

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МИКОЛАЇВСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ**

**ФАКУЛЬТЕТ МЕНЕДЖМЕНТУ
КАФЕДРА ЕКОНОМІЧНОЇ КІБЕРНЕТИКИ, КОМП'ЮТЕРНИХ НАУК ТА
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

ІНФОРМАТИКА ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Конспект лекцій
для здобувачів першого (бакалаврського) рівня вищої освіти ОПП
«Харчові технології» спеціальності G13 «Харчові технології»
денної форми здобуття вищої освіти



Миколаїв
2025

УДК 004.9:004.4

I-74

Друкується за рішенням науково-методичної комісії факультету менеджменту Миколаївського національного аграрного університету від 24 квітня 2025 року, протокол №8.

Укладачі:

- С. І. Тищенко - канд. пед. наук, доцент, доцент кафедри економічної кібернетики, комп'ютерних наук та інформаційних технологій, Миколаївський національний аграрний університет;
- В. В. Співак - асистент кафедри економічної кібернетики, комп'ютерних наук та інформаційних технологій, Миколаївський національний аграрний університет;

Рецензенти:

- О. С. Садовий - канд. техн. наук, доцент, завідувач кафедри агроінженерії Миколаївського національного аграрного університету
- Ю. В. Грицук - канд. техн. наук, доцент кафедри загальної інженерної підготовки Донбаської національної академії будівництва і архітектури

Інформатика та інформаційні технології : конспект лекцій / уклад. С. І. Тищенко, В. В. Співак. Миколаїв : МНАУ, 2025. 114 с.

Конспект лекцій призначений для вивчення теоретичних та практичних основ інформатики та інформаційних технологій, ознайомлення з основними поняттями, принципами та інструментами, забезпечення глибокого засвоєння специфіки застосування інформаційних технологій у професійній діяльності, опанування методів обробки, аналізу та захисту інформації, формування розуміння ролі інформаційних систем у сучасному світі. Містить навчальні матеріали з основних тем курсу «Інформатика та інформаційні технології», що передбачені освітньо-професійною програмою «Харчові технології» підготовки здобувачів першого (бакалаврського) рівня вищої освіти спеціальності G13 «Харчові технології», галузі знань G – "Інженерія, виробництво та будівництво".

УДК 004.9:004.4

© Миколаївський національний
аграрний університет, 2025

ЗМІСТ

ПЕРЕДМОВА.....	4
ЗМІСТОВИЙ МОДУЛЬ 1	5
ОСНОВНІ ПОНЯТТЯ ПРО ІНФОРМАТИКУ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ, MICROSOFT OFFICE	5
ТЕМА 1.1. ВСТУП ДО ІНФОРМАТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ	5
ТЕМА 1.2. КОДУВАННЯ ТА ВИМІРЮВАННЯ ІНФОРМАЦІЇ	8
ТЕМА 1.3. КОДУВАННЯ ГРАФІЧНОЇ ТА ЗВУКОВОЇ ІНФОРМАЦІЇ.....	12
ТЕМА 1.4. АРХІТЕКТУРА КОМП'ЮТЕРА.....	19
ТЕМА 1.5. СИСТЕМНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.....	24
ТЕМА 1.6. ПРИКЛАДНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ТА ЛІЦЕНЗУВАННЯ	29
ТЕМА 1.7. ТЕКСТОВИЙ ПРОЦЕСОР MS WORD: ОСНОВИ РОБОТИ.....	35
ТЕМА 1.8. ТЕКСТОВИЙ ПРОЦЕСОР MS WORD: ФОРМАТУВАННЯ ТА ОБ'ЄКТИ.....	40
ЗМІСТОВИЙ МОДУЛЬ 2	49
ТАБЛИЧНИЙ ПРОЦЕСОР MS EXCEL ТА СУБД ACCESS.....	49
ТЕМА 2.1. ТАБЛИЧНИЙ ПРОЦЕСОР MS EXCEL: ОСНОВИ РОБОТИ	49
ТЕМА 2.2. ТАБЛИЧНИЙ ПРОЦЕСОР MS EXCEL: ФОРМУЛИ ТА ФУНКЦІЇ	55
ТЕМА 2.3. ТАБЛИЧНИЙ ПРОЦЕСОР MS EXCEL: АНАЛІЗ ТА ВІЗУАЛІЗАЦІЯ ДАНИХ.....	63
ТЕМА 2.4. БАЗИ ДАНИХ ТА СИСТЕМИ УПРАВЛІННЯ БАЗАМИ ДАНИХ (СУБД).....	70
ТЕМА 2.5. СУБД MS ACCESS: ОСНОВИ РОБОТИ.....	78
ТЕМА 2.6. СУБД MS ACCESS: ЗАПИТИ, ФОРМИ, ЗВІТИ	87
ТЕМА 2.7. ОСНОВИ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ.....	94
ТЕМА 2.8. ПРАВОВІ ТА ЕТИЧНІ АСПЕКТИ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ.....	102
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	113

Передмова

Курс дисципліни «Інформатика та інформаційні технології» призначений для забезпечення здобувачів вищої освіти спеціальності G13 «Харчові технології» фундаментальними знаннями та практичними навичками у сфері сучасних інформаційних технологій, необхідними для ефективної професійної діяльності в умовах цифровізації. Курс має на меті ознайомити здобувачів вищої освіти з основними поняттями інформатики, принципами роботи з інформацією, опанування ними сучасних засобів та інструментів інформаційних технологій, формування розуміння ролі та можливостей ІТ у різних сферах життя, науки та професійної діяльності.

Предметом вивчення дисципліни «Інформатика та інформаційні технології» є загальні принципи роботи з інформацією, методи її збору, зберігання, обробки, передачі та захисту, основи алгоритмізації та програмування, архітектура комп'ютерних систем, функціонування комп'ютерних мереж, а також сучасні програмні та апаратні засоби реалізації інформаційних процесів та технологій.

Об'єктом вивчення дисципліни є дані, інформація, алгоритми, програмне забезпечення (операційні системи, прикладні програми, системи управління базами даних тощо), апаратні засоби обчислювальної техніки, комунікаційні технології (мережі, Інтернет), засоби розробки та використання інформаційних технологій, а також інформаційні процеси, що реалізуються за допомогою цих технологій.

Викладання дисципліни ставить за **мету** сформувати високий рівень інформаційної компетентності, розвинути навички ефективного використання сучасних інформаційних технологій для розв'язання типових та професійно-орієнтованих завдань, сприяти розвитку логічного та алгоритмічного мислення, а також виробити розуміння соціальних, правових та етичних аспектів використання ІТ.

Основними завданнями, що мають бути вирішені у процесі викладання дисципліни, є:

- навчити розуміти основні поняття та термінологію інформатики та інформаційних технологій.
- навчити застосовувати базові принципи алгоритмізації та структурування даних.
- надати практичні навички роботи з поширеним прикладним програмним забезпеченням та сервісами.
- ознайомити з основами функціонування комп'ютерних систем та мережевих технологій.
- сформувати розуміння принципів вибору та ефективного використання інформаційних технологій у професійній діяльності.
- навчити аналізувати завдання з точки зору можливості їх автоматизації та вибору відповідних технологічних рішень.
- ознайомити з питаннями інформаційної безпеки та надійного використання технологій.

ЗМІСТОВИЙ МОДУЛЬ 1

ОСНОВНІ ПОНЯТТЯ ПРО ІНФОРМАТИКУ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ, MICROSOFT OFFICE

ТЕМА 1.1. ВСТУП ДО ІНФОРМАТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Мета: Ознайомити здобувачів вищої освіти з основними поняттями інформатики та ІТ, їхнім предметом вивчення, історією та роллю в сучасному суспільстві.

План:

1. Визначення інформатики.
2. Визначення інформаційних технологій (ІТ).
3. Поняття інформації, даних та інформаційних процесів.
4. Історія розвитку ІТ.
5. Роль ІТ у сучасному суспільстві.

1. Визначення інформатики

Інформатика – це фундаментальна технічна наука, що системно досліджує структуру, загальні властивості та закономірності створення, зберігання, пошуку, перетворення, передавання та використання інформації за допомогою комп'ютерної техніки. Вона вивчає не лише самі процеси обробки даних, але й принципи функціонування засобів обчислювальної техніки, які ці процеси реалізують.

Предмет інформатики охоплює:

- Апаратне забезпечення (Hardware) – фізичні компоненти комп'ютерних систем.
- Програмне забезпечення (Software) – набір програм та інструкцій, що керують роботою апаратного забезпечення.
- Алгоритмічні основи – методи та підходи до розв'язання задач за допомогою обчислювальних пристроїв.
- Взаємодію людини з комп'ютером (інтерфейс).

Інформатика тісно пов'язана з іншими науковими дисциплінами, такими як теорія інформації, математика, логіка, кібернетика та штучний інтелект. Вона слугує теоретичною базою для розвитку та застосування інформаційних технологій у різних сферах людської діяльності. Окремим напрямком є економічна інформатика, що фокусується на інформаційних системах для підтримки прийняття рішень в управлінні, економіці та бізнесі.

2. Визначення інформаційних технологій (ІТ)

Інформаційні технології (ІТ) – це сукупність методів, процесів та засобів, що використовуються для здійснення інформаційних процесів: збору, накопичення,

зберігання, обробки, подання, передавання та використання інформації. По суті, ІТ є практичним застосуванням досягнень інформатики та комп'ютерної інженерії для вирішення конкретних завдань.

ІТ можна розглядати як цілеспрямовану людино-машинну технологію роботи з інформацією, що базується на використанні обчислювальної техніки та програмного забезпечення. Часто використовується також термін "інформаційно-комунікаційні технології" (ІКТ), який підкреслює важливу роль сучасних засобів комунікації, зокрема комп'ютерних мереж, у передаванні даних.

Фахівців у сфері ІТ умовно можна поділити на тих, хто обслуговує обладнання, розробляє програмне забезпечення та працює з готовими інформаційними продуктами.

3. Поняття інформації, даних та інформаційних процесів

- **Інформація:** Це відомості про об'єкти, явища та процеси навколишнього світу, незалежно від форми їх подання. У контексті ІТ, інформація – це дані, що несуть певний зміст для їх отримувача. Це фундаментальне поняття, яке не має єдиного строгого визначення.
- **Дані:** Це інформація, подана у формалізованому вигляді, придатному для зберігання, обробки та передавання за допомогою технічних засобів, зокрема комп'ютерів. Дані можуть бути представлені у вигляді сигналів (знаків), які фіксуються на носіях.
- **Інформаційні процеси:** Це дії, що виконуються з інформацією. До основних інформаційних процесів належать:
 - **Пошук:** Знаходження потрібної інформації у певних джерелах.
 - **Збирання:** Накопичення інформації з різних джерел.
 - **Зберігання:** Забезпечення можливості доступу до інформації протягом певного часу (наприклад, на дисках, флеш-накопичувачах).
 - **Опрацювання (обробка):** Перетворення інформації з одного виду в інший за певними правилами (наприклад, обчислення, сортування, кодування).
 - **Передавання:** Переміщення інформації від джерела до приймача за допомогою каналів зв'язку (наприклад, через комп'ютерні мережі).
 - **Використання:** Застосування інформації для прийняття рішень або виконання певних дій.
 - **Захист:** Забезпечення конфіденційності, цілісності та доступності інформації.

Інформаційні системи – це організаційно-технічні системи, що реалізують інформаційні процеси за допомогою комп'ютерної техніки.

4. Історія розвитку ІТ

Розвиток інформаційних технологій нерозривно пов'язаний з еволюцією обчислювальної техніки. Ключовими етапами були:

- Механічні обчислювальні пристрої (арифмометри, аналітична машина Беббіджа).
- Поява електронних обчислювальних машин (ЕОМ) у середині ХХ століття (наприклад, ENIAC). Це стало початком автоматизованої обробки інформації.
- Винахід транзисторів та інтегральних схем, що призвело до зменшення розмірів, вартості та підвищення надійності ЕОМ.
- Розробка перших мов програмування високого рівня.
- Поява персональних комп'ютерів (ПК) у 1970-80-х роках, що зробило обчислювальну техніку доступною для широкого кола користувачів.
- Розвиток комп'ютерних мереж та Інтернету, що революціонізувало процеси комунікації та доступу до інформації.
- Сучасний етап характеризується розвитком мобільних технологій, хмарних обчислень, штучного інтелекту, великих даних та Інтернету речей.

5. Роль ІТ у сучасному суспільстві

Інформаційні технології стали невід'ємною складовою сучасного суспільства та професійної діяльності. Інформатизація є однією з основ розвитку економіки та суспільства в цілому.

ІТ використовуються практично в усіх сферах:

- **Освіта:** Дистанційне навчання, електронні підручники, освітні платформи, комп'ютерне тестування.
- **Наука:** Моделювання процесів, аналіз великих обсягів даних, наукові розрахунки, доступ до наукових публікацій.
- **Бізнес та управління:** Автоматизація процесів, електронна комерція, аналіз даних для прийняття рішень, управління ресурсами.
- **Промисловість:** Автоматизація виробництва, проектування (САПР), управління логістикою.
- **Медицина:** Електронні медичні картки, діагностика, телемедицина, аналіз медичних даних.
- **Культура та мистецтво:** Цифрові архіви, віртуальні музеї, створення цифрового контенту.
- **Юриспруденція:** Правові бази даних, електронний документообіг, інформаційні ресурси для юридичної діяльності.
- **Повсякденне життя:** Спілкування (соціальні мережі, месенджери), пошук інформації, онлайн-покупки, розваги.

Володіння інформаційно-комунікаційними технологіями є однією з ключових компетентностей сучасної людини. Метою вивчення дисципліни "Інформатика та ІТ" є формування навичок роботи з комп'ютером для вирішення професійних та освітніх завдань.

ТЕМА 1.2. КОДУВАННЯ ТА ВИМІРЮВАННЯ ІНФОРМАЦІЇ

Мета: Пояснити принципи двійкового кодування, одиниці вимірювання інформації та способи кодування текстових даних (ASCII, Unicode).

План:

1. Двійкове кодування інформації.
2. Одиниці вимірювання інформації: біт, байт та похідні.
3. Кодування текстової інформації.
4. Таблиці кодування: ASCII, Windows-1251, KOI8-U.
5. Стандарт Unicode.

1. Двійкове кодування інформації

В основі роботи сучасної цифрової обчислювальної техніки лежить **двійкове кодування** – представлення будь-якої інформації за допомогою лише двох символів (сигналів). Зазвичай ці символи позначають цифрами **0** та **1**. Такий вибір зумовлений простотою технічної реалізації двох станів: наявність/відсутність електричного сигналу, намагнічено/не намагнічено, увімкнено/вимкнено тощо.

Кожна цифра (0 або 1) у двійковому коді називається **біт** (англ. *binary digit* – двійкова цифра). Біт є найменшою одиницею вимірювання кількості інформації. Один біт дозволяє закодувати лише два різних стани або повідомлення (наприклад, "так"/"ні", "чорний"/"білий").

Для кодування більшої кількості станів використовують послідовності бітів. За допомогою n бітів можна закодувати 2^n різних повідомлень або станів. Наприклад:

- 2 біти: $2^2=4$ коди (00, 01, 10, 11).
- 3 біти: $2^3=8$ кодів (000, 001, ..., 111).
- 8 бітів: $2^8=256$ кодів.

Таким чином, будь-яка інформація в комп'ютері – текст, зображення, звук, відео – зрештою представляється у вигляді послідовностей нулів та одиниць.

2. Одиниці вимірювання інформації: біт, байт та похідні

Як зазначалося, **біт** є найменшою одиницею інформації. Однак для вимірювання обсягів даних, що зберігаються та обробляються комп'ютером, біт є надто дрібною одиницею.

Основною одиницею вимірювання обсягу даних є байт (англ. *byte*). Байт – це послідовність з 8 бітів.

1 Байт=8 біт

Саме 8 біт (1 байт) зазвичай достатньо для кодування одного символу тексту (літери, цифри, знака пунктуації) у поширених таблицях кодування. Тому обсяг текстових даних часто вимірюють у байтах, де кожен символ займає 1 байт.

Для вимірювання великих обсягів даних використовують похідні одиниці:

- **Кілобайт (КБ, KB)**
- **Мегабайт (МБ, MB)**
- **Гігабайт (ГБ, GB)**
- **Терабайт (ТБ, TB)**
- **Петабайт (ПБ, PB)** та інші.

Історично склалося, що в інформатиці префікси кіло-, мега-, гіга- тощо часто означають степені двійки, а не десятки:

1 КБ=1024 Байт=2¹⁰ Байт

1 МБ=1024 КБ=2²⁰ Байт

1 ГБ=1024 МБ=2³⁰ Байт

1 ТБ=1024 ГБ=2⁴⁰ Байт

Однак, варто зазначити, що існують і стандарти, де ці префікси означають степені десяти (як у системі СІ: 1 кБ = 1000 Байт). Для уникнення плутанини були введені двійкові префікси (Кібі-, Мебі-, Гібі- тощо), де 1 КіБ (кібібайт) = 1024 Байт. Проте на практиці, особливо при позначенні ємності пам'яті, часто використовують традиційне розуміння КБ, МБ, ГБ як степенів 1024. Виробники жорстких дисків, навпаки, часто вказують ємність, виходячи зі степенів 1000, що призводить до розбіжностей у відображуваному обсязі.

Таблиця 1. Одиниці вимірювання інформації

Одиниця (Двійкова)	Символ (IEC)	Співвідношення (Степінь 2)	Приблиз на кількість Байт	Одиниця (Десятко ва)	Символ (СІ)	Співвідношення (Степінь 10)
Біт	біт (b)	-	1/8	Біт	біт (b)	-
Байт	Б (B)	8 біт	1	Байт	Б (B)	8 біт
Кібібайт	КиБ (KiB)	1024 Б (2 ¹⁰ Б)	1 024	Кілобайт	кБ (kB)	1000 Б (10 ³ Б)

Мебібайт	МіБ (MiB)	1024 KiB (220 Б)	1 048 576	Мегабайт	МБ (MB)	1000 кБ (106 Б)
Гібібайт	ГіБ (GiB)	1024 MiB (230 Б)	1 073 741 824	Гігабайт	ГБ (GB)	1000 МБ (109 Б)
Пебібайт	ПіБ (PiB)	1024 TiB (250 Б)	1 125 899 906 842 624	Петабайт	ПБ (PB)	1000 ТБ (1015 Б)

3. Кодування текстової інформації

Для представлення текстової інформації в комп'ютері кожному символу (літері, цифрі, знаку пунктуації, пробілу, спеціальному символу) ставиться у відповідність унікальний двійковий код. Ця відповідність визначається **таблицею кодування**.

Необхідність стандартизації таблиць кодування виникла для того, щоб різні комп'ютери та програми могли однаково інтерпретувати текстові дані.

