стандартизації вигляду даних.

4. Помилки в роботі з діаграмами

- **Помилка:** Неправильне створення діаграм або відсутність відповідності між даними та типом діаграми.
- **Метод усунення:**
 - Переконалися, що дані для побудови діаграм правильно організовані.
 - Вибирати правильний тип діаграми для відповідного типу даних.
 - Використовувати функцію "Прев'ю діаграми", щоб оцінити, як виглядатимуть ваші дані в графічному вигляді.

5. Перевантаження робочої книги

- **Помилка:** Використання надмірної кількості формул, даних або надмірно великих файлів може призвести до уповільнення роботи Excel.
- **Метод усунення:**
 - Мінімізувати використання важких функцій, таких як складні масиви або численні зведені таблиці.
 - Використовувати функцію **Power Query** для завантаження та обробки великих обсягів даних.
 - Очищати непотрібні стовпці та рядки.

6. Недостатньо надійне використання зведених таблиць

- **Помилка:** Невірне налаштування зведених таблиць може призвести до некоректних результатів або втрати даних.
- **Метод усунення:**
 - Переконалися, що всі поля правильно обрані та налаштовані в зведених таблицях.
 - Використовувати зведені таблиці лише для добре організованих даних.
 - Після змін у вихідних даних поновлювати зведені таблиці.

7. Невикористання абсолютних і відносних посилань

- **Помилка:** Некоректне використання абсолютних (\$A\$1) та відносних (A1) посилань може призвести до неправильних результатів при копіюванні формул.
- **Метод усунення:**
 - Використовувати абсолютні посилання для фіксованих клітинок, і відносні — для клітинок, які повинні змінюватися при копіюванні.
 - Перевіряти формули перед їх копіюванням на наявність потрібних типів посилань.

8. Невірне використання фільтрів та сортування

- **Помилка:** Невірне сортування або фільтрація даних може призвести до плутанини або до втрати даних.
- **Метод усунення:**
 - Перед сортуванням або фільтрацією завжди перевіряти, чи не виділені випадкові клітинки.
 - Використовувати функцію "Таблиця", щоб автоматично обробляти фільтри та сортування.

9. Відсутність захисту даних

- **Помилка:** Відсутність захисту робочих листів та клітинок може призвести до випадкового редагування або пошкодження даних.
- **Метод усунення:**
 - Використовувати функцію захисту листа або книги для обмеження редагування.
 - Захищати критичні клітинки, залишаючи інші для введення даних.

10. Неувага до сумісності версій Excel

- **Помилка:** Використання функцій, які не підтримуються в більш ранніх версіях Excel, може призвести до проблем з відкриттям або використанням файлів.
- **Метод усунення:**
 - Переконайтеся, що використовувані функції підтримуються в усіх версіях Excel, де буде відкриватися ваш файл.
 - Використовувати формат .xlsx, який є найбільш сумісним із новими версіями Excel.

ГОТОВІ ШАБЛОНИ ДЛЯ ЗАВАНТАЖЕННЯ

Тут представлені готові шаблони кодів на мові VBA, які достатньо скопіювати і вставити поміж рядками типу

Sub AAA()

End Sub

Якщо у наведених рядках коду вже є такі, то з шаблону, що його запропонує редактор VBA, треба видалити всі символи. А потім виправити адресу комірок чи ім'я сторінки на те, яке має бути у цьому розрахунку. Якщо у межах однієї пари вищевказаних рядків помістити декілька рядків з наведеного нижче списку, то Excel буде їх виконувати у порядку розташування, тобто, зверху вниз.

Щоб після закриття файлу Excel макроси збереглися для цього потрібно виконати кілька кроків:

Кроки для збереження файлу з макросами:

-
1. Відкрийте файл в Excel.
 2. Перейдіть до меню "Файл" (або натисніть на значок Excel у лівому верхньому кутку).
 3. Оберіть "Зберегти як".
 4. У діалоговому вікні вибору місця для збереження, а у полі "Тип файлу" виберіть "Книга Excel з макросами (*.xlsm)".
 5. Натисніть "Зберегти".