4. Таблиці кодування: ASCII, Windows-1251, KOI8-U

- **ASCII (American Standard Code for Information Interchange):** Історично один із перших і найпоширеніших стандартів.
 - Базова версія використовує 7 біт для кодування 128 символів (коди від 0 до 127), що включають цифри, латинські літери (великі та малі), знаки пунктуації та керуючі символи.
 - Розширена версія ASCII використовує 8 біт (1 байт) для кодування 256 символів (коди від 0 до 255). Перші 128 кодів збігаються з базовим ASCII, а друга половина (коди 128-255) використовується для символів національних алфавітів, символів псевдографіки тощо.
 - Існує багато варіантів розширеного ASCII для різних мов.
- **Windows-1251:** 8-бітне кодування, розроблене Microsoft для підтримки кирилических алфавітів (включаючи український) в ОС Windows.
- **KOI8-U:** Інше 8-бітне кодування, популярне в Unix-подібних системах та Інтернеті для української мови.

Проблема 8-бітних кодувань: Різні 8-бітні кодування використовують однакові коди (128-255) для позначення різних символів. Це призводить до проблем

сумісності при передачі або спільному використанні текстових файлів, створених у різних кодуваннях (текст відображається неправильно, так звані "кракозябри").

Таблиця 2. Приклади кодів ASCII (фрагмент)

Символ	Десятковий код	Двійковий код (8 біт)
A	65	01000001
B	66	01000010
a	97	01100001
b	98	01100010
0	48	00110000
1	49	00110001
Пробіл	32	00100000
{	123	01111011

5. Стандарт Unicode

Для вирішення проблеми сумісності та забезпечення можливості представлення символів практично всіх писемностей світу був розроблений стандарт **Unicode**.

- **Мета:** Присвоїти унікальний код кожному символу, незалежно від платформи, програми чи мови.
- **Обсяг:** Unicode може представляти величезну кількість символів (понад мільйон).
- **Форми представлення:**
 - **UTF-8 (Unicode Transformation Format - 8-bit):** Найпоширеніший

формат в Інтернеті. Використовує змінну довжину коду: 1 байт для символів ASCII (коди 0-127), 2-4 байти для інших символів. Забезпечує сумісність з ASCII.

- **UTF-16:** Використовує 2 байти для більшості поширених символів і 4 байти для рідкісних. Основний формат у Windows та Java.
- **UTF-32:** Використовує фіксовану довжину 4 байти для кожного символу.

Перехід на Unicode дозволив уніфікувати представлення текстових даних, значно спростивши обмін інформацією між різними системами та підтримку багатомовності. У кодуванні Unicode один символ може займати більше одного байта (наприклад, 2 байти в UTF-16 або від 1 до 4 байтів в UTF-8).

ТЕМА 1.3. КОДУВАННЯ ГРАФІЧНОЇ ТА ЗВУКОВОЇ ІНФОРМАЦІЇ

Мета: Пояснити принципи кодування графічних (растрових, векторних) та звукових даних, основні формати файлів та колірні моделі.

План:

1. Кодування графічної інформації.
 - 1.1. Растрова графіка: принцип, параметри, переваги/недоліки, формати.
 - 2.1. Векторна графіка: принцип, переваги/недоліки, сфери застосування, формати.
2. Колірні моделі (RGB, CMYK, HSB/HSV).
3. Кодування звукової інформації: дискретизація, квантування, кодування.
4. Формати аудіофайлів.

1. Кодування графічної інформації

Подібно до тексту, графічна інформація для комп'ютерної обробки також представляється у двійковому вигляді. Існують два основні підходи до кодування графіки: растровий та векторний.

1.1. Растрова графіка

- **Принцип:** Зображення представляється як сітка (растр) з окремих точок – **пікселів** (англ. *picture element*). Кожен піксель має певний колір. Якість зображення залежить від розміру растра (кількості пікселів по ширині та висоті) та кількості кольорів, які може приймати кожен піксель.
- **Процес створення:**
 - **Дискретизація:** Просторова дискретизація – розбиття неперервного зображення на пікселі.

- **Квантування кольору:** Присвоєння кожному пікселю певного колірної значення з обмеженої палітри.
- **Кодування:** Запис кодів кольорів усіх пікселів у файл.
- **Параметри:**
 - **Розмір зображення (роздільна здатність):** Кількість пікселів по ширині та висоті (наприклад, 1920x1080).
 - **Глибина кольору:** Кількість бітів, що використовуються для кодування кольору одного пікселя. Визначає максимальну кількість кольорів у палітрі.
 - 1 біт: 2 кольори (чорно-біле).
 - 8 біт: 256 кольорів (індексована палітра).
 - 16 біт (High Color): 65 536 кольорів.
 - 24 біти (True Color): $2^{24} \approx 16.7$ мільйонів кольорів (по 8 біт на R, G, B компоненти).
 - 32 біти: True Color + 8 біт на альфа-канал (прозорість).
- **Переваги:**
 - Фотореалістичність: добре відтворює складні зображення з великою кількістю деталей, тонів та напівтіней.
- **Недоліки:**
 - Великий розмір файлів, особливо для зображень з високою роздільною здатністю та глибиною кольору.
 - Втрата якості при масштабуванні (збільшенні) – з'являється "пікселізація".
 - Складність редагування окремих об'єктів на зображенні.
- **Формати файлів:**
 - **BMP (Bitmap):** Зберігає інформацію без стиснення або з простим стисненням без втрат. Великий розмір файлу.
 - **JPEG/JPG (Joint Photographic Experts Group):** Використовує стиснення з **втратами** якості, що дозволяє значно зменшити розмір файлу. Ідеально підходить для фотографій. Ступінь стиснення можна регулювати.
 - **GIF (Graphics Interchange Format):** Використовує стиснення **без втрат** якості. Підтримує обмежену палітру (до 256 кольорів), прозорість та анімацію. Добре підходить для простих зображень, логотипів, іконок.
 - **PNG (Portable Network Graphics):** Використовує стиснення **без втрат** якості. Підтримує велику глибину кольору та прозорість (альфа-канал). Є сучасною альтернативою GIF (без анімації) та частково JPEG (коли важлива відсутність втрат). Кращий за SVG для растрової прозорості.
 - **TIFF (Tagged Image File Format):** Універсальний формат, підтримує зберігання без втрат та різні колірні моделі. Часто використовується у поліграфії.

1.2. Векторна графіка

- **Принцип:** Зображення описується за допомогою математичних формул, які визначають геометричні об'єкти (примітиви): лінії, криві (наприклад, криві Безьє), прямокутники, еліпси, текст тощо. Зберігається не зображення піксель за пікселем, а інструкції для його побудови.
- **Переваги:**
 - **Масштабованість:** Зображення можна збільшувати або зменшувати до будь-яких розмірів без втрати якості, оскільки воно перераховується за формулами.
 - **Малий розмір файлів:** Файли зазвичай значно менші за растрові аналоги, оскільки зберігаються лише параметри об'єктів, а не кожен піксель.
 - **Легкість редагування:** Окремі об'єкти (лінії, фігури) можна легко модифікувати (переміщувати, змінювати розмір, колір, форму) незалежно від інших.
- **Недоліки:**
 - **Складність відтворення фотореалістичних зображень:** Важко точно передати плавні переходи кольорів, текстури, складні деталі фотографій. Зображення можуть виглядати "неприродно" або стилізовано.
- **Сфери застосування:** Створення логотипів, іконок, ілюстрацій, технічних креслень, діаграм, карт, шрифтів, веб-графіки.
- **Формати файлів:**
 - **SVG (Scalable Vector Graphics):** Відкритий стандарт на основі XML для векторної графіки в Інтернеті. Підтримує інтерактивність та анімацію.³⁵ Завдяки текстовому представленню (XML) індексується пошуковими системами.
 - **AI (Adobe Illustrator):** Власний формат програми Adobe Illustrator.
 - **CDR (CorelDRAW):** Власний формат програми CorelDRAW.
 - **WMF/EMF (Windows Metafile / Enhanced Metafile):** Формати для обміну векторними даними між програмами в ОС Windows.

Вибір між растровою та векторною графікою залежить від конкретного завдання. Для фотографій краще підходить растрова графіка (JPEG), для логотипів та схем – векторна (SVG, AI).

Таблиця 3. Порівняння растрової та векторної графіки

Характеристика	Растрова графіка	Векторна графіка
Принцип представлення	Сітка пікселів	Математичний опис геометричних об'єктів (лінії, криві, фігури)
Основний елемент	Піксель	Геометричний примітив (точка, лінія, крива)
Масштабування	Втрата якості при збільшенні (пікселізація)	Без втрати якості
Розмір файлу	Зазвичай великий (залежить від роздільної здатності)	Зазвичай малий (залежить від складності об'єктів)
Реалістичність	Висока (фотографії, складні зображення)	Обмежена (краще для ілюстрацій, схем)
Редагування	Редагування окремих пікселів	Редагування окремих об'єктів (форм, ліній)
Сфери застосування	Фотографії, скановані зображення, веб-графіка	Логотипи, ілюстрації, креслення, шрифти, карти, діаграми, веб-графіка
Основні формати	BMP, JPEG, GIF, PNG, TIFF	SVG, AI, CDR, WMF, EMF

2. Колірні моделі (RGB, CMYK, HSB/HSV)

Колірна модель – це спосіб опису кольорів за допомогою числових значень (координат) у певній системі.

- **RGB (Red, Green, Blue – Червоний, Зелений, Синій):**

- **Адитивна модель:** Кольори утворюються шляхом додавання (змішування) світла трьох основних кольорів.
- **Застосування:** Пристрої, що випромінюють світло (монітори, проектори, сканери, цифрові камери).
- **Принцип:** Кожен колір описується трьома значеннями інтенсивності червоної, зеленої та синьої складових (зазвичай від 0 до 255 для 24-бітного кольору).
 - Чорний: R=0, G=0, B=0 (відсутність світла).
 - Білий: R=255, G=255, B=255 (максимальна інтенсивність усіх компонентів).
 - Сірий: R=G=B.

- **CMYK (Cyan, Magenta, Yellow, Key/Black – Блакитний, Пурпурний, Жовтий, Чорний):**

- **Субтрактивна модель:** Кольори утворюються шляхом віднімання певних кольорів з білого світла при його відбитті від поверхні, покритої фарбами.
- **Застосування:** Поліграфія, друк.
- **Принцип:** Кожен колір описується чотирма значеннями, що вказують на кількість блакитної, пурпурної, жовтої та чорної фарб (зазвичай у відсотках від 0 до 100).
 - Білий: C=0, M=0, Y=0, K=0 (колір паперу).
 - Чорний: Теоретично C=100, M=100, Y=100, але на практиці використовують K=100 (Key color) для отримання глибокого чорного та економії фарб.
- Зв'язок з RGB: Cyan = White - Red; Magenta = White - Green; Yellow = White - Blue.

- **HSB/HSV (Hue, Saturation, Brightness/Value – Тон, Насиченість, Яскравість/Значення):**

- **Модель, орієнтована на сприйняття:** Описує колір так, як його сприймає людина.
- **Компоненти:**
 - **Hue (Тон):** Сам колір (наприклад, червоний, зелений, синій). Вимірюється в градусах на колірному колі (0-360°).
 - **Saturation (Насиченість):** Інтенсивність або чистота кольору (0% – відтінок сірого, 100% – чистий колір).
 - **Brightness/Value (Яскравість/Значення):** Світлота кольору (0% – чорний, 100% – найяскравіший колір).

- **Застосування:** Зручна для вибору кольору в графічних редакторах.

3. Кодування звукової інформації

Звук у природі є **аналоговим сигналом** – неперервною хвилею коливань тиску повітря. Щоб комп'ютер міг обробляти звук, аналоговий сигнал потрібно перетворити на цифровий (послідовність двійкових кодів). Цей процес називається аналого-цифровим перетворенням (АЦП) і включає два основні етапи: дискретизацію та квантування

- **Дискретизація (за часом):**

- Вимірювання амплітуди (гучності) звукової хвилі через рівні, дуже короткі проміжки часу.
- **Частота дискретизації (Sampling Rate):** Кількість вимірювань (відліків) за одну секунду. Вимірюється в Герцах (Гц) або кілогерцах (кГц).
- Чим вища частота дискретизації, тим точніше відтворюється форма вихідної хвилі, але тим більший обсяг даних.
- **Теорема Котельникова-Найквіста-Шеннона:** Для точного відновлення сигналу частота дискретизації має бути щонайменше вдвічі вищою за максимальну частоту в спектрі сигналу. Оскільки людина чує звуки приблизно до 20 кГц, для якісного запису використовують частоти 44.1 кГц (Audio CD), 48 кГц (відео) або вище.

- **Квантування (за рівнем):**

- Представлення кожного вимірюваного значення амплітуди найближчим дискретним рівнем з обмеженого набору можливих значень.
- **Розрядність квантування (Bit Depth):** Кількість бітів, що використовується для запису значення кожного відліку. Визначає кількість можливих рівнів гучності.
- Чим більша розрядність, тим точніше передається динамічний діапазон звуку (різниця між найтихішим і найгучнішим звуком), менший шум квантування, але більший обсяг даних.
- Приклади: 8 біт ($2^8=256$ рівнів), 16 біт ($2^{16}=65536$ рівнів – стандарт Audio CD), 24 біти.

- **Кодування:**

- Кожному квантованому значенню (рівню) присвоюється унікальний двійковий код. Послідовність цих кодів і є цифровим представленням звуку.

Обсяг нестисненого аудіофайлу можна розрахувати за формулою:

$V = f \times d \times t \times k$, де

V – обсяг (у бітах),

f – частота дискретизації (Гц),

d – розрядність квантування (біт),

t – тривалість звучання (с),

k – кількість каналів (1 – моно, 2 – стерео).

4. Формати аудіофайлів

Існує багато форматів для зберігання цифрового аудіо, які відрізняються методами кодування та стиснення:

- **Без стиснення:**
 - **WAV (Waveform Audio File Format):** Стандартний формат для Windows, зазвичай зберігає нестиснені дані (PCM - Pulse Code Modulation). Висока якість, великий розмір файлу.
 - **AIFF (Audio Interchange File Format):** Аналогічний WAV, стандартний для macOS.
- **Зі стисненням без втрат (Lossless):**
 - **FLAC (Free Lossless Audio Codec):** Популярний відкритий формат, зменшує розмір файлу без втрати якості звуку.
 - **ALAC (Apple Lossless Audio Codec):** Формат Apple.
- **Зі стисненням із втратами (Lossy):**
 - **MP3 (MPEG-1 Audio Layer III):** Найпоширеніший формат. Значно зменшує розмір файлу, видаляючи інформацію, яку людське вухо сприймає погано. Якість залежить від бітрейту (кількості біт на секунду).
 - **AAC (Advanced Audio Coding):** Більш сучасний та ефективний формат, ніж MP3, при тому ж бітрейті. Використовується Apple, YouTube.
 - **WMA (Windows Media Audio):** Формат Microsoft.
 - **Ogg Vorbis:** Відкритий формат, порівнянний з AAC.

Вибір формату залежить від потреб: для архівування та професійної роботи краще використовувати формати без втрат, для портативних пристроїв та передачі через Інтернет – формати зі стисненням із втратами.

Таблиця 4. Основні колірні моделі

Модель	Принцип	Основні компоненти	Основне застосування
RGB	Адитивна (додавання світла)	Red (Червоний), Green (Зелений), Blue (Синій)	Монітори, екрани, сканери, камери

CMYK	Субтрактивна (віднімання світла фарбами)	Сяан (Блакитний), Magenta (Пурпурний), Yellow (Жовтий), Key/Black (Чорний)	Друк, поліграфія
HSB/HSV	Орієнтована на сприйняття людиною	Hue (Тон), Saturation (Насиченість), Brightness/Value (Яскравість/Знач ення)	Вибір кольору в редакторах

ТЕМА 1.4. АРХІТЕКТУРА КОМП'ЮТЕРА

Мета: Ознайомити з класичною архітектурою фон Неймана, основними компонентами персонального комп'ютера (ПК) та їх функціями.

План:

1. Архітектура фон Неймана: принципи та компоненти.
2. Апаратне забезпечення (Hardware) ПК.
 - 2.1. Системний блок та його компоненти (материнська плата, ЦП, ОЗП, накопичувачі, відеокарта, блок живлення).
 - 2.2. Периферійні пристрої (введення та виведення).
3. Взаємодія компонентів комп'ютера.

1. Архітектура фон Неймана: принципи та компоненти

Більшість сучасних комп'ютерів, від персональних до великих серверів, побудовані на принципах, сформульованих видатним математиком Джоном фон Нейманом у середині 1940-х років (близько 1945-1946 рр.). Цей набір принципів отримав назву **архітектури фон Неймана** (іноді її також називають Принстонською архітектурою, оскільки вона описувала комп'ютер ENIAC, створений в Принстонському університеті).

Основні принципи архітектури фон Неймана:

1. **Принцип збереженої програми:** Команди (програма) та дані зберігаються в одній і тій самій пам'яті комп'ютера. Це дозволяє розглядати програми як дані та модифікувати їх під час виконання (хоча це рідко робиться в сучасних системах з міркувань безпеки).

2. **Принцип двійкового кодування:** Вся інформація, включаючи команди та дані, представляється у вигляді двійкових кодів (послідовностей 0 та 1).

3. **Принцип адресності пам'яті:** Основна пам'ять складається з окремих комірок, кожна з яких має унікальну адресу. Це дозволяє процесору звертатися до будь-якої комірки пам'яті для читання або запису даних/команд.

4. **Принцип послідовного виконання команд:** Процесор вибирає команди з пам'яті послідовно, одну за одною (відповідно до порядку їх розташування або спеціальних команд переходу), та виконує їх. Послідовність виконання є фундаментальною рисою цієї архітектури.

5. **Наявність основних логічних блоків:** Комп'ютер складається з взаємопов'язаних функціональних пристроїв.

Основні компоненти (блоки) архітектури фон Неймана:

- **Арифметико-логічний пристрій (АЛП / ALU - Arithmetic Logic Unit):** Виконує всі арифметичні (додавання, віднімання, множення, ділення) та логічні (AND, OR, NOT, порівняння) операції над даними.
- **Пристрій керування (ПК / CU - Control Unit):** Керує роботою всіх інших пристроїв комп'ютера. Він зчитує команди з пам'яті, розшифровує їх та генерує сигнали керування для АЛП, пам'яті та пристроїв введення/виведення, забезпечуючи тим самим виконання програми.
- **Пам'ять (Запам'ятовуючий пристрій / Memory):** Призначена для зберігання програм (послідовностей команд) та даних, над якими ці програми працюють. В архітектурі фон Неймана мається на увазі основна (оперативна) пам'ять.
- **Пристрої введення/виведення (ПВВ / I/O Devices):** Забезпечують зв'язок комп'ютера із зовнішнім світом: введення початкових даних та програм, виведення результатів роботи.

АЛП та пристрій керування часто об'єднують в один блок, який називається **Центральним процесором (ЦП / CPU - Central Processing Unit)**.

Існує також **Гарвардська архітектура**, де використовується роздільна пам'ять для команд та даних, що дозволяє одночасно вибирати команду та оперувати даними, потенційно підвищуючи швидкодію. Однак класична архітектура фон Неймана залишається основою для більшості універсальних комп'ютерів.

2. Апаратне забезпечення (Hardware) ПК

Апаратне забезпечення (Hardware) – це сукупність усіх фізичних (електронних та механічних) компонентів комп'ютера та периферійних пристроїв. Типовий персональний комп'ютер складається з системного блоку та підключених до нього периферійних пристроїв.

2.1. Системний блок

Це основний блок, в якому розміщені ключові компоненти комп'ютера.

- **Материнська (системна) плата (Motherboard):** Велика друкована плата, яка є основою комп'ютера. На ній розміщуються процесор, оперативна пам'ять, слоти для карт розширення (відеокарта, звукова карта тощо), чіпсет (набір мікросхем, що керують взаємодією компонентів), роз'єми для підключення накопичувачів та периферійних пристроїв. Вона забезпечує електричне та логічне з'єднання всіх компонентів системи.
- **Центральний процесор (ЦП / CPU):** "Мозок" комп'ютера, що виконує обчислення, обробку даних та керування роботою всіх пристроїв відповідно до програмних інструкцій.
 - Основні характеристики:
 - *Тактова частота:* Кількість операцій (тактів), що виконуються за секунду (вимірюється в ГГц).
 - *Кількість ядер:* Сучасні процесори часто мають кілька ядер, що дозволяє паралельно виконувати кілька задач.
 - *Кеш-пам'ять:* Надшвидка пам'ять малого обсягу, вбудована в процесор для зберігання даних, що часто використовуються.
- **Оперативна пам'ять (ОЗП / RAM - Random Access Memory):** Швидка, енергозалежна пам'ять, призначена для тимчасового зберігання програм та даних, з якими процесор працює в поточний момент. При вимкненні живлення дані в ОЗП втрачаються. Основні характеристики: обсяг (ГБ), тип (DDR3, DDR4, DDR5), частота.
- **Постійна пам'ять (Накопичувачі):** Пристрої для довготривалого зберігання операційної системи, програм та даних користувача.
 - *Жорсткий диск (HDD - Hard Disk Drive):* Традиційний накопичувач на основі магнітних пластин, що обертаються. Має велику ємність, але повільніший за SSD.
 - *Твердотільний накопичувач (SSD - Solid State Drive):* Накопичувач на основі мікросхем флеш-пам'яті. Значно швидший за HDD, тихіший, стійкіший до ударів, але зазвичай дорожчий при однаковій ємності.
 - *Оптичні приводи (CD/DVD/Blu-ray):* Для читання/запису оптичних дисків (використовуються все рідше).
- **Відеокарта (Графічний адаптер / Graphics Card):** Пристрій, що обробляє графічну інформацію та виводить зображення на монітор. Може бути інтегрованою (вбудованою в процесор або материнську плату) або дискретною

(окремою платою, що встановлюється в слот). Дискретні відеокарти зазвичай значно потужніші.

- **Звукова карта (Аудіоадаптер):** Обробляє звук та забезпечує його введення (з мікрофона) та виведення (на колонки/навушники). Може бути інтегрованою або дискретною.
- **Мережева карта (Мережевий адаптер / NIC - Network Interface Card):** Дозволяє комп'ютеру підключатися до локальної мережі або Інтернету (дротове з'єднання Ethernet або бездротове Wi-Fi).
- **Блок живлення (Power Supply Unit - PSU):** Перетворює змінний струм з електромережі на постійний струм низької напруги, необхідний для живлення всіх компонентів системного блоку.

2.2. Периферійні пристрої

Пристрої, що підключаються до системного блоку для взаємодії з користувачем та обміну даними із зовнішнім світом.

- **Пристрої введення:**
 - **Клавіатура:** Основний пристрій для введення текстової інформації та команд.
 - **Миша:** Маніпулятор для керування курсором та взаємодії з графічним інтерфейсом.
 - **Сканер:** Для введення графічних зображень та тексту з паперових носіїв.
 - **Мікрофон:** Для введення звукової інформації.
 - **Веб-камера:** Для введення відеозображення.
 - Інші: джойстики, графічні планшети.
- **Пристрої виведення:**
 - **Монітор (Дисплей):** Основний пристрій для візуального відображення текстової та графічної інформації.
 - **Принтер:** Для виведення інформації на папір (друк).
 - **Акустична система (Колонки) / Навушники:** Для виведення звукової інформації.
 - **Проектор:** Для виведення зображення на великий екран.

3. Взаємодія компонентів комп'ютера

Робота комп'ютера – це злагоджена взаємодія його апаратних та програмних компонентів. Узагальнена схема обробки даних виглядає так:

1. **Введення:** Дані та команди вводяться користувачем за допомогою пристроїв введення (наприклад, клавіатури).
2. **Зберігання в ОЗП:** Введені дані та програми завантажуються в оперативну пам'ять для швидкого доступу процесором.
3. **Обробка:** Процесор (під керівництвом Пристрою Керування) зчитує команди та дані з ОЗП. Арифметико-логічний пристрій виконує необхідні операції. Проміжні та кінцеві результати записуються назад в ОЗП.

4. **Виведення:** Результати обробки передаються з ОЗП на пристрої виведення (наприклад, монітор або принтер) для представлення користувачеві.

Пристрій керування координує всі ці процеси, забезпечуючи передачу даних між пам'яттю, процесором та пристроями введення/виведення.

Таблиця 5. Основні компоненти ПК та їх функції

Компонент	Основна функція
Центральний процесор (ЦП)	Виконання обчислень, обробка даних, керування роботою комп'ютера
Оперативна пам'ять (ОЗП)	Тимчасове зберігання програм та даних, що активно використовуються процесором
Накопичувач (HDD/SSD)	Довготривале зберігання операційної системи, програм та файлів користувача
Відеокарта	Обробка графічної інформації та виведення зображення на монітор
Материнська плата	З'єднання та забезпечення взаємодії всіх основних компонентів системного блоку
Пристрої введення	Введення даних та команд у комп'ютер (клавіатура, миша, сканер, мікрофон)

Пристрої виведення	Представлення результатів роботи комп'ютера користувачеві (монітор, принтер, колонки)
Блок живлення	Забезпечення електроенергією компонентів системного блоку

ТЕМА 1.5. СИСТЕМНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

Мета: Дати класифікацію програмного забезпечення (ПЗ), детально розглянути системне ПЗ, зокрема операційні системи (ОС), їх функції, класифікацію та приклади, а також роль драйверів та утиліт.

План:

1. Класифікація програмного забезпечення (ПЗ).
2. Операційні системи (ОС): призначення та функції.
3. Класифікація та приклади ОС.
4. Драйвери пристроїв.
5. Утиліти (сервісні програми).

1. Класифікація програмного забезпечення (ПЗ)

Програмне забезпечення (ПЗ / Software) – це сукупність програм, що керують роботою комп'ютера та дозволяють вирішувати за його допомогою різноманітні завдання. На відміну від апаратного забезпечення (Hardware), ПЗ є нематеріальним компонентом комп'ютерної системи.

Все програмне забезпечення прийнято поділяти на три основні категорії:

1. **Системне ПЗ (System Software):** Комплекс програм, що забезпечують керування компонентами комп'ютерної системи (апаратним забезпеченням) та створюють середовище для виконання прикладних програм. Це базовий рівень ПЗ, без якого комп'ютер не може функціонувати.
2. **Прикладне ПЗ (Application Software):** Програми, призначені для виконання конкретних завдань користувача. Це можуть бути офісні програми, графічні редактори, ігри, бухгалтерські системи тощо.
3. **Інструментальне ПЗ (Programming Software / Utilities):** Програми, що використовуються для розробки, тестування та налагодження іншого програмного забезпечення. До них належать компілятори, інтерпретатори, відладчики, середовища розробки (IDE).

2. Операційні системи (ОС): призначення та функції

Операційна система (ОС) – це основна та найважливіша частина системного програмного забезпечення. Це комплекс взаємопов'язаних програм, який керує апаратними ресурсами комп'ютера, організовує роботу з файлами, забезпечує взаємодію (інтерфейс) між користувачем та комп'ютером, а також створює середовище для виконання прикладних програм.

Основні функції операційної системи:

- **Керування ресурсами:** ОС розподіляє та контролює використання всіх апаратних ресурсів комп'ютера (процесорний час, оперативна пам'ять, дисковий простір, периферійні пристрої) між різними програмами та користувачами.
- **Керування процесами:** ОС керує запуском, виконанням, призупиненням та завершенням програм (процесів). Вона забезпечує можливість одночасного виконання кількох програм (багатозадачність) шляхом розподілу процесорного часу.
- **Керування пам'яттю:** ОС розподіляє оперативну пам'ять між процесами, що виконуються, відстежує вільні та зайняті ділянки пам'яті. Сучасні ОС реалізують механізми **віртуальної пам'яті**, що дозволяють програмам використовувати більше пам'яті, ніж фізично встановлено, за рахунок використання частини дискового простору.
- **Керування файловою системою:** ОС організовує зберігання даних на довготривалих носіях (жорстких дисках, SSD) у вигляді файлів та каталогів (папок). Вона надає засоби для створення, видалення, копіювання, перейменування файлів та навігації по файловій структурі.
- **Керування пристроями введення/виведення (I/O):** ОС забезпечує взаємодію програм з периферійними пристроями (клавіатура, миша, монітор, принтер, мережева карта тощо) за допомогою спеціальних програм – **драйверів**.
- **Забезпечення інтерфейсу користувача:** ОС надає користувачеві засоби для взаємодії з комп'ютером. Це може бути **інтерфейс командного рядка (CLI)**, де команди вводяться з клавіатури, або **графічний інтерфейс користувача (GUI)**, що використовує вікна, меню, піктограми та маніпулятор (мишу).
- **Обробка помилок та збоїв:** ОС відстежує помилки в роботі апаратного та програмного забезпечення та інформує про них користувача.
- **Керування безпекою та захистом:** ОС забезпечує захист системи та даних від несанкціонованого доступу, вірусів та інших загроз. Це включає механізми аутентифікації (перевірка імені користувача та пароля), авторизації (надання прав доступу до ресурсів) та шифрування даних.
- **Мережева взаємодія:** Сучасні ОС містять засоби для роботи в комп'ютерних мережах, дозволяючи обмінюватися даними та спільно використовувати ресурси.

3. Класифікація та приклади ОС

Операційні системи можна класифікувати за різними ознаками:

- **За кількістю одночасно виконуваних задач:**
 - *Однозадачні:* Можуть виконувати лише одну програму в даний момент часу (наприклад, MS-DOS).
 - *Багатозадачні:* Дозволяють одночасно виконувати кілька програм, перемикаючись між ними (наприклад, Windows, macOS, Linux, OS/2, Unix).
- **За кількістю одночасно працюючих користувачів:**
 - *Однокористувацькі:* Розраховані на роботу одного користувача (наприклад, MS-DOS, Windows 9x/ME).
 - *Багатокористувацькі:* Підтримують одночасну роботу кількох користувачів, розмежовуючи їхні ресурси та дані (наприклад, Windows Server, Linux, macOS, Unix).
- **За типом інтерфейсу:**
 - *З командним рядком (CLI):* Взаємодія через текстові команди (наприклад, MS-DOS, консоль Linux).
 - *З графічним інтерфейсом (GUI):* Використання вікон, іконок, меню (наприклад, Windows, macOS, графічні оболонки Linux).
- **За призначенням / апаратною платформою:**
 - *Для настільних ПК та ноутбуків:* Windows, macOS, Linux (Ubuntu, Mint, Fedora тощо).
 - *Для серверів:* Windows Server, Linux (Debian, CentOS, Ubuntu Server), Unix.⁵¹
 - *Для мобільних пристроїв:* Android, iOS.
 - *Для вбудованих систем:* Спеціалізовані ОС для керування пристроями (наприклад, у маршрутизаторах, банкоматах).
 - *Для мейнфреймів:* Високопродуктивні ОС для великих обчислювальних систем (наприклад, OS/390, z/OS).
- **За розрядністю процесора:**
 - *16-розрядні, 32-розрядні, 64-розрядні:* Визначає максимальний обсяг пам'яті, що адресується, та ширину оброблюваних даних. Сучасні ОС переважно 64-розрядні.
- **За типом реалізації багатозадачності:**
 - *Невитісняюча (кооперативна):* Програма сама віддає керування ОС (наприклад, Windows 3.x).
 - *Витісняюча (пріоритетна):* ОС сама перериває виконання програми для передачі керування іншій (наприклад, Windows NT/XP/..., Linux, macOS, OS/2, Unix).

Таблиця 6. Класифікація операційних систем

Критерій класифікації	Типи ОС	Приклади
Кількість задач	Однозадачні, Багатозадачні	MS-DOS; Windows, Linux, macOS
Кількість користувачів	Однокористувацькі, Багатокористувацькі	MS-DOS, Windows 9x; Windows Server, Linux, macOS, Unix
Тип інтерфейсу	Командний рядок (CLI), Графічний (GUI)	MS-DOS, Linux console; Windows, macOS, Linux (KDE, Gnome)
Платформа/Призначення	Для ПК, Для серверів, Для мобільних пристроїв, Для вбудованих систем	Windows, macOS, Linux; Windows Server, Linux; Android, iOS; QNX, VxWorks

4. Драйвери пристроїв

Драйвер (англ. *driver* – водій) – це спеціальна системна програма, яка дозволяє операційній системі взаємодіяти з конкретним апаратним пристроєм. Кожен пристрій (відеокарта, принтер, мережевий адаптер, звукова карта, сканер, веб-камера тощо) потребує свого драйвера для коректної роботи.

Функції драйверів:

- Забезпечують "розуміння" операційною системою команд та можливостей конкретного пристрою.
- Перекладають стандартні запити ОС на мову, зрозумілу апаратному забезпеченню.
- Керують роботою пристрою, обробляють переривання від нього.
- Оптимізують продуктивність та функціональність апаратних компонентів.

Драйвери зазвичай розробляються виробниками обладнання і постачаються разом з пристроєм (на диску або доступні для завантаження з веб-сайту). Деякі базові драйвери для стандартних пристроїв (клавіатура, миша, базовий відеоадаптер) можуть входити до складу самої операційної системи. Встановлення правильних та актуальних драйверів є критично важливим для стабільної та ефективної роботи комп'ютера.

5. Утиліти (сервісні програми)

Утиліти (англ. *utility* – корисність, обслуговуюча програма) – це допоміжні програми, які розширюють можливості операційної системи або надають інструменти для обслуговування та налаштування комп'ютерної системи.

Призначення утиліт:

- **Діагностика системи:** Перевірка працездатності апаратних компонентів (пам'ять, диски) та ПЗ.
- **Обслуговування дисків:** Перевірка на помилки, дефрагментація (для HDD), очищення від тимчасових файлів.
- **Архівація даних:** Стиснення файлів для економії місця або зручності передачі (наприклад, WinRAR, 7-Zip).
- **Антивірусний захист:** Пошук, виявлення та видалення шкідливого ПЗ (вірусів, троянів, шпигунських програм).
- **Керування файлами:** Розширені можливості для роботи з файлами та папками порівняно зі стандартними засобами ОС (файлові менеджери, наприклад, Total Commander).
- **Оптимізація системи:** Налаштування параметрів ОС для підвищення продуктивності (наприклад, CCleaner).
- **Відновлення даних:** Спроба відновити випадково видалені файли.
- **Моніторинг системи:** Відображення інформації про стан системи (завантаження ЦП, використання пам'яті, температура).
- **Налаштування інтерфейсу та параметрів:** Зміна зовнішнього вигляду ОС, налаштування пристроїв.

Деякі утиліти входять до складу операційної системи, інші встановлюються користувачем окремо.

Таблиця 7. Приклади системних утиліт та їх призначення

Тип утиліти	Призначення	Приклади програм
Антивірус	Захист від шкідливого ПЗ	ESET NOD32, Avast, Zillya!

Архіватор	Стиснення файлів, створення архівів	WinRAR, 7-Zip, WinZip
Файловий менеджер	Розширене керування файлами та папками	Total Commander, FAR Manager
Діагностична утиліта	Перевірка стану апаратного забезпечення (HDD, RAM), моніторинг системи	Victoria, MemTest86+, AIDA64
Утиліта оптимізації	Очищення системи від сміття, налаштування автозавантаження, оптимізація ОС	CCleaner, Auslogics BoostSpeed
Утиліта дефрагментації	Оптимізація розташування файлів на HDD (для SSD не потрібна)	Вбудована у Windows, Defraggler

ТЕМА 1.6. ПРИКЛАДНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ТА ЛІЦЕНЗУВАННЯ

Мета: Розглянути класифікацію та приклади прикладного ПЗ, а також основні типи ліцензій на програмне забезпечення.

План:

- Прикладне програмне забезпечення (ППЗ): визначення та класифікація.
 - 1.1. ППЗ загального призначення.
 - 1.2. ППЗ спеціального призначення.
- Ліцензування програмного забезпечення: поняття та типи.

- 2.1. Власницьке (Proprietary) ПЗ (Комерційне, Shareware).
- 2.2. Безкоштовне (Freeware) ПЗ.
- 2.3. Вільне та відкрите (FOSS) ПЗ.
- 2.4. Суспільне надбання (Public Domain).

1. Прикладне програмне забезпечення (ППЗ): визначення та класифікація

Прикладне програмне забезпечення (ППЗ) – це категорія програм, які безпосередньо призначені для виконання конкретних завдань користувача та забезпечення його інформаційних потреб. На відміну від системного ПЗ, яке керує комп'ютером, прикладне ПЗ використовує можливості комп'ютера для вирішення завдань у певній предметній області.

Прикладне ПЗ можна класифікувати за різними ознаками, але найчастіше його поділяють на дві великі групи:

- **ППЗ загального призначення:** Програми, що мають універсальний характер і можуть використовуватися широким колом користувачів для вирішення типових завдань, незалежно від їхньої професійної діяльності.
- **ППЗ спеціального призначення (проблемно-орієнтоване):** Програми, розроблені для вирішення специфічних завдань у конкретній професійній галузі або для певної категорії користувачів.

1.1. ППЗ загального призначення

Це найпоширеніша категорія прикладних програм. До них належать:

- **Текстові процесори (редактори):** Призначені для створення, редагування та форматування текстових документів. Дозволяють працювати зі шрифтами, абзацами, стилями, вставляти таблиці, зображення, перевіряти орфографію тощо (наприклад, Microsoft Word, Google Docs, LibreOffice Writer).
- **Табличні процесори:** Дозволяють працювати з даними, організованими у вигляді електронних таблиць. Основні функції: введення даних, виконання обчислень за формулами, аналіз даних, побудова діаграм та графіків (наприклад, Microsoft Excel, Google Sheets, LibreOffice Calc).
- **Графічні редактори:** Призначені для створення та обробки графічних зображень. Поділяються на:
 - *Растрові редактори:* Робота з піксельними зображеннями (наприклад, Adobe Photoshop, GIMP, Paint.NET).
 - *Векторні редактори:* Робота з об'єктами, описаними математично (наприклад, Adobe Illustrator, CorelDRAW, Inkscape).
 - *3D-редактори:* Створення тривимірної графіки та анімації (наприклад, Blender, Autodesk 3ds Max).
- **Системи управління базами даних (СУБД) для користувачів:** Програми для створення та управління невеликими базами даних, введення, пошуку,

сортування та представлення інформації (наприклад, Microsoft Access, LibreOffice Base).