Пояснення:

-
- Формат .xlsm підтримує макроси і дозволяє їх зберігати в файлі.
 - Якщо ви зберігаєте файл у форматі .xlsx, макроси не зберігаються, оскільки цей формат не підтримує VBA-код (макроси).
-

Після цього при кожному відкритті файлу макроси будуть доступні. Якщо при відкритті файлу виникає запит на увімкнення макросів, вам потрібно підтвердити, що ви хочете їх увімкнути, щоб вони працювали.

1. Перехід на інший лист книги Excel

Worksheets("ім'я листа, на який потрібно перейти").Activate

2. Перехід на конкретну комірку у цьому ж листі

Application.Goto Range("адреса комірки")

3. Звертання до **Solver** через **VBA**.

Підпрограма **Solver** є дуже корисною для вирішення задач оптимізації в Excel, таких як лінійне і нелінійне програмування, планування ресурсів, фінансовий аналіз та інші.

Уявимо, що у нас є задача максимізувати значення в клітинці **B1**, змінюючи значення клітинок **A1** та **A2**. При цьому ми маємо обмеження на значення змінних: **A1 ≥ 0** і **A2 ≥ 0**.

Код VBA, в якому важливим є навіть відступи у рядках тексту:

Sub SolveWithVBA()

' Перевірка чи Solver активований

If Not SolverOk Then

MsgBox "Solver не активований", vbExclamation

Exit Sub

End If

' Визначення цільової клітинки, яку будемо максимізувати

**SolverOk SetCell:="B1", MaxMinVal:=1, ValueOf:=0,
ByChange:="A1,A2", Engine:="Тип методу оптимізації "**

' Додавання обмежень для змінних

SolverAdd CellRef:="A1", Relation:=3, FormulaText:="0" ' A1 >= 0

SolverAdd CellRef:="A2", Relation:=3, FormulaText:="0" ' A2 >= 0

' Запуск Solver для пошуку рішення

SolverSolve UserFinish:=True

' Повідомлення про завершення

MsgBox "Рішення знайдено!", vbInformation

End Sub

Пояснення коду:

SolverOk:

- o SetCell:="B1" — це клітинка, яку потрібно максимізувати або мінімізувати.
- o MaxMinVal:=1 — означає, що ми хочемо максимізувати значення в клітинці B1 (1 = максимізувати, 2 = мінімізувати).
- o ValueOf:=0 — це значення, яке ми хочемо досягти, якщо б ми вибрали мінімізувати чи максимізувати до конкретного значення (у даному випадку це не застосовується, бо ми просто максимізуємо).
- o ByChange:="A1,A2" — клітинки, які Solver змінюватиме, щоб досягти оптимального результату.
- o Engine: задає метод оптимізації.
 - ✓ "Simplex" — для лінійних задач.
 - ✓ "GRG Nonlinear" — для нелінійних задач.
 - ✓ "Evolutionary" — для еволюційного методу.

SolverAdd:

- o Додається обмеження для клітинки A1 ($A1 \geq 0$) і A2 ($A2 \geq 0$).
Оператор Relation:=3 означає, що це обмеження більше або рівне.
Можливі значення для параметра Relation:

1. `Relation:=1` – `<=` (менше або рівне). Це обмеження вимагає, щоб значення клітинки було менше або рівне значенню, зазначеному в

Приклад: `SolverAdd CellRef:="A1", Relation:=1, FormulaText:="10"` означає, що значення клітинки A1 повинно бути менше або рівне 10.

2. `Relation:=2` – `=` (рівно). Це обмеження вимагає, щоб значення клітинки було рівним значенню, зазначеному в `FormulaText`.

Наприклад: `SolverAdd CellRef:="A1", Relation:=2, FormulaText:="5"` означає, що значення клітинки A1 повинно бути рівним 5.

3. `Relation:=3` – `>=` (більше або рівне). Це обмеження вимагає, щоб значення клітинки було більше або рівне значенню, зазначеному в `FormulaText`.

Приклад: `SolverAdd CellRef:="A1", Relation:=3, FormulaText:="0"` означає, що значення клітинки A1 повинно бути більше або рівне 0. Тобто, A1 не може бути від'ємним числом.

4. `Relation:=4` — `>` (більше). Це обмеження вимагає, щоб значення клітинки було строго більше значення, зазначеного в `FormulaText`.

Приклад: `SolverAdd CellRef:="A1", Relation:=4, FormulaText:="1"` означає, що значення клітинки A1 повинно бути строго більше 1.

У цьому є параметрі через кому встановлюється тип змінних.

Для цілого типу – `Integer:=True`, для двійкового – `Binary:=True`.

Приклад для цілих змінних:

Ви можете задати ці обмеження для діапазону клітинок, наприклад A1:A60.

Sub SolverExampleInteger()

' Налаштовуємо Solver

SolverReset

' Встановлюємо обмеження для всіх клітинок A1:A60

SolverAdd CellRef:="A1:A60", Integer:=True **' Всі клітинки в діапазоні A1:A60 повинні бути цілими числами**

' Виконання Solver для пошуку рішення

```

        SolverOk      SetCell:="B1",      MaxMinVal:=3,      ValueOf:=0,
ByChange:="A1:A60" ' Змінімо A1:A60
        SolverSolve UserFinish:=True
End Sub

```

Приклад для двійкових змінних:

```

Sub SolverExampleBinary()
    ' Налаштовуємо Solver
    SolverReset
    ' Встановлюємо обмеження для всіх клітинок A1:A60
    SolverAdd CellRef:="A1:A60", Binary:=True ' Всі клітинки в
діапазоні A1:A60 повинні бути двійковими (0 або 1)
    ' Виконання Solver для пошуку рішення
    SolverOk      SetCell:="B1",      MaxMinVal:=3,      ValueOf:=0,
ByChange:="A1:A60" ' Змінімо A1:A60
    SolverSolve UserFinish:=True
End Sub

```

Тут адреси A1:A60 – діапазон клітинок, що змінюються для знаходження рішення.

FormulaText:

FormulaText:="0" задає значення, з яким буде порівнюватися значення в клітинці A1. Це значення буде використовуватися для визначення обмеження.

Наприклад, у команді SolverAdd CellRef:="A1", Relation:=3, FormulaText:="0" значення 0 задає порогове значення, з яким буде порівнюватися клітинка A1.

У цьому випадку, Relation:=3 означає, що A1 повинно бути більше або рівне 0 (тобто A1 не може бути від'ємним числом).

SolverSolve:

Цей метод запускає процес пошуку рішення. Параметр UserFinish:=True означає, що після завершення Solver не буде показувати діалогове вікно, а рішення буде застосовано автоматично.

MsgBox:

Повідомлення користувачу після завершення розв'язання задачі.

4. Звертання до підпрограми Regression

Ця підпрограма знаходить коефіцієнти лінійної залежності типу

$$y = a_0 + \sum_{i=1}^K a_i x_i$$

Код звертання до підпрограми, ті рядки коду, що переносяться на інший рядок тесту, треба у редакторі VBA розміщати в один рядок:

Sub RunRegressionWithHeaders()

' Перевірка, чи активована надбудова "Аналіз даних"

If Not IsAddinLoaded("Analysis ToolPak") Then

MsgBox "Надбудова 'Аналіз даних' не активована.",
vbExclamation

Exit Sub

End If

' Вказуємо діапазони X та Y змінних

Dim xRange As Range

Dim yRange As Range

' Приклад: X - стовпці A та B, Y - стовпець C

Set xRange = Range("A2:B10") ' X - значення незалежних змінних (без
заголовка)

Set yRange = Range("C2:C10") ' Y - значення залежної змінної (без
заголовка)