- **Редактори презентацій:** Програми для створення електронних презентацій, що складаються зі слайдів, які можуть містити текст, графіку, анімацію, відео та звук (наприклад, Microsoft PowerPoint, Google Slides, LibreOffice Impress).
- **Веб-браузери:** Програми для перегляду веб-сторінок та доступу до ресурсів Інтернету (наприклад, Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, Opera).
- **Поштові клієнти:** Програми для роботи з електронною поштою (наприклад, Microsoft Outlook, Mozilla Thunderbird, The Bat!).
- **Мультимедіа-програвачі:** Для відтворення аудіо- та відеофайлів (наприклад, VLC media player, Windows Media Player).
- **Комунікаційні програми:** Месенджери (Telegram, WhatsApp, Viber), програми для відеоконференцій (Zoom, Skype, Google Meet).
- **Архіватори:** Програми для стиснення файлів та створення архівів (наприклад, WinRAR, 7-Zip).
- **Файлові менеджери:** Утиліти для зручної роботи з файлами та папками (наприклад, Total Commander).

1.2. ППЗ спеціального призначення

Ці програми орієнтовані на конкретні професійні завдання:

- **Бухгалтерські системи:** Автоматизація бухгалтерського та податкового обліку (наприклад, М.Е.Дос).
- **Системи автоматизованого проектування (САПР / CAD):** Для інженерного проектування, створення креслень та 3D-моделей (наприклад, AutoCAD, SolidWorks, КОМПАС-3D).
- **Геоінформаційні системи (ГІС):** Для роботи з географічними даними та картами (наприклад, ArcGIS, QGIS).
- **Видавничі системи (DTP):** Для професійної верстки книг, журналів, газет (наприклад, Adobe InDesign, QuarkXPress).
- **Математичні та статистичні пакети:** Для складних наукових та інженерних розрахунків, статистичного аналізу (наприклад, MATLAB, Statistica, R).
- **Медичні інформаційні системи (МІС):** Автоматизація роботи медичних закладів, ведення електронних карток пацієнтів.
- **Банківські інформаційні системи:** Автоматизація банківської діяльності.
- **Системи управління взаємовідносинами з клієнтами (CRM):** Для управління продажами, маркетингом та обслуговуванням клієнтів (наприклад, Salesforce, Bitrix24).
- **Системи планування ресурсів підприємства (ERP):** Комплексна автоматизація управління ресурсами підприємства.
- **Навчальні програми та енциклопедії:** Електронні підручники, тренажери, тести, довідники.
- **Ігри:** Програми для розваг.

Таблиця 8. Класифікація Прикладного Програмного Забезпечення

Категорія	Призначення	Приклади програм
Загального призначення	Вирішення типових завдань широкого кола користувачів	MS Word, MS Excel, GIMP, Inkscape, MS Access, MS PowerPoint, Google Chrome, VLC media player, WinRAR, Telegram
Спеціального призначення	Вирішення специфічних завдань у конкретній професійній галузі або для певної мети	AutoCAD, ArcGIS, Adobe InDesign, MATLAB, CRM-системи, ERP-системи, навчальні програми, ігри

2. Ліцензування програмного забезпечення: поняття та типи

Ліцензія на програмне забезпечення – це юридичний документ (угода), який визначає права та обмеження користувача щодо використання, копіювання, модифікації та розповсюдження програмного продукту. Використання ПЗ без дотримання умов ліцензії є порушенням авторських прав розробника.

Існує багато типів ліцензій, які можна згрупувати за рівнем обмежень та вартістю:

- **Власницьке (Пропрієтарне) ПЗ (Proprietary Software):**

- Права на ПЗ належать компанії-розробнику або автору.
- Використання, копіювання, модифікація та розповсюдження суттєво обмежені або заборонені ліцензійною угодою.
- Вихідний код програми зазвичай закритий і недоступний користувачам.
- Підтипи:

- **Комерційне ПЗ (Commercial Software):** Створене з метою отримання прибутку; для використання необхідно придбати ліцензію (оплатити вартість ПЗ).

- **Умовно-безкоштовне ПЗ (Shareware):** Дозволяє користувачеві безкоштовно ознайомитися з програмою протягом певного

періоду або з обмеженою функціональністю ("спробуй перед покупкою"). Після закінчення пробного періоду або для отримання повного функціоналу необхідно придбати ліцензію.

- *Trialware*: Повнофункціональна версія, обмежена за часом використання (наприклад, 30 днів).
- *Demoware (Crippleware)*: Версія з обмеженою функціональністю (деякі можливості заблоковані).

- **Безкоштовне ПЗ (Freeware):**

- Програми, які можна використовувати безкоштовно без обмежень за часом.
- Авторські права залишаються у розробника, вихідний код зазвичай закритий.
- Ліцензія може містити певні обмеження, наприклад, дозволяти безкоштовне використання лише для особистих, некомерційних цілей.
- *Adware / Ad-supported*: Безкоштовне ПЗ, яке фінансується за рахунок показу реклами користувачеві під час роботи або встановлення.

- **Вільне та Відкрите ПЗ (Free and Open Source Software - FOSS):**

- Надає користувачам чотири основні свободи:
 - Свобода запускати програму для будь-яких цілей.
 - Свобода вивчати, як працює програма, та модифікувати її (вимагає доступу до вихідного коду).
 - Свобода розповсюджувати копії програми.
 - Свобода розповсюджувати модифіковані версії програми.
- Термін "Free" тут означає "свободу", а не "безкоштовність", хоча більшість такого ПЗ є безкоштовним.
- **Open Source (Відкритий код)**: Акцентує увагу на доступності вихідного коду для вивчення та модифікації.
- **Ліцензії FOSS**: Існує багато ліцензій (GNU GPL, MIT, Apache, BSD тощо), які реалізують принципи вільного та відкритого ПЗ.
 - *Copyleft (наприклад, GNU GPL)*: Вимагає, щоб похідні роботи (модифіковані версії) також розповсюджувалися під тією ж вільною ліцензією.
 - *Non-copyleft (дозвільні ліцензії, наприклад, MIT, BSD)*: Дозволяють включати код у власницькі продукти без обов'язку відкривати код похідної роботи.
- Приклади: ОС Linux, веб-сервер Apache, СУБД MySQL/PostgreSQL, офісний пакет LibreOffice, браузер Firefox, графічний редактор GIMP.

- **Суспільне надбання (Public Domain):**

- Програмне забезпечення, на яке не поширюються авторські права (автор відмовився від прав або термін їх дії закінчився).
- Можна використовувати, копіювати, модифікувати та розповсюджувати без будь-яких обмежень.

Розуміння типів ліцензій є важливим для легального використання програмного забезпечення та вибору інструментів, що відповідають потребам та можливостям користувача чи організації.

Таблиця 9. Основні типи ліцензій на ПЗ

Тип ліцензії	Основні характеристики	Приклади
Власницька / Комерційна	Платне використання, закритий код, обмежені права на копіювання/модифікацію.	Microsoft Windows, Microsoft Office, Adobe Photoshop
Shareware (Умовно-безкоштовна)	Безкоштовний пробний період (Trialware) або обмежена функціональність (Demoware), потім потрібна оплата.	WinRAR (trial), багаторічний (demo)
Freeware (Безкоштовна)	Безкоштовне використання, закритий код, права у автора, можливі обмеження (напр., некомерційне використання).	Skype, Adobe Acrobat Reader, CCleaner (Free version), Opera ⁶⁴
FOSS (Вільне та Відкрите)	Свобода використання, вивчення, модифікації, поширення. Відкритий вихідний код. Може бути платним або безкоштовним.	Linux, LibreOffice, Firefox, GIMP, Apache Web Server, MySQL

Public Domain (Суспільне надбання)	Відсутність авторських прав, повна свобода використання та поширення.	Деякі старі програми, спеціально передані у суспільне надбання
--	---	--

ТЕМА 1.7. ТЕКСТОВИЙ ПРОЦЕСОР MS WORD: ОСНОВИ РОБОТИ

Мета: Ознайомити здобувачів вищої освіти з призначенням та інтерфейсом текстового процесора MS Word, навчити базовим операціям зі створення, відкриття, збереження документів, введення та редагування тексту, а також використанню вбудованих інструментів перевірки правопису.

План:

1. Призначення та інтерфейс програми.
2. Створення, відкриття та збереження документів.
3. Введення та редагування тексту.
4. Перевірка правопису.

1. Призначення та інтерфейс програми

Microsoft Word є потужним текстовим процесором, призначеним для створення, редагування, форматування та друку текстових документів різної складності – від простих листів до складних звітів, статей та книг. Він є стандартом де-факто у світі обробки документів і пропонує широкий спектр можливостей для роботи з текстом, зображеннями, таблицями та іншими об'єктами.

Основні елементи інтерфейсу MS Word (актуальні для сучасних версій, таких як Word 2016, 2019, 2021 та Microsoft 365) включають:

- **Стрічка (Ribbon):** Головний елемент керування, розташований у верхній частині вікна. Організована за принципом вкладок (табів) – "Файл", "Головна", "Вставлення", "Макет" (або "Розмітка сторінки", "Дизайн", "Посилання", "Розсилки", "Рецензування", "Вигляд", "Розробник" (за потреби)). Кожна вкладка містить групи пов'язаних команд, представлених кнопками або меню (наприклад, на вкладці "Головна" є групи "Буфер обміну", "Шрифт", "Абзац", "Стилі", "Редагування"). Стрічку можна приховувати або розгортати для оптимізації робочої області.

- **Область Backstage (Вкладка "Файл"):** Надає доступ до команд керування файлами та параметрів програми: "Створити" (новий документ, з шаблону), "Відкрити" (існуючий документ), "Зберегти", "Зберегти як" (для першого збереження або збереження під іншим ім'ям/форматом), "Друк", "Спільний доступ", "Експорт" (наприклад, у PDF), "Обліковий запис" (інформація про користувача та продукт), "Параметри" (налаштування програми).

- **Панель швидкого доступу (Quick Access Toolbar):** Зазвичай знаходиться над стрічкою зліва, хоча її можна розташувати і під стрічкою. Містить кнопки для часто використовуваних команд, які доступні незалежно від поточної вкладки стрічки (за замовчуванням "Зберегти", "Скасувати" – дія назад, "Повторити" – дія вперед). Користувач може налаштовувати цю панель, додаючи або видаляючи команди.

- **Рядок заголовка:** Розташований у самому верху вікна, відображає ім'я поточного документа та назву програми.

- **Лінійки:** Вертикальна та горизонтальна лінійки, що відображають розміри сторінки, поля, позиції табуляції та відступи абзаців. Допмагають візуально орієнтуватися в документі та точно розташовувати елементи.

- **Робоча область:** Основна частина вікна, де відображається вміст документа. Тут здійснюється безпосереднє введення, редагування та форматування тексту. Містить текстовий курсор – миготливу вертикальну риску, що вказує на поточне місце введення або редагування.

- **Смуги прокручування:** Вертикальна та горизонтальна смуги, що дозволяють переміщуватися по вмісту документа, якщо він не поміщається повністю у вікні.

- **Рядок стану:** Розташований внизу вікна. Надає корисну інформацію про поточний документ (номер поточної сторінки та загальну кількість сторінок, кількість слів у документі, мова перевірки правопису). Також містить кнопки для зміни режимів перегляду документа (режим читання, режим розмітки, режим веб-документа) та повзунок для зміни масштабу відображення документа.

2. Створення, відкриття та збереження документів

Операції з файлами є базовими для роботи в будь-якій офісній програмі, зокрема в MS Word. Вони доступні через вкладку "Файл" (Область Backstage).

- **Створення нового документа:**

- Вибирається "Файл" -> "Створити".
- Пропонується створити "Пустий документ" – найпоширеніший варіант для створення документа з нуля.
- Також можна обрати один з доступних **шаблонів** (Templates) – попередньо відформатованих документів для типових завдань (резюме, листи, брошури, звіти, календарі тощо). Використання шаблонів значно прискорює створення документів зі стандартизованим виглядом.

Можна використовувати вбудовані шаблони або завантажувати нові з офісу Microsoft Office.com.

- **Відкриття існуючого документа:**

- Вибирається "Файл" -> "Відкрити".
- Необхідно вказати місцезнаходження файлу: на локальному комп'ютері ("Цей ПК"), у хмарних сховищах (OneDrive), на мережевих дисках або в інших розташуваннях.
- Можна переглядати "Останні" документи для швидкого доступу.
- Після вибору місця необхідно знайти та вибрати потрібний файл (як правило, з розширенням .docx для сучасних версій Word).

- **Збереження документа:**

- **"Зберегти" (Save):** Використовується для збереження поточних змін у вже існуючому файлі. Якщо документ ще не був збережений (новий документ), ця команда діє як "Зберегти як". Найчастіше використовується поєднання клавіш Ctrl+S або кнопка "Зберегти" на Панелі швидкого доступу.
- **"Зберегти як" (Save As):** Використовується в наступних випадках:
 - Перше збереження нового документа.
 - Збереження існуючого документа під новим ім'ям.
 - Збереження документа в іншому форматі файлу (наприклад, .doc для сумісності зі старими версіями Word, .pdf, .rtf, .txt, веб-сторінка).
 - Збереження документа в іншому місці.
- При використанні "Зберегти як" необхідно вибрати місце збереження, ввести ім'я файлу та вибрати тип файлу зі списку, що розкривається. Стандартний формат для MS Word починаючи з версії 2007 – **.docx**. Цей формат базується на XML (Office Open XML) і є більш стабільним та меншим за розміром порівняно зі старим бінарним форматом .doc.

3. Введення та редагування тексту

Введення тексту здійснюється безпосередньо в робочій області документа з клавіатури у позиції текстового курсора. Під час введення тексту Word автоматично переносить слова на наступний рядок, якщо вони не поміщаються в поточному (режим "перенесення по словах"). Для примусового переходу на новий абзац використовується клавіша Enter.

Редагування тексту включає широкий спектр операцій для зміни вмісту документа:

- **Виділення тексту:** Необхідне для застосування форматування (зміна шрифту, кольору тощо), виконання операцій копіювання/переміщення/видалення, застосування стилів або інших дій до конкретного фрагмента тексту. Word пропонує різноманітні зручні способи виділення:

- Клацання та перетягування миші з натиснутою лівою кнопкою – виділення довільного фрагмента тексту.
- Подвійне клацання мишею – виділення слова.
- Потрійне клацання мишею – виділення абзацу.
- Клацання на полі виділення (сіра область ліворуч від тексту) навпроти рядка – виділення одного рядка.
- Перетягування миші по полю виділення – виділення кількох рядків.
- Двічі клацнути по полю виділення – виділення абзацу.
- Тричі клацнути по полю виділення – виділення всього документа.
- Виділення за допомогою клавіатури: Утримуючи клавішу Shift та використовуючи стрілки керування курсором (вліво, вправо, вгору, вниз), Page Up, Page Down, Home, End.
- Комбінації клавіш: Ctrl+A – виділення всього документа. Ctrl+Shift + стрілки – виділення до/від слова/абзацу.
- **Копіювання, Вирізання та Вставка (Робота з буфером обміну):** Ці операції використовують системний буфер обміну – тимчасове сховище даних.
 - Виділити потрібний фрагмент тексту (або інший об'єкт – зображення, таблицю).
 - **Копіювати (Copy):** Скопіювати виділений елемент у буфер обміну, залишаючи його на поточному місці. Використовується комбінація клавіш Ctrl+C або кнопка "Копіювати" в групі "Буфер обміну" на вкладці "Головна".
 - **Вирізати (Cut):** Скопіювати виділений елемент у буфер обміну та видалити його з поточного місця. Використовується комбінація клавіш Ctrl+X або кнопка "Вирізати" в групі "Буфер обміну".
 - Встановити текстовий курсор у місце вставки.
 - **Вставити (Paste):** Вставити вміст буфера обміну у поточну позицію курсора. Використовується комбінація клавіш Ctrl+V або кнопка "Вставити" в групі "Буфер обміну". Word пропонує різні параметри вставки (зберегти вихідне форматування, об'єднати форматування, зберегти тільки текст), доступні за допомогою маленької стрілки під кнопкою "Вставити" або при клацанні правою кнопкою миші.
- **Видалення тексту:**
 - Клавіша Backspace – видаляє символ ліворуч від курсора.
 - Клавіша Delete – видаляє символ праворуч від курсора або весь виділений фрагмент тексту.
- **Перетягування (Drag and Drop):** Виділений текст можна перемістити в інше місце документа, просто перетягнувши його мишею з натиснутою лівою кнопкою. Для копіювання тексту за допомогою перетягування потрібно утримувати клавішу Ctrl під час перетягування.
- **Знайти та Замінити (Find and Replace):** Потужні інструменти для пошуку певного тексту у документі та його заміни на інший текст. Доступні на вкладці

"Головна" в групі "Редагування" або за допомогою Ctrl+F (Знайти) та Ctrl+H (Замінити). Можуть використовуватися для пошуку та заміни не тільки тексту, але й певного форматування, спеціальних символів тощо.

4. Перевірка Правопису

MS Word має вбудовані інструменти для перевірки орфографії та граматики, що є важливим етапом для підвищення якості та професійності документа. Ефективність перевірки залежить від мовних налаштувань Word та встановлених словників.

- **Автоматична перевірка:** Під час введення тексту Word автоматично перевіряє орфографію та граматику і підкреслює потенційні помилки:
 - Орфографічні помилки (невідомі слова, неправильне написання) – підкреслюються червоною хвилястою лінією.
 - Граматичні помилки (неправильна узгодженість, пунктуація, структура речення) – підкреслюються синьою або зеленою хвилястою лінією (колір може відрізнятися залежно від версії Word та налаштувань).
 - Стилiстичні рекомендації або питання ясності – можуть підкреслюватися фіолетовою або іншою кольоровою пунктирною лінією (особливо при використанні Microsoft Editor).
 - Для перегляду запропонованих виправлень достатньо клацнути правою кнопкою миші на підкресленому слові чи фразі. З'явиться контекстне меню з варіантами виправлення, можливістю пропустити помилку, додати слово до словника тощо.
- **Ручна перевірка:** Запускається для комплексної перевірки всього документа або його частини.
 - Перейдіть на вкладку "Рецензування" -> група "Правопис" -> кнопка "Правопис і граматика".
 - Відкривається панель або діалогове вікно "Правопис", де послідовно відображаються знайдені помилки. Для кожної помилки пропонуються варіанти виправлення. Користувач може вибрати один із варіантів, "Пропустити один раз", "Пропустити все" (ігнорувати це слово/фразу в усьому документі), "Додати до словника" (якщо слово написане правильно, але відсутнє у стандартному словнику, його можна додати до користувацького словника Word, щоб воно більше не позначалося як помилка), "Автозаміна" (автоматично виправляти цю помилку надалі).
- **Microsoft Editor:** У новітніх версіях Word та Microsoft 365 інтегровано більш досконалий інструмент – Microsoft Editor. Він виходить за межі базової орфографії та граматики, надаючи розширені рекомендації щодо ясності, стислості, формальності, включеності, лексики тощо. Editor може працювати з текстом багатьма мовами та допомагає вдосконалити стиль письма. Його налаштування доступні через "Файл" -> "Параметри" -> "Правопис".