' Виклик регресії з Data Analysis Toolpak

Application.Run "ATPVBAEN.XLAM!Regress", _

SourceData:=yRange.Address & "," & xRange.Address, _ ' Вказуємо
Y та X діапазони

OutputRange:=Range("E1"), _ ' Місце виведення результатів
Constant:=True, _ ' Додавання константи (перехоплення)
ConfLevel:=95, ' Рівень довіри
NewWorkSheet:=False, ' Вивести результат на існуючому аркуші
Labels:=True ' Перший рядок містить заголовки

End Sub

Пояснення:

- **SourceData:=Range("A1:B10")** – вхідні дані для регресії. У цьому прикладі дані знаходяться в діапазоні A1:B10 (потрібно змінити на відповідний вашому випадку).
- **OutputRange:=Range("D1")** – це місце на аркуші, куди виводитимуться результати регресії.
- **Constant:=True** – вказує, що необхідно враховувати константу (перехоплення) в рівнянні регресії.
- **ConfLevel:=95** – рівень довіри (наприклад, 95%).
- **NewWorkSheet:=False** – означає, що результати виводяться на тому ж аркуші, а не на новому.
- **SourceData** – тут ми передаємо діапазони для Y та X змінних. Ми використовуємо адресу діапазонів yRange та xRange через .Address, що дозволяє передавати правильні діапазони.
- **Labels:=True** – цей параметр вказує Excel, що перший рядок у діапазонах X та Y містить заголовки (назви стовпців). Таким чином, Excel не буде використовувати перший рядок як частину даних для регресії.

5. Звертання до функції LINEST

Ця функція як і попередня знаходить коефіцієнти лінійної регресії, але тільки для пари стовпців – один для X, другий – для Y. Адреси клітинок подані для прикладу. В реальній задачі їх треба змінити.

Код:

```

Sub DisplayLINESTResults()
    Dim YRange As Range
    Dim XRange As Range
    Dim linEst As Variant
    Dim OutputRange As Range
    ' Визначення діапазонів
    Set YRange = Range("B1:B10")
    Set XRange = Range("A1:A10")
    Set OutputRange = Range("D1")
    ' Виклик LINEST для обчислення регресії
    linEst = Application.WorksheetFunction.LinEst(YRange, XRange, True,
True)
    ' Виведення результатів у таблицю
    OutputRange.Value = linEst(1, 1) ' Коефіцієнт регресії
    OutputRange.Offset(0, 1).Value = linEst(1, 2) ' Вільний член
    OutputRange.Offset(1, 0).Value = linEst(2, 1) ' Стандартна похибка для
коефіцієнта
    OutputRange.Offset(1, 1).Value = linEst(2, 2) ' Стандартна похибка для
вільного члену
End Sub

```

Пояснення коду:

YRange і XRange – це діапазони клітинок, які містять дані для залежної та незалежної змінної.

Application.WorksheetFunction.LinEst(YRange, XRange, True, True) – це виклик функції LINEST, яка обчислює лінійну регресію. Параметри:

- Перший True — включити константу (перехоплення).
- Другий True — повернути додаткові статистичні дані (похибки, R-квадрат тощо).

6. Звертання до підпрограми Descriptive Statistics

Ця підпрограма, що за статистичною вибіркою для декількох типів даних, (наприклад, доходи, витрати, штрафи, прибуток) розраховує такі параметри як:

1. Кількість спостережень (Count):
2. Сума (Sum): Сума всіх значень у вибраному діапазоні.
3. Середнє значення (Mean): Середнє арифметичне значення для вибраних даних.
4. Стандартне відхилення (Standard Deviation): Міра розсіювання значень навколо середнього. Визначає, як дані варіюються від середнього значення.
5. Стандартна похибка середнього (Standard Error): Стандартна похибка середнього (SE) вимірює, наскільки середнє значення вибірки може відрізнятись від справжнього середнього в популяції.
6. Мінімум (Minimum): Найменше значення в наборі даних.
7. 25-й перцентиль (1st Quartile, Q1): Значення, яке розподіляє дані на чверті, де 25% спостережень менше цього значення.
8. Медіана (Median): Середнє значення в упорядкованому наборі даних, коли 50% значень менше, а 50% — більші. 75-й перцентиль (3rd Quartile, Q3): Значення, яке відокремлює верхні 25% даних від решти.
9. Максимум (Maximum): Найбільше значення в наборі даних.
10. Діапазон (Range): Різниця між максимальним і мінімальним значенням у наборі даних.
11. Сума квадратів відхилень (Variance): Міра варіативності, що показує середнє квадратів відхилень кожного значення від середнього.
12. Коефіцієнт асиметрії (Skewness): Міра симетрії розподілу. Якщо значення асиметрії більше нуля, розподіл зміщений вправо (з більшими значеннями). Якщо менше нуля — зміщений вліво.
13. Коефіцієнт ексцесу (Kurtosis): Міра "пікованості" розподілу. Високе значення ексцесу вказує на те, що розподіл має високий пік (тобто більша кількість значень зібрана навколо середнього), низьке — вказує на "плоский" розподіл.

14. Сума значень квадратів (Sum of Squares): Відображає суму квадратів відхилень значень від середнього.

15. (Довіра до середнього).

Код:

```
Sub RunDescriptiveStatistics()  
    ' Перевірка, чи активована надбудова "Аналіз даних"  
    If Not IsAddinLoaded("Analysis ToolPak") Then  
        MsgBox "Надбудова 'Аналіз даних' не активована.",  
vbExclamation  
        Exit Sub  
    End If  
    ' Визначаємо діапазони для входу та виходу  
    Dim inputRange As Range  
    Dim outputRange As Range  
    ' Задаємо діапазони  
    Set inputRange = Range("A2:A20") ' Вхідний діапазон (дані для  
аналізу)  
    Set outputRange = Range("C2") ' Діапазон для виведення  
результатів  
    ' Виклик інструмента "Descriptive Statistics" з пакету "Аналіз  
даних"  
    Application.Run "ATPVBAEN.XLAM!DescriptiveStatistics", _  
        inputRange.Address, _ ' Вхідний діапазон даних  
        True, ' Чи включати заголовки (True означає, що перший рядок –  
заголовки)  
        True, ' Вивести описову статистику (Summary Statistics)  
        False, ' Чи розраховувати інтервал довіри для середнього (False —  
не розраховувати)  
        False, ' Чи розраховувати кореляцію  
        False, ' Чи розраховувати похибку стандартного відхилення
```

False, ' Чи розраховувати варіацію

outputRange.Address ' Місце для виведення результатів

End Sub

Пояснення коду:

✓ **inputRange.Address** – діапазон для даних, для яких треба виконати описову статистику. У прикладі це діапазон **A2:A20**.

✓ **True** (параметр після діапазону) – вказує, що перший рядок містить заголовки. Якщо заголовки відсутні, поставте **False**.

✓ **True** (параметр після заголовків) – розраховувати описову статистику, яка включає основні показники (середнє, стандартне відхилення і т.д.).

✓ Параметри після **inputRange.Address** дозволяють вам вказати, які статистичні показники ви хочете отримати.

✓ **ATPVBAEN.XLAM!DescriptiveStatistics** — це шлях до функції для запуску інструмента **Descriptive Statistics**.

7. Звертання до підпрограми Correlation

Програма розраховує парні коефіцієнти кореляції для вибірки, представленої таблицею з розташуванням даних по стовпцях.