- **Мова перевірки:** Важливо правильно встановити мову для перевірки правопису для різних частин документа, особливо якщо документ містить текст різними мовами. Мову можна змінити через "Рецензування" -> "Мова" -> "Мова перевірки правопису".

ТЕМА 1.8. ТЕКСТОВИЙ ПРОЦЕСОР MS WORD: ФОРМАТУВАННЯ ТА ОБ'ЄКТИ

Мета: Навчити здобувачів вищої освіти застосовувати різноманітні параметри форматування до символів, абзаців та сторінок, ефективно використовувати стилі для стандартизації документів, а також додавати та формувати таблиці, зображення та інші графічні об'єкти, створювати автоматичний зміст.

План:

1. Форматування символів.
2. Форматування абзаців.
3. Форматування сторінок.
4. Робота зі стилями.
5. Вставка та форматування таблиць.
6. Вставка та форматування зображень та фігур.
7. Створення змісту.

1. Форматування символів

Форматування символів (або шрифту) дозволяє змінювати зовнішній вигляд окремих символів, слів, фраз або невеликих фрагментів тексту. Основні інструменти для форматування символів зібрані на вкладці "Головна" в групі "Шрифт". Додаткові параметри доступні у діалоговому вікні "Шрифт", яке відкривається за допомогою маленької стрілки в правому нижньому куті групи "Шрифт" або комбінацією Ctrl+D.

- **Шрифт (Font):** Вибір гарнітури шрифту (Font Family) – набору символів певного накреслення та стилю (наприклад, Times New Roman, Arial, Calibri, Verdana). Вибір шрифту впливає на загальне візуальне сприйняття тексту.
- **Розмір (Font Size):** Встановлення розміру шрифту в пунктах (pt). 1 пункт приблизно дорівнює 1/72 дюйма. Стандартні розміри для основного тексту – 10-14 pt.
- **Накреслення (Font Style):** Застосування стилів накреслення для виділення тексту:
 - **Жирний (Bold, Ctrl+B):** Робить текст товстішим.

- *Курсив* (Italic, Ctrl+I): Нахиляє текст.
- Підкреслений (Underline, Ctrl+U): Додає лінію під текстом. Доступні різні типи підкреслення та кольори.
- Закреслений: Перекреслює текст горизонтальною лінією.
- Подвійне закреслення, Верхній індекс (Superscript), Нижній індекс (Subscript) – доступні у діалоговому вікні "Шрифт".
- **Колір шрифту (Font Color):** Зміна кольору тексту. Можна обрати зі стандартної палітри, кольорів теми або задати користувацький колір.
- **Колір виділення тексту (Text Highlight Color):** Застосування фонового кольору до тексту, імітуючи ефект маркера-виділювача. Це не те саме, що заливка комірки чи абзацу.
- **Ефекти тексту (Text Effects and Typography):** Додавання до тексту художніх ефектів, таких як тінь, віддзеркалення, світіння, рельєф, контур.
- **Зміна регістра (Change Case):** Автоматична зміна регістру виділеного тексту: речення (з великої літери на початку), всі малі, ВСІ ВЕЛИКІ, Починати З Великих Слів, ЗМІНИТИ РЕГІСТР.
- **Очистити все форматування (Clear All Formatting):** Видаляє все форматування (шрифт, розмір, стиль, колір тощо), застосоване до виділеного тексту, повертаючи його до стандартного стилю абзацу (зазвичай "Звичайний").

2. Форматування абзаців

Форматування абзаців визначає вигляд та розташування цілих абзаців тексту. Абзацом у Word вважається будь-який текст, що закінчується символом кінця абзацу (¶), який з'являється при натисканні клавіші Enter. Основні інструменти форматування абзаців знаходяться на вкладці "Головна" в групі "Абзац", а також на вкладці "Макет" в групі "Абзац". Більш детальні налаштування доступні у діалоговому вікні "Абзац" (відкривається стрілкою в правому нижньому куті групи "Абзац").

- **Вирівнювання (Alignment):** Визначає, як рядки абзацу розташовуються відносно лівого та правого полів сторінки:
 - По лівому краю (Align Left, Ctrl+L): Найпоширеніше вирівнювання для звичайного тексту, лівий край рівний, правий – ні.
 - По центру (Center, Ctrl+E): Текст центрується відносно полів. Використовується для заголовків, підписів.
 - По правому краю (Align Right, Ctrl+R): Правий край рівний, лівий – ні. Використовується рідше, наприклад, для дат або підписів.
 - По ширині (Justify, Ctrl+J): Текст вирівнюється по обох краях (лівому та правому) шляхом автоматичного регулювання пробілів між словами. Використовується у книгах, статтях для створення щільного блоку тексту.
- **Відступи (Indentation):** Зсув абзацу або його частин відносно лівого та правого полів сторінки або відносно меж, встановлених для розділу/документа.

- Відступ зліва (Left Indent): Зсуває весь абзац від лівого поля.
- Відступ справа (Right Indent): Зсуває весь абзац від правого поля.
- Відступ першого рядка (First Line Indent): Відступ лише першого рядка абзацу ("червоний рядок").
- Виступ (Hanging Indent): Відступ всіх рядків абзацу, крім першого (використовується, наприклад, для списків літератури). Налаштовується через діалогове вікно "Абзац" або за допомогою маркерів на горизонтальній лінійці.
- **Інтервали (Spacing):** Визначає відстань між рядками всередині абзацу та відстань між сусідніми абзацами.
 - Міжрядковий інтервал (Line Spacing): Відстань між базовими лініями рядків всередині одного абзацу (одинарний, півтора рядка, подвійний, мінімальний, точний, множник). Налаштовується через кнопку "Міжрядковий інтервал" у групі "Абзац" або в діалоговому вікні.
 - Міжабзацний інтервал (Spacing Before/After): Відстань, яка додається перед першим рядком абзацу або після останнього рядка абзацу. Дозволяє візуально відокремлювати абзаци без використання порожніх рядків (що є кращою практикою). Налаштовується у діалоговому вікні "Абзац" або на вкладці "Макет".
- **Маркери та нумерація (Bullets and Numbering):** Створення маркованих (символьні маркери) та нумерованих (цифри, літери, римські числа) списків. Дозволяє структурувати інформацію та покращити її сприйняття. Підтримується багаторівнева нумерація.
- **Багаторівневий список (Multilevel List):** Створення ієрархічних списків зі складною нумерацією (наприклад, 1., 1.1., 1.1.1., 2., 2.1. тощо). Часто використовується для структури звітів, дипломних робіт.
- **Межі та заливка (Borders and Shading):** Додавання рамок (ліній) навколо абзацу (один або кілька боків, рамка навколо всього абзацу) та зміна кольору фону абзацу (заливка).

3. Форматування сторінок

Форматування сторінок визначає загальний вигляд документа на аркуші паперу або екрані. Налаштування, що впливають на цілу сторінку або розділ, зазвичай знаходяться на вкладці "Макет" (Layout).

- **Поля (Margins):** Встановлення розмірів порожніх областей по периметру сторінки – відступів від країв паперу до меж тексту. Встановлюються окремо для верхнього, нижнього, лівого та правого полів. Можна вибрати стандартні набори полів або встановити користувацькі значення. Важливі для друку та візуального сприйняття документа.
- **Орієнтація (Orientation):** Вибір книжкової (Portrait - вертикальної) або альбомної (Landscape - горизонтальної) орієнтації сторінки.

- **Розмір (Size):** Вибір стандартного розміру паперу, на якому буде друкуватися документ (A4 – найпоширеніший формат, A5, Letter, Legal тощо). Можна також встановити користувацький розмір сторінки.
- **Колонки (Columns):** Розбиття тексту на сторінці або в межах розділу на кілька вертикальних колонок. Використовується, наприклад, для газетних статей, брошур. Можна задати кількість колонок, їх ширину та інтервал між ними.
- **Розриви (Breaks):** Вставка спеціальних маркерів, що керують потоком тексту та розташуванням елементів. Різні типи розривів дозволяють:
 - Розрив сторінки (Page Break): Примусово переносить текст на наступну сторінку (Ctrl+Enter).
 - Розрив колонки (Column Break): Переносить текст на початок наступної колонки.
 - Розриви розділів (Section Breaks): Найважливіші для складного форматування. Дозволяють поділити документ на розділи, кожен з яких може мати власні параметри форматування сторінки (поля, орієнтація, розмір, кількість колонок, колонтитули, нумерація сторінок), відмінні від інших розділів. Типи: наступна сторінка, поточна сторінка (для колонок), парна сторінка, непарна сторінка.
- **Нумерація рядків (Line Numbers):** Додавання послідовних номерів ліворуч від кожного рядка тексту (використовується, наприклад, у юридичних документах або коді).
- **Розстановка переносів (Hyphenation):** Автоматична або ручна розстановка знаків переносу в кінці рядків для зменшення нерівномірності правого краю тексту, вирівняного по лівому краю, або покращення вигляду тексту, вирівняного по ширині.
- **Колонтитули (Headers and Footers):** Області, розташовані у верхній (верхній колонтитул) та нижній (нижній колонтитул) частині кожної сторінки або в межах розділу. Використовуються для розміщення повторюваної інформації: номери сторінок, назва документа/розділу, ім'я автора, дата, логотип. Доступ до редагування колонтитулів здійснюється подвійним клацанням миші у верхньому чи нижньому полі сторінки або через вкладку "Вставлення" -> група "Колонтитули". Можна налаштувати різні колонтитули для першої сторінки, парних та непарних сторінок, а також для різних розділів документа.
- **Нумерація сторінок:** Автоматичне додавання номерів сторінок. Використовується функція "Номер сторінки" на вкладці "Вставлення" -> група "Колонтитули". Можна вибрати місце розташування номера (вгорі, внизу, на полях), формат нумерації та вказати, з якої сторінки починати нумерацію або яким числом.

4. Робота зі стилями

Стилі є одним із найпотужніших інструментів у Word для забезпечення єдиного, послідовного та професійного форматування документа, а також для його

структурування.

- **Поняття стилю:** Стиль – це іменований набір параметрів форматування, який може включати форматування символів (шрифт, розмір, колір, накреслення) та/або форматування абзаців (вирівнювання, відступи, інтервали, межі, заливка, маркери, нумерація). Існують також стилі для таблиць та списків.
- **Переваги використання стилів:**
 - **Швидкість:** Застосування складного комбінованого форматування (наприклад, для заголовка – певний шрифт, розмір, жирний, інтервал перед/після) одним клацанням миші.
 - **Послідовність:** Забезпечення однакового вигляду однотипних елементів (усі заголовки першого рівня виглядають однаково, весь основний текст має однакове форматування) у всьому документі. Це критично важливо для довгих документів.
 - **Легкість зміни форматування:** Якщо потрібно змінити форматування всіх елементів, до яких застосовано певний стиль (наприклад, змінити колір всіх заголовків першого рівня), достатньо модифікувати (змінити параметри) сам стиль. Усі фрагменти тексту, відформатовані цим стилем, автоматично оновлять свій вигляд.
 - **Структурування документа:** Використання вбудованих стилів заголовків ("Заголовок 1", "Заголовок 2", "Заголовок 3" тощо) є необхідною умовою для автоматичного створення змісту, навігаційної області, а також для коректної структури при експорті в PDF або інших форматах.
 - **Навігація:** Навігаційна область (Вигляд -> Показати -> Область переходів) використовує стилі заголовків для створення інтерактивного змісту, що дозволяє швидко переміщуватися по розділах документа.
- **Типи стилів:**
 - Стилі абзаців (Paragraph Styles): Впливають як на форматування символів, так і на форматування абзаців. Застосовуються до цілих абзаців.
 - Стилі символів (Character Styles): Впливають тільки на форматування символів. Застосовуються до виділених фрагментів тексту в межах абзацу.
 - Зв'язані стилі (Linked Styles): Можуть застосовуватися як стилі абзаців або стилі символів.
 - Стилі таблиць, Стилі списків.
- **Застосування стилів:** Виділити текст або абзац, до якого потрібно застосувати стиль, та вибрати потрібний стиль з галереї стилів на вкладці "Головна" -> група "Стилі". Галерея відображає найпоширеніші стилі. Повний список стилів можна побачити, відкривши панель стилів (натиснувши стрілку в правому нижньому куті групи "Стилі").

- **Створення та модифікація стилів:** Користувач може створювати власні стилі на основі існуючого форматування або "з нуля". Також можна змінювати параметри існуючих вбудованих або користувацьких стилів через панель стилів (клацнути правою кнопкою миші на стилі -> "Змінити"). При модифікації стилю можна вибрати, чи застосовувати зміни тільки в поточному документі, чи в шаблоні, на якому базується документ (щоб новий стиль був доступний у всіх майбутніх документах на основі цього шаблону).

5. Вставка та форматування таблиць

Таблиці використовуються для впорядкованого представлення даних у вигляді рядків та стовпців. У Word таблиці є потужним інструментом для організації інформації, не тільки числових даних, але й тексту, зображень тощо.

- **Вставка таблиці:** Здійснюється через вкладку "Вставлення" -> група "Таблиці".
 - Вибір розміру візуально: Навести курсор на сітку, щоб вказати потрібну кількість рядків та стовпців.
 - Вставка таблиці (Insert Table): Відкриває діалогове вікно, де можна точно вказати кількість стовпців та рядків, а також налаштувати автопідбір ширини стовпців.
 - Намалювати таблицю (Draw Table): Дозволяє "намалювати" таблицю вручну, проводячи лінії стовпців та рядків олівцем.
 - Вставити таблицю Excel: Вставляє в документ об'єкт електронної таблиці Excel, який можна редагувати безпосередньо у Word, використовуючи функціонал Excel.
 - Експрес-таблиці (Quick Tables): Вставка готових шаблонів таблиць (наприклад, календар, списки).
- **Робота з таблицею:** Переміщення між комірками здійснюється клавішами Tab (вперед) та Shift+Tab (назад), а також стрілками керування курсором.
- **Форматування таблиці:** Після створення або виділення таблиці в стрічці з'являються дві контекстні вкладки, присвячені роботі з таблицями: "Конструктор таблиць" (Table Design) та "Макет" (Layout).
- **Вкладка "Конструктор таблиць":** Відповідає за зовнішній вигляд таблиці.
 - Стили таблиць (Table Styles): Застосування готових візуальних стилів до таблиці (різні кольори, межі, заливка, спеціальне форматування для першого/останнього рядка/стовпця).
 - Параметри стилів таблиць: Включення/виключення спеціального форматування для окремих частин таблиці (рядок заголовка, перший стовпець, підсумковий рядок, остання колонка, смугасті рядки/стовпці).
 - Межі (Borders): Налаштування меж комірок (тип, товщина, колір лінії) та їх відображення.

- Заливка (Shading): Зміна кольору фону для окремих комірок або діапазонів.
- **Вкладка "Макет":** Відповідає за структуру та розташування елементів таблиці.
 - Виділення (Selection): Виділення комірок, рядків, стовпців, усієї таблиці.
 - Рядки та Стовпці (Rows & Columns): Вставка нових рядків (вище/нижче) та стовпців (ліворуч/праворуч). Видалення рядків, стовпців, комірок, таблиці.
 - Об'єднання (Merge): Об'єднання кількох виділених комірок в одну.
 - Розбиття (Split): Розбиття однієї комірки на кілька. Розбиття таблиці на дві окремі таблиці.
 - Розмір комірки (Cell Size): Налаштування висоти рядків та ширини стовпців (точне значення, автопідбір за вмістом/вікном). Рівномірний розподіл ширини стовпців або висоти рядків.
 - Вирівнювання (Alignment): Горизонтальне та вертикальне вирівнювання тексту всередині комірки (по верхньому краю зліва, по центру по центру, по нижньому краю справа тощо). Напрямок тексту.
 - Дані (Data): Сортування даних у таблиці, повторення рядків заголовка на кожній сторінці, перетворення таблиці на текст, вставка формул у комірки таблиці (наприклад, SUM, AVERAGE).
- **Перетворення тексту на таблицю та навпаки:** Можливість перетворити наявний текст, розділений певними символами (знаками табуляції, комою, абзацами), на таблицю, і навпаки – перетворити таблицю на текст.

6. Вставка та форматування зображень та фігур

Word дозволяє додавати до документів графічні об'єкти для ілюстрації, оформлення та візуалізації інформації.

- **Вставка зображення (Pictures):**
 - Вкладка "Вставлення" -> група "Зображення".
 - "Цей пристрій" (This Device): Вставка зображення з файлу, збереженого на вашому комп'ютері.
 - "Онлайн-зображення" (Online Pictures) або "Зображення з фотостокових ресурсів" (Stock Images): Пошук та вставка зображень з Інтернету (Bing Image Search) або з бібліотеки стокових зображень (доступно в Microsoft 365).
- **Вставка фігури (Shapes):**
 - Вкладка "Вставлення" -> група "Ілюстрації" -> "Фігури".
 - Вибір потрібної фігури з бібліотеки (лінії, прямокутники, основні фігури, стрілки, блок-схеми, зірки та стрічки, виносні написи).
 - "Малювання" фігури на сторінці за допомогою миші.

- **Вставка піктограм (Icons), тривимірних моделей (3D Models), SmartArt:** Доступні також через вкладку "Вставлення" в групі "Ілюстрації". Піктограми – векторні зображення. SmartArt – графічні об'єкти для візуалізації інформації (списки, процеси, цикли, ієрархії).
- **Форматування зображень/фігур:** Після виділення вставленого об'єкта (зображення, фігури) в стрічці з'являється контекстна вкладка "Формат зображення" (Picture Format) або "Формат фігури" (Shape Format) з інструментами для налаштування їх вигляду та розташування.
- **Для зображень:**
 - Корекція (Corrections): Налаштування яскравості, контрастності, різкості.
 - Колір (Color): Зміна насиченості, відтінку, застосування кольорових фільтрів.
 - Художні ефекти (Artistic Effects): Застосування ефектів, що імітують малюнок, акварель, розмиття тощо.
 - Стиснення зображень (Compress Pictures): Зменшення розміру файлу зображення для зменшення загального розміру документа.
 - Змінити зображення (Change Picture): Заміна зображення на інше, зберігаючи його форматування.
 - Скинути зображення (Reset Picture): Видалення всього форматування, застосованого до зображення.
 - Стили зображень (Picture Styles): Застосування готових стилів рамок, тіней, віддзеркалень, рельєфу.
 - Межі зображення (Picture Border), Ефекти зображення (Picture Effects), Макет зображення (Picture Layout - перетворення на SmartArt).
- **Обтікання текстом (Text Wrapping):** Ключове налаштування, що визначає, як текст документа розташовуватиметься навколо зображення або іншого об'єкта. Найпоширеніші варіанти: "У тексті" (In Line with Text - об'єкт поводить себе як великий символ), "Квадрат" (Square), "За контуром" (Tight), "Крізь" (Through), "Зверху та знизу" (Top and Bottom), "Перед текстом" (In Front of Text), "За текстом" (Behind Text).
- **Розташування (Position):** Вибір стандартного розташування об'єкта на сторінці відносно полів або тексту.
 - Вирівнювання (Align): Вирівнювання об'єкта відносно сторінки, полів або інших об'єктів (по лівому краю, по центру, по правому краю, вгорі, посередині, внизу).
 - Перемістити вперед/назад (Bring Forward/Send Backward): Керування шарами об'єктів (який об'єкт знаходиться поверх іншого).
 - Область виділення (Selection Pane): Дозволяє бачити список усіх об'єктів на сторінці та керувати їх видимістю та порядком.
 - Обрізання (Crop): Видалення непотрібних частин зображення.