Код:

Sub RunCorrelationAnalysis()

' Перевірка, чи активована надбудова "Аналіз даних"

If Not IsAddinLoaded("Analysis ToolPak") Then

MsgBox "Надбудова 'Аналіз даних' не активована.",
vbExclamation

Exit Sub

End If

' Визначаємо діапазони для входу та виходу

```

Dim inputRange As Range
Dim outputRange As Range
' Задаємо діапазони
Set inputRange = Range("A2:B10") ' Вхідний діапазон для двох
змінних (X та Y)
Set outputRange = Range("D2") ' Місце для виведення результатів
' Виклик інструмента "Correlation" з пакету "Аналіз даних"
Application.Run "ATPVBAEN.XLAM!Correlation", _
    InputRange:=inputRange.Address, _ ' Вхідний діапазон для
обчислення кореляції
    OutputRange:=outputRange.Address, _ ' Місце для виведення
результатів
    Labels:=True, ' Чи містить перший рядок заголовки
(True - містить заголовки)
    OutputMatrix:=True ' Вивести матрицю кореляцій
End Sub

```

Пояснення:

✓ InputRange:=inputRange.Address — це діапазон, який містить дані для обчислення кореляції. У прикладі це A2:B10, тобто два стовпці (наприклад, для двох змінних X і Y).

✓ OutputRange:=outputRange.Address — діапазон, де будуть виведені результати кореляції.

✓ Labels:=True — якщо ви вказали вхідні дані, де перший рядок містить заголовки (наприклад, назви змінних), встановіть це значення в True. Якщо заголовків немає, ставте False.

✓ OutputMatrix:=True — якщо True, буде виведена матриця кореляцій для кожної пари змінних у вибраному діапазоні. Якщо вам потрібна просто одна кореляція між двома змінними, можна встановити це значення в False.

✓ **Labels:=True:** Якщо ваші дані мають заголовки в першому рядку, ця опція дозволить програмі правильно розпізнати назви змінних. В іншому випадку, якщо заголовки відсутні, вкажіть **False**.

✓ **OutputMatrix:=True:** Якщо ви працюєте з кількома змінними та хочете отримати кореляційну матрицю для всіх пар, залиште **True**. Якщо ж вам достатньо кореляції для двох змінних, встановіть **False**.

8. Звертання до будь-якої функції

В Excel VBA ви можете звертатися до будь-якої функції або підпрограми, яка доступна в Excel, за допомогою методу **Application.Run**. Це дає змогу викликати функції або макроси, зокрема ті, які знаходяться в надбудовах, таких як **Data Analysis Toolpak** або інші інструменти Excel.

Основний синтаксис:

Application.Run "Назва_функції", [параметри]

Також вбудовані функції Excel можна викликати через VBA, використовуючи функцію **Application.WorksheetFunction**. Наприклад, для використання функції **SUM**, **AVERAGE** чи **VLOOKUP** можна використовувати наступний синтаксис:

Приклад 1: Використання функції **SUM**:

```
Sub UseSumFunction()
```

```
    Dim result As Double
```

```
    result = Application.WorksheetFunction.Sum(Range("A1:A10"))
```

```
    MsgBox "Сума значень в діапазоні A1:A10: " & result
```

```
End Sub
```

Приклад 2: Використання функції **AVERAGE**:

```
Sub UseAverageFunction()
```

```
Dim result As Double  
result = Application.WorksheetFunction.Average(Range("B1:B10"))  
MsgBox "Середнє значення в діапазоні B1:B10: " & result  
End Sub
```

Результат роботи функції, яку викликаєте через VBA, можна вивести в різні місця в Excel, залежно від того, як ви налаштуєте параметри OutputRange (діапазон для результатів). Ось кілька можливих варіантів, куди буде виведено результат:

Розглянемо приклад використання простої функції, такої як SUM (сумування значень у діапазоні клітинок), у VBA.

Приклад 1: Використання функції SUM у VBA для обчислення суми значень у діапазоні клітинок:

Код VBA:

```
Sub SumExample()  
    ' Задаємо діапазон клітинок для обчислення суми  
    Dim inputRange As Range  
    Set inputRange = Range("A1:A5") ' Вхідний діапазон для суми  
    (A1:A5)  
  
    ' Обчислюємо суму за допомогою функції SUM  
    Dim result As Double  
    result = Application.WorksheetFunction.Sum(inputRange)  
  
    ' Виводимо результат в клітинку B1  
    Range("B1").Value = result  
End Sub
```

9. Зміна налаштувань вікна Excel

Щоб змінити представлення вікна Excel через VBA, ви можете приховати меню та зробити вікно максимально компактним, залишивши лише рамку. Для цього можна скористатися властивостями вікна через VBA.

У цьому коді використовується перевірка для того, щоб перемикати стан вікна між повноекранним і нормальним режимом залежно від того, в якому режимі знаходиться вікно. Якщо вікно не в повноекранному режимі, воно переходить у цей режим; якщо вже в повноекранному режимі, воно повертається до звичайного вигляду.

Sub ToggleWindowDisplay()

' Перевіряємо, чи вже включено повноекранний режим

If Application.DisplayFullScreen = False Then

' Включаємо повноекранний режим та ховаємо інтерфейс

Application.DisplayFullScreen = True

Application.CommandBars("Ribbon").Visible = False

Application.DisplayFormulaBar = False

Application.DisplayStatusBar = False

Else

' Відновлюємо нормальний вигляд вікна

Application.DisplayFullScreen = False

Application.CommandBars("Ribbon").Visible = True

Application.DisplayFormulaBar = True

Application.DisplayStatusBar = True

End If

End Sub

Пояснення:

- **Application.DisplayFullScreen = True:** Включає повноекранний режим, де вікно займає весь екран без меню та панелей інструментів.
- **Application.CommandBars("Ribbon").Visible = False:** Приховує стрічку (Ribbon), де знаходяться основні меню Excel.

- `Application.DisplayFormulaBar = False`: Вимикає відображення панелі формул.
 - `Application.DisplayStatusBar = False`: Вимикає відображення рядка стану Excel внизу.
 - `Application.DisplayFullScreen = False`: Вимикає повноекранний режим і повертає звичайний вигляд вікна.
 - `Application.CommandBars("Ribbon").Visible = True`: Показує стрічку меню (Ribbon).
 - `Application.DisplayFormulaBar = True`: Показує панель формул.
 - `Application.DisplayStatusBar = True`: Відновлює рядок стану.
-

10. Вимкнення вікна Excel

Код:

Sub CloseExcel()

Application.Quit

End Sub

`Application.Quit` – команда, яка закриває Excel і всі відкриті робочі книги. Якщо є незбережені зміни, Excel запитає, чи потрібно їх зберегти перед закриттям.

Якщо ви хочете закрити Excel без запиту про збереження незбережених змін, можна використовувати код, щоб вимкнути це попередження, а потім викликати `Application.Quit`.

Код:

Sub CloseExcelWithoutPrompt()

Application.DisplayAlerts = False ' Вимикає попередження

Application.Quit ' Закриває Excel без запиту на збереження

Application.DisplayAlerts = True ' Включає попередження назад

End Sub

11. Використання UserForm для вставки тексту, картинки та кольорів

Для досягнення бажаного результату можна створити **UserForm**, на якому можна додавати текст і картинки, а також змінювати колір тексту.

Кроки для створення UserForm:

1. У редакторі VBA натисніть **Insert** → **UserForm**.
2. Додайте на форму:
 - о **Label** (мітка) для тексту.
 - о **Image** (картинка) для зображення.
3. Можна змінювати властивості для елементів на формі, наприклад, колір тексту або шрифт.

Приклад коду для UserForm:

1. Створіть UserForm з елементами **Label** і **Image**.
2. У властивостях Label змініть параметри **Font** (шрифт) і **ForeColor** (колір тексту).
3. Завантажте картинку в елемент Image.