- Розмір (Size): Точне встановлення висоти та ширини об'єкта, збереження пропорцій.
 - Альтернативний текст (Alt Text): Додавання опису зображення для покращення доступності документа (для людей, що використовують програми читання з екрана).
- **Для фігур:** Схожі інструменти, але з акцентом на заливку фігури (Shape Fill), контур фігури (Shape Outline - колір, товщина, тип лінії), ефекти фігури (тінь, світіння, об'єм), додавання тексту до фігури (Add Text).

7. Створення змісту

Автоматичний зміст (Таблиця змісту) є обов'язковим елементом для офіційних документів, звітів, дипломних робіт тощо. Він значно полегшує навігацію у великих документах і швидко оновлюється при внесенні змін.

- **Передумова:** Для того, щоб Word міг автоматично створити зміст, необхідно послідовно застосувати вбудовані стилі заголовків ("Заголовок 1" для основних розділів, "Заголовок 2" для підрозділів, "Заголовок 3" для підрозділів тощо) до відповідних назв розділів та підрозділів у тексті документа. Word сканує документ, знаходить абзаци, до яких застосовані стилі заголовків, збирає їхній текст та номери сторінок і формує на їх основі зміст.
- **Вставка змісту:**
 - Встановити текстовий курсор у те місце документа, де має бути розташований зміст (зазвичай на початку документа, після титульної сторінки).
 - Перейти на вкладку "Посилання" (References) -> група "Зміст" (Table of Contents).
 - Натиснути кнопку "Зміст". У випадаючому меню можна вибрати один із запропонованих **автоматичних стилів** змісту ("Автоматична таблиця 1", "Автоматична таблиця 2"). Ці стилі автоматично формують зміст на основі стилів заголовків у документі. Можна також вибрати "Настроюваний зміст..." для більш детального налаштування вигляду та рівня заголовків, що включаються. "Ручна таблиця" не рекомендується, оскільки вона не оновлюється автоматично.
- **Оновлення змісту:** Якщо після створення змісту в документ були внесені зміни, що впливають на заголовки (змінено текст заголовка, додано/видалено розділи, змінилася нумерація сторінок через додавання/видалення тексту), зміст необхідно оновити, щоб він відповідав поточному стану документа.
 - Клацнути лівою кнопкою миші на будь-якій частині створеного змісту. Над ним з'явиться кнопка "Оновити таблицю" (Update Table).
 - Або перейти на вкладку "Посилання" -> група "Зміст" -> кнопка "Оновити таблицю".
 - З'явиться діалогове вікно з питанням: "Оновити лише номери сторінок" або "Оновити цілком таблицю".

- "Оновити лише номери сторінок": Використовується, якщо змінилися лише номери сторінок заголовків, але самі заголовки та їхня ієрархія (рівні) залишилися незмінними. Це швидше оновлення.
- "Оновити цілком таблицю": Використовується, якщо змінилися текст заголовків, їхні рівні або додані/видалені нові заголовки. Це оновлення повністю пересканує документ та перебудує зміст.
- **Видалення змісту:** Клацнути на змісті, щоб виділити його (або клацнути на кнопці зі стрілкою біля заголовка "Зміст") та вибрати "Видалити зміст".

ЗМІСТОВИЙ МОДУЛЬ 2

ТАБЛИЧНИЙ ПРОЦЕСОР MS EXCEL ТА СУБД ACCESS

ТЕМА 2.1. ТАБЛИЧНИЙ ПРОЦЕСОР MS EXCEL: ОСНОВИ РОБОТИ

Мета: Ознайомити здобувачів вищої освіти з призначенням та інтерфейсом табличного процесора MS Excel, розкрити структуру електронної книги, аркушів та комірок, навчити базовим навичкам введення, редагування та форматування даних різних типів.

План:

1. Призначення та інтерфейс програми.
2. Структура електронної книги: аркуші, комірки.
3. Введення, редагування та форматування даних.
4. Типи даних в Excel та їх значення.

1. Призначення та інтерфейс програми

Microsoft Excel – це провідна програма для роботи з електронними таблицями, що є частиною пакета Microsoft Office. Вона розроблена для ефективної організації, обробки, аналізу та візуалізації даних, особливо числових. Excel широко використовується в бізнесі, фінансах, науці, освіті та домашньому господарстві для таких завдань, як:

- Ведення бюджетів та фінансовий облік.
- Створення та аналіз звітів.
- Статистичні розрахунки та моделювання.
- Управління списками та базами даних (прості випадки).
- Побудова діаграм та графіків.

- Автоматизація розрахунків за допомогою формул та функцій.

Ключові елементи інтерфейсу MS Excel (схожі з Word, але з урахуванням специфіки таблиць):

- **Стрічка (Ribbon):** Розташована вгорі, містить вкладки ("Файл", "Головна", "Вставлення", "Розмітка сторінки", "Формули", "Дані", "Рецензування", "Вигляд", "Розробник" (за потреби), "Надбудови" (за потреби)) з групами команд, організованими за функціональним призначенням (наприклад, "Буфер обміну", "Шрифт", "Вирівнювання", "Число", "Стилі", "Комірки", "Редагування" на вкладці "Головна"; "Бібліотека функцій", "Визначені імена", "Аудит формул" на вкладці "Формули").
- **Область Backstage (Вкладка "Файл"):** Як і в Word, містить команди для роботи з файлами (створити, відкрити, зберегти, друк, спільний доступ, експорт) та параметрами програми.
- **Панель швидкого доступу (Quick Access Toolbar):** Настроювана панель для швидкого доступу до часто використовуваних команд.
- **Рядок заголовка:** Відображає ім'я поточної книги та назву програми.
- **Поле імені (Name Box):** Знаходиться ліворуч від рядка формул. Відображає адресу активної комірки (наприклад, A1), ім'я діапазону, якщо комірці або діапазону було присвоєно ім'я, або ім'я об'єкта (діаграми, фігури). Дозволяє швидко переходити до комірки або діапазону за його адресою/ім'ям.
- **Рядок формул (Formula Bar):** Розташований під стрічкою, праворуч від поля імені. Відображає вміст активної комірки – введені дані або формулу, якщо комірка її містить. Дозволяє вводити або редагувати дані чи формули безпосередньо. Це особливо зручно для довгих формул. Може бути розширений для відображення кількох рядків формули.
- **Заголовки стовпців та рядків:** Букви (A, B, C...) у верхній частині робочої області позначають стовпці, а числа (1, 2, 3...) ліворуч позначають рядки. Клацання на заголовку виділяє весь стовпець або рядок.
- **Робоча область (Сітка комірок):** Основна частина вікна, що являє собою велику таблицю, утворену перетином стовпців та рядків. Тут розташовуються дані.
- **Ярлички аркушів (Sheet Tabs):** Розташовані внизу вікна, над рядком стану. Кожна вкладка представляє окремий аркуш в поточній книзі. Можна перемикатися між аркушами, додавати нові аркуші (кнопка "+"), видаляти, перейменовувати (подвійне клацання), переміщувати, копіювати та приховувати аркуші (через контекстне меню правою кнопкою миші). Можна також змінювати колір ярличка.
- **Рядок стану (Status Bar):** Внизу вікна. Відображає різноманітну службову інформацію: режим роботи (готовність, введення), результат автообчислень для виділених комірок (середнє, кількість, сума, мінімум, максимум), стан Num

Lock/Caps Lock, елементи керування масштабом та режимами перегляду аркуша.

2. Структура електронної книги: аркуші, комірки

Основний файл, з яким працює Microsoft Excel, називається **книгою (workbook)**. Книга – це контейнер, що складається з одного або кількох **аркушів (worksheets)**.

- **Книга:** За замовчуванням нова книга Excel має ім'я "Книга1" і містить один або кілька аркушів (кількість аркушів за замовчуванням можна налаштувати в параметрах Excel). Книга зберігається як файл з розширенням .xlsx (для версій 2007 і новіших) або .xls (для старіших версій).
- **Аркуш:** Являє собою основний робочий простір для введення та обробки даних. Це величезна двовимірна сітка, утворена перетином **рядків (rows)** та **стовпців (columns)**.
 - **Рядки:** Нумеруються цілими числами, починаючи з 1. У сучасних версіях Excel аркуш може містити понад 1 мільйон рядків.
 - **Стовпці:** Позначаються латинськими літерами (A, B, C, ..., Z, AA, AB, ..., XFD). У сучасних версіях Excel аркуш може містити понад 16 тисяч стовпців.
- **Комірка (Cell):** Є базовим елементом аркуша, що розташований на перетині певного рядка та певного стовпця. Кожна комірка має унікальну **адресу (cell address)**, що складається з літери стовпця та номера рядка (наприклад, A1, B5, AA100). Це аналогічно координатам у шахах.
 - **Активна комірка (Active Cell):** Це комірка, яка наразі виділена (обведена товстою рамкою) і готова для введення або редагування даних. Її адреса відображається в Полі імені.
- **Діапазон комірок (Range):** Прямокутна група суміжних комірок. Діапазон визначається адресами його верхньої лівої та нижньої правої комірок, розділеними двокрапкою (наприклад, A1:C5 – діапазон, що включає комірки A1, B1, C1, A2, B2, C2, ..., A5, B5, C5). Діапазони можуть бути використані у формулах та функціях для посилання на групу даних. Можна виділяти несуміжні діапазони, утримуючи клавішу Ctrl під час виділення.

3. Введення, редагування та форматування даних

Робота з даними в комірках є основним процесом в Excel.

- **Введення даних:**
 - Виберіть комірку, зробивши її активною (клацніть на ній мишею або перейдіть за допомогою клавіш керування курсором).
 - Введіть дані з клавіатури. Під час введення текст відображається як у самій комірці, так і в Рядку формул.
 - Для завершення введення та переходу до наступної комірки використовуйте:

- Enter: Перехід на комірку нижче.
 - Tab: Перехід на комірку праворуч.
 - Стрілки керування курсором: Перехід у відповідному напрямку.
 - Клацання мишею на іншій комірці.
- **Редагування даних:**
 - Якщо потрібно повністю замінити вміст комірки: виберіть її та почніть вводити нові дані.
 - Якщо потрібно змінити частину вмісту комірки:
 - Двічі клацніть лівою кнопкою миші на комірці – курсор з'явиться всередині комірки для редагування.
 - Виберіть комірку та натисніть клавішу F2 – також переводить комірку в режим редагування.
 - Виберіть комірку та відредагуйте вміст у Рядку формул.
 - Використовуйте клавіші Backspace, Delete, стрілки, Home, End для навігації та редагування тексту всередині комірки/рядка формул. Натисніть Enter для завершення редагування.
- **Видалення даних:**
 - Виділіть комірку(и) або діапазон.
 - Натисніть клавішу **Delete** – видаляється лише вміст комірок (дані або формули), але форматування залишається.
 - Використовуйте команду "Очистити" (Clear) на вкладці "Головна" -> група "Редагування". Дозволяє вибрати, що саме очистити: "Очистити все" (вміст і форматування), "Очистити формати", "Очистити вміст", "Очистити примітки" тощо.
- **Копіювання та Переміщення:**
 - Виділіть комірку(и) або діапазон.
 - Копіювання: Ctrl+C або кнопка "Копіювати" на вкладці "Головна".
 - Вирізання (для переміщення): Ctrl+X або кнопка "Вирізати".
 - Встановіть активну комірку в місце призначення.
 - Вставка: Ctrl+V або кнопка "Вставити". Excel також пропонує спеціальні параметри вставки (Paste Special) через меню кнопки "Вставити" (наприклад, вставити тільки значення, тільки формати, тільки формули, транспонувати дані, вставити зв'язок).
- **Автозаповнення (AutoFill):** Дуже корисний інструмент для швидкого копіювання даних, формул або створення послідовностей.
 - Виберіть комірку(и), що містить дані або формулу, яку потрібно скопіювати.
 - Наведіть курсор миші на маленький чорний квадрат у правому нижньому куті рамки виділення активної комірки/діапазону. Курсор зміниться на чорний хрестик (маркер заповнення).
 - Перетягніть маркер заповнення вниз, вгору, вправо або вліво.

- Якщо Excel розпізнає послідовність (наприклад, число 1, 2, 3..., дні тижня Пн, Вт..., місяці Січ, Лют..., дати), він автоматично продовжить її. Для чисел, щоб створити послідовність (наприклад, 10, 20, 30...), потрібно виділити перші два елементи послідовності перед перетягуванням.
- Якщо послідовність не розпізнано, Excel просто скопіює вміст вихідної комірки/діапазону.
- Після автозаповнення поруч з'являється кнопка "Параметри автозаповнення" (Auto Fill Options), що дозволяє вибрати, як саме має бути виконане заповнення (копіювати комірки, заповнити за форматом, заповнити тільки значення, заповнити дні/місяці/роки тощо).
- **Форматування даних (Формат комірок):** Зміна зовнішнього вигляду комірок та даних у них без зміни самого значення. Інструменти форматування знаходяться переважно на вкладці "Головна". Детальні налаштування – у діалоговому вікні "Формат комірок" (викликається правою кнопкою миші на комірці -> "Формат комірок..." або Ctrl+1).
 - **Група "Шрифт":** Налаштування шрифту, розміру, накреслення, кольору тексту, заливки комірки, меж комірки (аналогічно Word).
 - **Група "Вирівнювання":** Горизонтальне (по лівому краю, по центру, по правому краю, по ширині) та вертикальне (вгорі, посередині, внизу) вирівнювання даних у комірці. Орієнтація тексту (вертикально, під кутом). Перенесення тексту по словах (Wrap Text) – вміст комірки відображається в кількох рядках всередині комірки, якщо він не поміщається в один рядок по ширині стовпця. Об'єднання комірок (Merge & Center) – об'єднує кілька виділених комірок в одну велику, зазвичай центрує вміст.
 - **Група "Число":** Дуже важлива група для роботи з числовими даними. Дозволяє вибрати, як числові, грошові значення, дати, час тощо будуть відображатися в комірці.
 - **Числові формати:** Загальний (General), Числовий (Number - з роздільником тисяч, кількістю десяткових знаків), Грошовий (Currency - зі символом валюти), Фінансовий (Accounting - символ валюти вирівнюється по лівому краю), Відсотковий (Percentage), Дробовий (Fraction), Експоненційний (Scientific), Текстовий (Text), Дата (Date), Час (Time).
 - Кнопки для швидкого застосування грошового, відсоткового форматів, формату з роздільником тисяч.
 - Кнопки для збільшення/зменшення кількості десяткових знаків.

- **Група "Стилі":** Застосування готових стилів комірок (Cell Styles) для одноманітного форматування. Умове форматування (Conditional Formatting) – див. Лекцію 13.
- **Група "Комірки":** Вставка/видалення комірок, рядків, стовпців, аркушів. Формат (Format) – налаштування висоти рядка, ширини стовпця, приховування/відображення рядків/стовпців/аркушів, захист аркуша.

4. Типи даних в Excel та їх значення

Excel розпізнає та обробляє дані різних типів. Правильне розпізнавання типу даних є фундаментальним для коректної роботи формул, функцій, сортування, фільтрації та інструментів аналізу. Excel намагається автоматично визначити тип даних при введенні, але іноді може потребувати втручання користувача.

Число (Number): Включає цілі числа (123), десяткові дроби (123.45), від'ємні числа (-10), відсотки (15%), дроби (1/2), експоненційну форму запису (1.2E+5). Excel зберігає числові дані у внутрішньому числовому форматі, незалежно від того, як вони відображаються завдяки числовому формату комірки. Числа за замовчуванням вирівнюються по правому краю комірки. Можуть використовуватися в математичних обчисленнях.

Текст (Text): Будь-яка послідовність символів (літери, цифри, пробіли, знаки пунктуації), яка не може бути інтерпретована як число, дата, час, логічне значення чи формула. Excel розглядає як текст навіть числа, якщо вони введені з апострофом на початку ('123) або відформатовані як текстовий тип. Текст за замовчуванням вирівнюється по лівому краю. Текстові дані не можуть безпосередньо використовуватися в математичних операціях (хоча існують текстові функції для роботи з ними).

Дата/Час (Date/Time): Excel зберігає дати та час як спеціальний числовий формат. Дата зберігається як послідовне число, що представляє кількість днів, що минули з 1 січня 1900 року (наприклад, 01.01.1900 = 1, 01.01.2025 = 45650). Час зберігається як дробова частина доби (наприклад, 12:00 = 0.5). Така система дозволяє легко виконувати арифметичні операції з датами (наприклад, віднімати дати для отримання кількості днів) та часом. Відображення дати/часу залежить від числового формату комірки (наприклад, "dd.mm.yyyy", "h:mm AM/PM"). Дати/час за замовчуванням вирівнюються по правому краю.

Логічні значення (Boolean): TRUE (Істина) або FALSE (Хибність). Є результатом роботи логічних функцій (IF, AND, OR) та операторів порівняння (=, <, >, <=, >=). В Excel логічне значення TRUE внутрішньо представлене як 1, а FALSE – як 0, що дозволяє використовувати їх в деяких математичних операціях (хоча це не завжди рекомендується для ясності).

Формула (Formula): Вираз, що завжди починається зі знака рівності (=). Виконує обчислення, використовуючи значення, посилання на комірки, оператори та функції. У комірці відображається результат обчислення формули, а сама

формула – у Рядку формул. Якщо Excel не може обчислити формулу, він повертає значення помилки.

Помилки (Error Values): Спеціальні значення, що з'являються в комірці, коли формула не може бути обчислена коректно. Вказують на проблему, яку потрібно виправити. Приклади:

- #DIV/0! (Ділення на нуль)
- #NAME? (Excel не розпізнає ім'я у формулі - наприклад, ім'я функції написано з помилкою)
- #VALUE! (Неправильний тип аргументу або оператора)
- #REF! (Недійсне посилання на комірку)
- #N/A (Значення недоступне - часто результат функцій пошуку, коли шукане значення не знайдено)
- #NUM! (Проблема з числовим значенням)
- #NULL! (Перетин двох областей, які не перетинаються)

Важливість правильного типу даних: Неправильне визначення типу даних є частою причиною помилок в Excel. Наприклад, якщо числа випадково введені або відформатовані як текст, функції SUM, AVERAGE можуть їх ігнорувати, а сортування буде виконуватися за алфавітним принципом, а не за числовим значенням. Дати, введені у форматі, який Excel не може розпізнати як дату, будуть сприйматися як текст, і ви не зможете виконувати з ними календарні розрахунки. Тому перед виконанням обчислень або аналізу даних завжди перевіряйте, що дані мають правильний тип, і за потреби використовуйте інструменти форматування числа або функції для перетворення типів (наприклад, VALUE для перетворення тексту, що виглядає як число, у число).