Код для UserForm:

vba

Копіювати

```
Private Sub UserForm_Initialize()
```

```
    ' Задаємо текст у мітці
```

```
    Label1.Caption = "Це кастомізоване повідомлення."
```

```
    ' Задаємо колір тексту (червоний)
```

```
    Label1.ForeColor = RGB(255, 0, 0)
```

```
    ' Завантажуємо картинку
```

```
    Image1.Picture = LoadPicture("C:\Шлях_до_картинки\your_image.jpg")
```

End Sub

```
Sub ShowCustomMessage()
```

```
    ' Відкриваємо UserForm
```

```
    UserForm1.Show
```

```
End Sub
```

У цьому прикладі:

- **Label1.ForeColor = RGB(255, 0, 0)** — встановлює червоний колір для тексту.
 - **Image1.Picture = LoadPicture("C:\Шлях_до_картинки\your_image.jpg")** — завантажує картинку на форму.
-

12. Типові іконки Excel

В Excel є кілька типових іконок, які використовуються в діалогових вікнах або повідомленнях, таких як **MsgBox** для відображення повідомлень користувачам. Ці іконки є частиною стандартного набору, доступного у VBA.

Нижче наведено список типових іконок, які можна використовувати в **MsgBox** та інших діалогових вікнах у Excel:

Типи іконок у MsgBox:

1. **Іконка інформації** (використовується для повідомлень з інформацією)

- o **vbInformation**

Це стандартна іконка у вигляді синього кола з літерою "i".

MsgBox "Це повідомлення з іконкою інформації.", vbInformation, "Інформація"

2. **Іконка попередження** (використовується для попереджень або важливих повідомлень)

- o **vbExclamation**

Іконка у вигляді жовтого знака оклику (!).

MsgBox "Це повідомлення з іконкою попередження.", vbExclamation, "Попередження"

3. **Іконка помилки** (використовується для повідомлень про помилки)

o **vbCritical**

Іконка червоного кола з білим хрестиком.

MsgBox "Це повідомлення з іконкою помилки.", vbCritical, "Помилка"

4. **Іконка запитання** (використовується для запитів до користувача)

o **vbQuestion**

Іконка у вигляді синього питання.

MsgBox "Це повідомлення з іконкою запитання.", vbQuestion, "Запитання"

5. **Іконка без іконки** (якщо ви не хочете використовувати іконку взагалі)

- o **vbDefaultButton1, vbDefaultButton2, vbDefaultButton3** (вказує на те, яка кнопка є за замовчуванням, але не впливає на іконку)

Якщо не вказана іконка, то **MsgBox** не буде використовувати жодної іконки.

MsgBox "Це повідомлення без іконки.", vbOKOnly, "Без іконки"

Окрім типів іконок, можна налаштувати і типи кнопок у **MsgBox**, використовуючи відповідні параметри:

- **vbOKOnly** – лише кнопка OK
- **vbOKCancel** – кнопки OK і Cancel
- **vbYesNo** – кнопки Yes і No
- **vbRetryCancel** – кнопки Retry і Cancel
- **vbAbortRetryIgnore** – кнопки Abort, Retry, Ignore

Приклад коду з іконкою та кнопками:

```
Dim response As VbMsgBoxResult
response = MsgBox("Ви хочете зберегти зміни?",
vbYesNo + vbQuestion, "Запитання")
If response = vbYes Then
    MsgBox "Зміни збережено."
Else
```

MsgBox "Зміни не збережено."

End If

Типові іконки для **MsgBox**:

1. **vbInformation** – інформація.
2. **vbExclamation** – попередження.
3. **vbCritical** – помилка.
4. **vbQuestion** – запитання.
5. **vbDefaultButton1, vbDefaultButton2, vbDefaultButton3** – для кнопок без іконок.

Навчальне видання

Пістунов Ігор Миколайович

Smart Excel. Керовані розрахунки

Навчальний посібник

Видано в авторській редакції.

Електронний ресурс.
Підписано до видання 11.06.2024. Авт. арк. 13,9

Підготовлено до видання
в Національному технічному університеті «Дніпровська політехніка».
Свідоцтво про внесення до Державного реєстру ДК № 1842 від 11.06.2004.
49005, м. Дніпро, просп. Дмитра Яворницького, 19.