ТЕМА 2.2: ТАБЛИЧНИЙ ПРОЦЕСОР MS EXCEL: ФОРМУЛИ ТА ФУНКЦІЇ

Мета: Детально розглянути принципи створення та використання формул для обчислень в Excel, вивчити різні типи посилань на комірки (відносні, абсолютні, мішані) та їхнє значення при копіюванні формул, а також ознайомитись з основними вбудованими функціями Excel для автоматизації розрахунків.

План:

1. Використання формул для обчислень.
2. Абсолютні, відносні та мішані посилання.
3. Вбудовані функції (огляд за категоріями).

1. Використання формул для обчислень

Формули є серцем Excel, дозволяючи виконувати різноманітні обчислення, від простих арифметичних операцій до складних аналітичних розрахунків. Будь-яка формула в Excel завжди починається зі знака рівності (=). Цей знак вказує Excel, що вміст комірки є не просто даними, а інструкцією для обчислення.

- **Компоненти формули:** Формула може складатися з різних елементів:

- **Константи:** Фіксовані числові або текстові значення, які вводяться безпосередньо у формулу. Наприклад, =A1*1.2 (де 1.2 – константа).
- **Посилання на комірки (Cell References):** Адреси інших комірок, значення яких використовуються в обчисленнях. При зміні значення в комірці, на яку є посилання, результат формули автоматично оновлюється. Наприклад, =A1+B1 (використовуються значення з A1 та B1).
- **Оператори (Operators):** Символи, що визначають тип операції, яку потрібно виконати над значеннями або посиланнями.

- **Арифметичні оператори:** Виконують математичні обчислення.

- ✓ + (Додавання)
 - ✓ - (Віднімання)
 - ✓ * (Множення)
 - ✓ / (Ділення)
 - ✓ ^ (Піднесення до степеня)
 - ✓ % (Відсоток – ділення числа на 100)

- **Оператори порівняння (Relation Operators):** Порівнюють два значення і повертають логічне значення TRUE або FALSE.

- ✓ = (Дорівнює)
 - ✓ <> (Не дорівнює)
 - ✓ < (Менше)
 - ✓ > (Більше)
 - ✓ <= (Менше або дорівнює)
 - ✓ >= (Більше або дорівнює)

- **Текстовий оператор:**

- ✓ & (Амперсанд, Конкатенація): Об'єднує (з'єднує) кілька текстових рядків в один. Наприклад, =A1 & " " & B1 (об'єднає вміст A1, пробіл та вміст B1).

- **Оператори посилань:** Використовуються для об'єднання діапазонів для обчислень (пробіл – перетин, кома – об'єднання).

- **Функції (Functions):** Вбудовані або користувацькі іменовані формули, що виконують певні обчислення. Спрощують виконання складних або типових операцій (див. детальніше в п. 3). Наприклад, `=SUM(A1:A10)` (функція SUM для підсумовування діапазону).
- **Дужки ():** Використовуються для зміни стандартного порядку обчислень та для визначення аргументів функцій.
- **Порядок обчислень (Operator Precedence):** Excel обчислює компоненти формули у певному порядку за замовчуванням, слідуючи стандартним математичним правилам:
 - Вирази в круглих дужках.
 - Оператори діапазонів, перетину, об'єднання.
 - Оператор заперечення (- як знак мінус, не як віднімання).
 - Відсотки (%).
 - Піднесення до степеня (^).
 - Множення (*) та Ділення (/) – виконуються зліва направо.
 - Додавання (+) та Віднімання (-) – виконуються зліва направо.
 - Текстовий оператор конкатенації (&).
 - Оператори порівняння (=, <, >, <=, >=) – виконуються зліва направо.

Для зміни цього порядку використовуйте дужки. Наприклад, у формулі `=A1+B1*C1` спочатку буде виконано множення `B1*C1`, а потім додавання `A1`. У формулі `=(A1+B1)*C1` спочатку буде обчислена сума `A1+B1` (бо вона в дужках), а потім результат буде помножено на `C1`.

- **Введення та копіювання формул:**
 - Формули вводяться в активну комірку або рядок формул, починаючи зі знака `=`.
 - Після введення натисніть Enter, щоб побачити результат обчислення в комірці. Сама формула залишається видимою у Рядку формул при виділенні цієї комірки.
 - При копіюванні формул (за допомогою маркера заповнення або `Copy/Paste`) Excel автоматично коригує посилання на комірки залежно від їх типу (див. п. 2). Це одна з найпотужніших можливостей Excel, що дозволяє швидко застосувати одну й ту саму логіку розрахунків до великих обсягів даних.

2. Абсолютні, відносні та мішані посилання

Розуміння та правильне використання різних типів посилань на комірки є критично важливим для створення ефективних та гнучких формул, особливо при їх копіюванні.

- **Відносні посилання (Relative References):** (наприклад, A1, B5, C10)
 - Це стандартний тип посилання в Excel за замовчуванням.
 - При копіюванні формули, що містить відносні посилання, в іншу комірку, ці посилання автоматично змінюються **відносно** нового положення комірки, куди скопійовано формулу.
 - **Приклад:** Якщо в комірці C1 є формула $=A1+B1$, і ви скопіюєте її в комірку C2, Excel автоматично змінить посилання так, щоб вони вказували на комірки в тому ж *відносному* положенні, що й в оригіналі. Тобто, формула в C2 стане $=A2+B2$. При копіюванні в D1 вона стане $=B1+C1$.
 - Це дуже зручно для виконання однотипних обчислень для кожного рядка або стовпця таблиці (наприклад, розрахунок суми або прибутку для кожного товару в списку).
- **Абсолютні посилання (Absolute References):** (наприклад, \$A\$1, \$C\$5)
 - Посилання, що залишається **незмінним** (фіксованим) при копіюванні формули в будь-яке інше місце.
 - Перед літерою стовпця та перед номером рядка ставиться знак долара (\$). Знак долара "закріплює" відповідну частину посилання. \$A\$1 означає, що і стовпець A, і рядок 1 є фіксованими.
 - **Приклад:** Якщо у формулі $=A1 + \$B\1 з комірки C1 ви скопіюєте її в C2, формула стане $=A2 + \$B\1 . Посилання на B1 залишилося абсолютним, а посилання на A1 – відносним і змінилося на A2.
 - Абсолютні посилання використовуються, коли формула завжди повинна посилатися на одну конкретну комірку, незалежно від того, куди її копіюють. Наприклад, якщо комірка містить податкову ставку, курс валют, константу (наприклад, PI) або комірку з підсумком, на яку потрібно посилатися з багатьох інших формул.
- **Мішані посилання (Mixed References):** (наприклад, \$A1, B\$1)
 - У мішаному посиланні фіксується (робиться абсолютним) лише одна частина посилання – або стовпець, або рядок, а інша частина залишається відносною.
 - \$A1: Фіксований стовпець A, відносний рядок 1. При копіюванні формули вниз або вгору змінюватиметься лише номер рядка, стовпець A залишатиметься незмінним. При копіюванні формули вліво або вправо стовпець A залишиться незмінним, рядок 1 також не зміниться.
 - A\$1: Відносний стовпець A, фіксований рядок 1. При копіюванні формули вліво або вправо змінюватиметься лише літера стовпця, рядок 1 залишатиметься незмінним. При копіюванні вниз або вгору рядок 1 залишиться незмінним, стовпець A також не зміниться.
 - **Приклад:** Мішані посилання корисні при створенні таблиць з формулами, де потрібно посилатися на заголовки рядків та стовпців одночасно. Наприклад, для створення таблиці множення, де кожна

комірка є результатом множення відповідного заголовка стовпця на відповідний заголовок рядка. Формула виглядатиме приблизно так: $=\$A2 * B\1 (якщо заголовки рядків у стовпці A, а заголовки стовпців у рядку 1).

- **Перемикання типів посилань:** При редагуванні формули (перебуваючи в комірці або Рядку формул) можна швидко перемикатися між відносним, абсолютним та двома варіантами мішаних посилань для вибраної адреси комірки, натискаючи функціональну клавішу **F4**. Кожне натискання F4 циклічно змінює тип посилання (наприклад, A1 -> \$A\$1 -> A\$1 -> \$A1 -> A1).

3. Вбудовані функції (огляд за категоріями)

Excel містить величезну бібліотеку попередньо визначених (вбудованих) функцій, що покривають широкий спектр обчислень – від простих математичних до складних фінансових, статистичних, інженерних тощо. Використання функцій значно спрощує написання формул та автоматизує типові завдання.

- **Синтаксис функції:** Кожна функція має певне ім'я (наприклад, SUM, AVERAGE, IF). Після імені функції йдуть круглі дужки (), в яких містяться **аргументи** – значення або посилання на комірки/діапазони, які функція використовує для своїх обчислень. Аргументи функції в дужках розділяються **крапкою з комою (;)** в українській (та багатьох інших європейських) локалізації Excel або **комою (,)** в англійській локалізації.
 - Приклад: $=SUM(A1:A10; C5)$ – функція SUM з двома аргументами: діапазоном A1:A10 та окремою коміркою C5.
- **Введення функції:**
 - Ввести знак = та ім'я функції (Excel пропонує автозавершення імен функцій).
 - Відкрити дужку (.
 - Ввести аргументи, розділяючи їх крапкою з комою. Аргументи можуть бути константами, посиланнями на комірки/діапазони, іншими формулами або вкладеними функціями.
 - Закрити дужку).
 - Натиснути Enter.
 - Використовувати кнопку "Вставити функцію" (Insert Function) поруч з Рядком формул (fx) або на вкладці "Формули" -> "Бібліотека функцій". Це відкриває діалогове вікно "Аргументи функції", яке допомагає ввести аргументи та дає опис функції.
- **Бібліотека функцій:** На вкладці "Формули" функції згруповані за категоріями ("Фінансові", "Логічні", "Текстові", "Дата й час", "Підстановка й посилання", "Математичні й тригонометричні" та інші), що полегшує пошук потрібної функції.

Огляд основних категорій та прикладів функцій:

- **Математичні та тригонометричні (Math & Trig):** Для виконання математичних операцій.
 - SUM(число1; [число2]; ...): Підсумовує всі числа в діапазоні або списку аргументів.
 - AVERAGE(число1; [число2]; ...): Обчислює середнє арифметичне значень.
 - ROUND(число; кількість_розрядів): Округлює число до вказаної кількості десяткових розрядів.
 - SUMIF(діапазон; критерій; [діапазон_підсумовування]): Підсумовує комірки з діапазону, що відповідають заданому критерію. Дуже корисна для підрахунку підсумків за певних умов.
 - SUMIFS(діапазон_підсумовування; діапазон_умови1; умова1; [діапазон_умови2; умова2]; ...): Підсумовує комірки на основі *кількох* критеріїв у різних діапазонах.
 - COUNTIF(діапазон; критерій): Підраховує кількість непустих комірок у діапазоні, що відповідають заданому критерію.
 - COUNTIFS(діапазон_умови1; умова1; [діапазон_умови2; умова2]; ...): Підраховує кількість комірок на основі *кількох* критеріїв.
 - SQRT(число): Повертає квадратний корінь числа.
 - RAND(): Повертає випадкове число від 0 до 1.
 - RANDBETWEEN(нижнє_число; верхнє_число): Повертає випадкове ціле число в заданому діапазоні.
- **Статистичні (Statistical):** Для статистичного аналізу даних.
 - MAX(число1; [число2]; ...): Повертає найбільше значення зі списку аргументів.
 - MIN(число1; [число2]; ...): Повертає найменше значення зі списку аргументів.
 - COUNT(значення1; [значення2]; ...): Підраховує кількість комірок у діапазоні, що містять *числа*.
 - COUNTA(значення1; [значення2]; ...): Підраховує кількість *непустих* комірок у діапазоні.
 - COUNTBLANK(діапазон): Підраховує кількість пустих комірок у діапазоні.
 - MEDIAN(число1; [число2]; ...): Повертає медіану (середнє значення) з набору даних.
 - MODE.SNGL(число1; [число2]; ...): Повертає моду (значення, що найчастіше зустрічається) з набору даних.
- **Логічні (Logical):** Для роботи з логічними значеннями та умовами.
 - IF(логічний вираз; значення_якщо_істина; значення_якщо_хибність): Найпоширеніша логічна функція. Перевіряє умову (логічний вираз) і повертає одне значення, якщо умова істинна (TRUE), та інше значення,

якщо умова хибна (FALSE). Приклад: =IF(B2>100; "План виконано"; "Потрібно більше").

- AND(логічне_значення1; [логічне_значення2]; ...): Повертає TRUE, якщо *всі* логічні аргументи є TRUE.
- OR(логічне_значення1; [логічне_значення2]; ...): Повертає TRUE, якщо *хоча б один* логічний аргумент є TRUE.
- NOT(логічне_значення): Змінює логічне значення на протилежне (TRUE стає FALSE, FALSE стає TRUE).
- IFS(логічний_вираз1; значення_якщо_істина1; [логічний_вираз2; значення_якщо_істина2]; ...): (Доступна в новіших версіях) Перевіряє кілька умов і повертає значення для першої умови, яка є TRUE. Спрощує вкладені функції IF.
- **Текстові (Text):** Для обробки текстових рядків.
 - CONCATENATE(текст1; [текст2]; ...) або оператор &: Об'єднує (конкатенує) кілька текстових рядків в один. CONCAT() та TEXTJOIN() – новіші, більш гнучкі функції.
 - LEFT(текст; [кількість_знаків]): Повертає вказану кількість символів з початку текстового рядка.
 - RIGHT(текст; [кількість_знаків]): Повертає вказану кількість символів з кінця текстового рядка.
 - MID(текст; початкова_позиція; кількість_знаків): Повертає вказану кількість символів з текстового рядка, починаючи з заданої позиції.
 - LEN(текст): Повертає довжину текстового рядка (кількість символів).
 - TRIM(текст): Видаляє зайві пробіли з тексту (залишає лише одинарні пробіли між словами).
 - LOWER(текст), UPPER(текст), PROPER(текст): Перетворюють регістр тексту (до нижнього, до верхнього, перша літера кожного слова – велика).
 - FIND(шуканий_текст; в_тексті; [початкова_позиція]), SEARCH(...): Знаходять позицію одного текстового рядка в іншому.
 - SUBSTITUTE(текст; старий_текст; новий_текст; [номер_входження]), REPLACE(...): Замінюють частину текстового рядка.
 - TEXT(значення; формат_тексту): Перетворює числове значення (або дату/час) на текст з вказаним форматуванням. Дуже корисна для форматування чисел/дат у тексті.
- **Дати та часу (Date & Time):** Для роботи з датами та часом.
 - TODAY(): Повертає поточну дату.
 - NOW(): Повертає поточну дату та час.
 - DATE(рік; місяць; день): Створює дату за заданими компонентами року, місяця та дня.
 - YEAR(дата), MONTH(дата), DAY(дата): Витягують рік, місяць або день з дати.

- HOUR(час), MINUTE(час), SECOND(час): Витягують годину, хвилину або секунду з часу.
- DATEDIF(початкова_дата; кінцева_дата; одиниця): Обчислює різницю між двома датами в заданих одиницях (наприклад, "Y" для років, "M" для місяців, "D" для днів). Корисна для обчислення віку, стажу.
- WEEKDAY(дата; [тип]): Повертає день тижня для дати (числом).
- WORKDAY(початкова_дата; кількість_днів; [вихідні]): Повертає дату до або після заданої кількості робочих днів, виключаючи вихідні та додаткові святкові дні.
- NETWORKDAYS(початкова_дата; кінцева_дата; [вихідні]): Обчислює кількість робочих днів між двома датами.
- **Пошуку та посилання (Lookup & Reference):** Для пошуку значень у діапазонах або отримання посилань.
 - VLOOKUP(шукане_значення; таблиця_масив; номер_стовпця_індексу; [інтервальний_перегляд]): Шукає значення в першому стовпці діапазону (таблиці_масиву) і повертає значення з тієї ж стрічки вказаного стовпця. Дуже часто використовується для пошуку даних у довідкових таблицях. інтервальний_перегляд (останній аргумент) вказує, чи шукати точну відповідність (FALSE або 0) чи приблизну (TRUE або 1, вимагає сортування першого стовпця).
 - HLOOKUP(...): Аналогічно VLOOKUP, але шукає значення в першому *рядку* таблиці і повертає значення з того ж стовпця вказаного рядка.
 - INDEX(масив; номер_рядка; [номер_стовпця]): Повертає значення комірки на перетині вказаного рядка та стовпця в заданому масиві (діапазоні).
 - MATCH(шукане_значення; масив_пошуку; [тип_зіставлення]): Шукає значення в діапазоні (масиві_пошуку) і повертає *відносну позицію* (порядковий номер) цього елемента в діапазоні.
 - INDEX у поєднанні з MATCH є більш гнучкою альтернативою VLOOKUP, особливо коли шуканий стовпець не є першим у таблиці.
 - OFFSET(посилання; зміщення_рядків; зміщення_стовпців; [висота]; [ширина]): Повертає посилання на діапазон, зміщений на задану кількість рядків та стовпців від початкового посилання.
 - INDIRECT(текст_посилання; [a1]): Перетворює текстовий рядок на дійсне посилання на комірку або діапазон.

Ефективне використання формул та функцій, а також розуміння типів посилань, є основою для створення потужних, динамічних та точних електронних таблиць в Excel. Вміння комбінувати різні функції та оператори дозволяє вирішувати складні аналітичні та розрахункові завдання.

ТЕМА 2.3. ТАБЛИЧНИЙ ПРОЦЕСОР MS EXCEL: АНАЛІЗ ТА ВІЗУАЛІЗАЦІЯ ДАНИХ

Мета: Ознайомити здобувачів вищої освіти з основними інструментами Excel для аналізу та візуалізації даних: сортуванням, фільтрацією, умовним форматуванням, побудовою діаграм та графіків, а також основами роботи зі зведеними таблицями для узагальнення великих обсягів даних.

План:

1. Сортування та фільтрація даних.
2. Умовне форматування.
3. Побудова діаграм та графіків.
4. Зведені таблиці (pivottables) та зведені діаграми (pivotcharts).

Після того, як дані введені, відредаговані та базові розрахунки за допомогою формул виконані, наступним етапом часто є аналіз та візуалізація цих даних. Excel надає потужні інструменти, що дозволяють швидко знаходити закономірності, виділяти важливу інформацію та наочно її представляти. Ефективність цих інструментів значною мірою залежить від того, наскільки якісно та правильно структуровані вихідні дані.

1. Сортування та фільтрація даних

Сортування та фільтрація – це базові, але дуже корисні інструменти для роботи з даними у табличному форматі. Вони дозволяють змінювати порядок відображення рядків та приховувати дані, які не відповідають певним критеріям. Ці інструменти знаходяться на вкладці "Дані" (Data) в групі "Сортування й фільтр".

- **Сортування (Sort):** Дозволяє впорядкувати рядки таблиці (списку даних) на основі значень в одному або кількох стовпцях. При сортуванні Excel автоматично переміщує весь рядок даних, зберігаючи цілісність записів.
 - **Просте сортування:** Сортування за одним стовпцем за зростанням (Ascending, наприклад, А-Я, від найменшого до найбільшого числа, від найстарішої дати до найновішої) або за спаданням (Descending, наприклад, Я-А, від найбільшого до найменшого, від найновішої дати до найстарішої). Використовуються кнопки "Сортування від А до Я" або "Сортування від Я до А" на вкладці "Дані" або на вкладці "Головна" в групі "Редагування" -> "Сортування й фільтр". Перед простим сортуванням достатньо виділити будь-яку комірку в стовпці, за яким потрібно сортувати.
 - **Багаторівневе сортування (Custom Sort):** Дозволяє вказати кілька рівнів сортування. Excel спочатку сортує за першим вказаним стовпцем, потім, для записів з однаковим значенням у першому стовпці,

сортують за другим вказаним стовпцем і так далі. Це налаштовується через діалогове вікно "Сортування", яке відкривається кнопкою "Сортування" на вкладці "Дані". У цьому вікні можна додавати, видаляти та змінювати порядок рівнів сортування. Також можна сортувати не тільки за значеннями комірок, але й за кольором комірки, кольором шрифту або піктограмою умовного форматування.

- **Важливо:** Перед сортуванням необхідно переконатися, що виділено весь діапазон даних, який потрібно сортувати, включаючи всі стовпці, що належать до одного набору записів. Якщо виділити лише частину даних, сортування призведе до переміщення лише цих даних, порушуючи зв'язок між значеннями в одному рядку. Якщо у ваших даних є рядок заголовків, встановіть прапорець "Мої дані мають заголовки" (My data has headers) у діалоговому вікні сортування, щоб перший рядок не брав участі в сортуванні.
- **Фільтрація (Filter):** Дозволяє тимчасово приховати рядки, які не відповідають заданим критеріям, залишаючи видимими лише ті дані, які вам потрібні. Це не видаляє дані, а лише приховує їх. Excel відображає номери прихованих рядків синім кольором.
 - **Автофільтр (AutoFilter):** Найпростіший і найпоширеніший спосіб фільтрації. Активується кнопкою "Фільтр" на вкладці "Дані" або на вкладці "Головна" -> "Сортування й фільтр". У заголовках кожного стовпця даних з'являються кнопки зі стрілками (фільтри). Кладання на такій кнопці відкриває меню фільтрації для цього стовпця.
 - **Опції фільтрації в Автофільтрі:**
 - Вибір зі списку значень: Можна вибрати конкретні унікальні значення зі стовпця, які потрібно відобразити.
 - Текстові фільтри: Умови для текстових даних (наприклад, "дорівнює", "не дорівнює", "починається з", "містить").
 - Числові фільтри: Умови для числових даних (наприклад, "дорівнює", "більше", "менше або дорівнює", "між", "перші 10...").
 - Фільтри за датами: Умови для дат (наприклад, "дорівнює", "до", "після", "завтра", "цей місяць", "минулий рік").
 - Фільтрація за кольором: Якщо в комірках застосовано форматування (заливка комірки, колір шрифту) або умовне форматування (піктограми), можна фільтрувати за цими атрибутами.
 - **Очистити фільтр (Clear Filter):** Після застосування фільтра в заголовку стовпця з'являється значок фільтра. Щоб скасувати фільтрацію для цього стовпця, клацніть на його кнопці фільтра та виберіть "Очистити фільтр з ...". Кнопка "Очистити" на вкладці "Дані" скасовує фільтрацію з усіх стовпців.

- **Розширений фільтр (Advanced Filter):** Використовується для більш складних умов фільтрації, які неможливо реалізувати за допомогою Автофільтра (наприклад, умови OR між різними стовпцями) або для копіювання відфільтрованих даних в інше місце на аркуші або на інший аркуш. Вимагає створення окремого "діапазону умов" на аркуші.

2. Умове форматування (Conditional Formatting)

Умове форматування – це потужний інструмент для візуального аналізу даних. Він дозволяє автоматично змінювати вигляд комірок (колір заливки, колір шрифту, стиль накреслення, додавання гістограм/півтограм) залежно від їхнього вмісту або виконання певних правил (умов). Це допомагає швидко виділити важливі значення, виявити тенденції, викиди або закономірності в даних без необхідності переглядати кожну комірку вручну. Інструменти умовного форматування знаходяться на вкладці "Головна" в групі "Стили".

- **Принцип роботи:** Ви виділяєте діапазон комірок і створюєте для нього одне або кілька правил форматування. Кожне правило складається з умови та формату, який має бути застосований, якщо умова виконується. Excel постійно перевіряє ці умови, і якщо вміст комірки змінюється так, що умова стає істинною, форматування автоматично застосовується.
- **Типи правил умовного форматування:** Excel пропонує готові шаблони правил, згруповані за категоріями:
 - **Правила виділення клітинок (Highlight Cells Rules):** Форматування комірок, значення яких:
 - Більше/Менше певного значення (Greater Than / Less Than)
 - Між двома значеннями (Between)
 - Дорівнює певному значенню (Equal To)
 - Містить певний текст (Text that Contains)
 - Відповідає певній даті (A Date Occurring)
 - Є повторюваними/унікальними значеннями (Duplicate / Unique Values)
 - **Правила для перших/останніх значень (Top/Bottom Rules):** Виділення:
 - Перших N елементів (Top 10 Items)
 - Останніх N елементів (Bottom 10 Items)
 - Перших N відсотків (Top 10% Items)
 - Останніх N відсотків (Bottom 10% Items)
 - Значень вище/нижче середнього (Above / Below Average)
 - **Гістограми (Data Bars):** Додають кольорові смуги всередину комірок. Довжина смуги пропорційна значенню комірки відносно інших значень у виділеному діапазоні. Візуально показують співвідношення значень.
 - **Колірні шкали (Color Scales):** Застосовують градієнтну заливку до діапазону комірок. Колір комірки змінюється поступово залежно від її

значення відносно інших значень у діапазоні. Наприклад, від зеленого (найвищі значення) до червоного (найнижчі значення). Допомагають швидко оцінити розподіл даних.

- **Набори піктограм (Icon Sets):** Додають невеликі значки (стрілки, світлофори, прапорці, зірочки, рейтинги) до комірок. Піктограми класифікують значення на категорії (наприклад, за допомогою 3 або 5 значків) залежно від їхнього положення в діапазоні значень.
- **Створення власних правил (New Rule):** Дозволяє задавати більш складні умови форматування, зокрема, використовуючи формули. Наприклад, можна форматувати рядок, якщо значення в одній з його комірок відповідає певній умові.
- **Керування правилами (Manage Rules):** Відкриває діалогове вікно "Диспетчер правил умовного форматування", де можна переглядати, редагувати, видаляти, дублювати та змінювати порядок застосування існуючих правил для вибраного діапазону, поточного аркуша або всієї книги. Порядок правил має значення, оскільки форматування застосовується у тому порядку, в якому правила перелічені.

3. Побудова діаграм та графіків

Діаграми (Charts) та графіки – це ефективний спосіб візуалізації числових даних, що дозволяє зробити їх більш зрозумілими та наочними. Вони допомагають швидко побачити тенденції, порівняти значення, показати розподіл або частку чогось у цілому.

● Процес створення:

- **Підготовка даних:** Переконайтеся, що дані для діаграми розташовані у зручному форматі (як правило, у стовпцях або рядках, з відповідними заголовками). Заголовки стовпців/рядків будуть використовуватися як назви категорій або елементів легенди.
 - **Виділення даних:** Виділіть діапазон комірок, що містить дані, які потрібно відобразити на діаграмі. Зазвичай включають заголовки стовпців/рядків.
 - **Вставка діаграми:** Перейдіть на вкладку "Вставлення" (Insert) -> група "Діаграми" (Charts).
 - **Вибір типу діаграми:** Виберіть рекомендований тип діаграми ("Рекомендовані діаграми") або виберіть потрібний тип зі списку доступних ("Гістограма", "Лінійчата", "Кругова" тощо). Excel покаже попередній перегляд діаграми на основі ваших даних.
- **Основні типи діаграм та їх призначення:**
 - **Гістограма (Column Chart) / Стовпчаста діаграма (Bar Chart):** Використовується для порівняння значень між різними категоріями. Гістограма використовує вертикальні стовпці, стовпчаста – горизонтальні.

- **Лінійна діаграма (Line Chart) / Графік:** Ідеально підходить для відображення **тенденцій** та змін даних у часі або за іншою безперервною шкалою. Точки даних з'єднуються лініями.
- **Кругова діаграма (Pie Chart) / Секторна діаграма:** Показує **частку** кожної категорії у загальному підсумку (як правило, у відсотках). Ефективна, коли потрібно порівняти кілька частин одного цілого. Не рекомендується використовувати для великої кількості категорій.
- **Точкова діаграма (Scatter Chart):** Використовується для візуалізації **залежності** між двома числовими змінними, пошуку кореляцій. Відображає кожну пару значень як точку на площині.
- **Діаграма з областями (Area Chart):** Схожа на лінійну діаграму, але область під лінією зафарбована. Використовується для відображення тенденцій та **обсягів** змін у часі.
- **Комбінована діаграма (Combo Chart):** Дозволяє об'єднати різні типи діаграм на одному графіку (наприклад, гістограма для обсягів продажів та лінійний графік для прибутку), часто з використанням вторинної осі Y.
- **Інші типи:** Пелюсткова (для порівняння кількох об'єктів за багатьма параметрами), Біржова (для відображення даних акцій), Бульбашкова (для відображення трьох змінних), Поверхнева (для відображення 3D-даних) тощо.
- **Налаштування діаграми (Chart Tools):** Після створення діаграми вона стає активним об'єктом, і в стрічці з'являються дві контекстні вкладки: "Конструктор діаграм" (Chart Design) та "Формат" (Format).
 - **Вкладка "Конструктор діаграм":** Для зміни даних, типу та загального вигляду діаграми.
 - Додати елемент діаграми (Add Chart Element): Додавання/редагування назви діаграми, назв осей, легенди, підписів даних, таблиці даних, ліній сітки, смуг похибок, лінії тренду.
 - Швидкий макет (Quick Layout): Застосування готових наборів елементів діаграми.
 - Стили діаграм (Chart Styles): Застосування готових візуальних стилів (кольорові схеми, ефекти).
 - Переключити рядок/стовпець (Switch Row/Column): Зміна способу відображення даних – рядки як категорії, стовпці як ряди даних, або навпаки.
 - Вибрати дані (Select Data): Редагування діапазону даних, що використовується для діаграми, додавання/видалення рядів даних, зміна підписів горизонтальної осі.
 - Змінити тип діаграми (Change Chart Type): Зміна типу діаграми після її створення.

- Перемістити діаграму (Move Chart): Переміщення діаграми на інший аркуш (як окремий аркуш діаграми) або як об'єкт на існуючий аркуш.
- **Вкладка "Формат":** Для детального форматування окремих елементів діаграми (заливка, контур, ефекти фігур, форматування тексту).

Вибір правильного типу діаграми: Вибір відповідного типу діаграми є критично важливим. Неправильно обрана діаграма може спотворити дані або ускладнити їх розуміння. Завжди думайте про те, яке повідомлення ви хочете донести своїми даними і який тип діаграми найкраще це візуалізує.

4. Зведені таблиці (PivotTables) та зведені діаграми (PivotCharts)

Зведені таблиці – це один з найпотужніших інструментів Excel для аналізу великих обсягів даних. Вони дозволяють швидко узагальнювати, групувати та представляти дані у вигляді інтерактивних звітів без написання складних формул.

- **Призначення:** Зведені таблиці дозволяють:
 - Групувати дані за різними категоріями (наприклад, підрахувати загальний дохід за кожним продуктом або регіоном).
 - Виконувати агрегуючі обчислення (сума, кількість, середнє, максимум, мінімум, стандартне відхилення тощо) над даними в групах.
 - Фільтрувати та сортувати узагальнені дані.
 - Легко змінювати структуру аналітичного звіту шляхом перетягування полів.
 - Швидко отримувати відповіді на запитання типу "Скільки було продано товару X в регіоні Y за період Z?".
- **Передумова:** Для створення зведеної таблиці дані повинні бути добре структуровані – представлені у вигляді плаского списку або таблиці, де кожен стовпець має унікальний заголовок, і кожен рядок є окремим записом або транзакцією. Не повинно бути пустих рядків або повністю пустих стовпців всередині діапазону даних.
- **Створення зведеної таблиці:**
 - Виділіть будь-яку комірку в діапазоні даних, який потрібно проаналізувати.
 - Перейдіть на вкладку "Вставлення" (Insert) -> група "Таблиці" (Tables) -> "Зведена таблиця" (PivotTable).
 - У діалоговому вікні "Створення зведеної таблиці":
 - Переконайтеся, що діапазон даних визначено правильно.
 - Виберіть місце, куди потрібно помістити зведену таблицю (новий аркуш – рекомендовано, або існуючий аркуш).
 - Натисніть "ОК".
- **Робота з полями зведеної таблиці:** Після створення зведеної таблиці на аркуші з'являється порожній макет, а праворуч – **Область полів зведеної таблиці**

(PivotTable Fields). Ця область містить список усіх заголовків стовпців з ваших вихідних даних як "Поля". Нижче розташовані чотири області макета:

- **Фільтри (Filters):** Перетягніть сюди поля, за якими потрібно мати можливість фільтрувати дані у всій зведеній таблиці.
- **Стовпці (Columns):** Перетягніть сюди поля, унікальні значення яких стануть заголовками стовпців у зведеній таблиці.
- **Рядки (Rows):** Перетягніть сюди поля, унікальні значення яких стануть заголовками рядків у зведеній таблиці.
- **Значення (Values):** Перетягніть сюди поля, за якими потрібно виконувати обчислення (агрегацію). Excel автоматично обирає стандартну функцію агрегації (наприклад, Сума для числових даних, Кількість для текстових), але її можна змінити (клацнути на полі в області "Значення" -> "Параметри поля значень" -> вибрати іншу функцію: Середнє, Максимум, Мінімум, Кількість, Відхилення, Дисперсія тощо).
- **Аналіз та налаштування зведеної таблиці:** Після розміщення полів в областях макета зведена таблиця автоматично будується та відображає узагальнені дані. З'являються контекстні вкладки "Аналіз зведеної таблиці" (PivotTable Analyze) та "Конструктор" (Design).
 - **Вкладка "Аналіз зведеної таблиці":** Містить інструменти для роботи зі зведеною таблицею: зміна джерела даних, оновлення даних (якщо вихідні дані змінилися, натисніть "Оновити!"), зміна назви зведеної таблиці, додавання зрізів (Slicers) та часових шкал (Timelines) для інтерактивної фільтрації, створення обчислюваних полів/елементів, групування/розгрупування елементів рядків/стовпців (наприклад, групування дат за місяцями, роками, кварталами), відображення деталей (двічі клацнути на підсумковому значенні, щоб побачити вихідні дані).
 - **Вкладка "Конструктор":** Містить інструменти для зміни вигляду зведеної таблиці: вибір стилів, налаштування макета звіту (компактна форма, форма структури, таблична форма), відображення/приховування проміжних та загальних підсумків, порожніх рядків.
- **Зведені діаграми (PivotCharts):** Це динамічні діаграми, які безпосередньо пов'язані зі зведеною таблицею. Вони візуалізують дані зі зведеної таблиці і автоматично оновлюються при зміні структури зведеної таблиці, застосуванні фільтрів, групування тощо. Створюються зі зведеної таблиці через вкладку "Аналіз зведеної таблиці" -> група "Інструменти" -> "Зведена діаграма".

Зведені таблиці та діаграми є надзвичайно цінними інструментами для швидкого та глибокого аналізу даних, що робить їх незамінними в багатьох професійних сферах. Важливо пам'ятати, що якість результатів аналізу прямо

залежить від якості та структури вихідних даних.

ТЕМА 2.4. БАЗИ ДАНИХ ТА СИСТЕМИ УПРАВЛІННЯ БАЗАМИ ДАНИХ (СУБД)

Мета: Сформуванати розуміння базових понять бази даних та системи управління базами даних (СУБД), ознайомити з основними моделями даних, зосередивши увагу на реляційній моделі та її ключових концепціях, а також пояснити призначення та етапи нормалізації даних у реляційних БД.

План:

1. Поняття бази даних та СУБД.
2. Моделі даних (ієрархічна, мережева, реляційна).
3. Основні поняття реляційної моделі (відношення, атрибут, кортеж, ключі).
4. Нормалізація даних та нормальні форми.

1. Поняття бази даних та СУБД

У сучасному світі обсяги інформації зростають експоненційно. Для ефективного зберігання, організації та доступу до цих даних використовуються спеціалізовані системи – бази даних та системи управління базами даних.

- **Дані vs інформація:** Важливо розрізняти ці два поняття. **Дані** – це необроблені факти, числа, символи, що самі по собі можуть не нести значного сенсу (наприклад, "Іванов", "15000", "2025-05-01"). **Інформація** – це дані, які були оброблені, структуровані та представлені у значущому контексті, що дозволяє приймати рішення або отримувати знання (наприклад, "Менеджер Іванов отримав премію 15000 грн за квітень 2025 року"). Бази даних зберігають дані, а СУБД допомагають перетворювати ці дані на корисну інформацію шляхом запитів та аналізу.
- **База даних (БД):** Це не просто набір файлів. Це **структурована, організована сукупність взаємопов'язаних даних**, що описують певну предметну область (наприклад, облік студентів університету, інформація про товари на складі, банківські операції). Дані в БД зберігаються:
 - **Централізовано:** Зменшує дублювання та спрощує управління.
 - **З мінімальною надлишковістю:** Уникає повторення одних і тих самих даних у різних місцях, що допомагає підтримувати цілісність.

- **Взаємопов'язано:** Між різними наборами даних існують визначені зв'язки (наприклад, студентів та їхні оцінки, товари та постачальники).
- **Для спільного використання:** Багато користувачів або програм можуть одночасно отримувати доступ до даних та працювати з ними.
- **Система Управління Базами Даних (СУБД):** Це комплекс програмних засобів, який є посередником між користувачами/додатками та самою базою даних. СУБД виконує наступні функції:
 - **Створення та модифікація структури БД:** Визначення таблиць, полів, зв'язків, обмежень.
 - **Маніпулювання даними (CRUD-операції):**
 - **Create:** Додавання нових даних.
 - **Read:** Вибірка (пошук) даних (основна функція запитів).
 - **Update:** Оновлення існуючих даних.
 - **Delete:** Видалення даних.
 - **Забезпечення цілісності даних:** Підтримка правил, що гарантують коректність та несуперечність даних (наприклад, посиальна цілісність).
 - **Управління безпекою:** Контроль доступу користувачів до різних даних та операцій (аутентифікація та авторизація).
 - **Управління паралельним доступом:** Забезпечення коректної роботи при одночасному доступі багатьох користувачів, запобігання конфліктам.
 - **Відновлення після збоїв:** Можливість повернути БД у коректний стан після апаратних чи програмних збоїв.
 - **Резервне копіювання та відновлення:** Інструменти для створення копій даних та їх відновлення у разі втрати.
- **Приклади СУБД:** Існує багато різних СУБД, які відрізняються розміром, функціональністю, моделлю даних та ліцензуванням:
 - **Настільні/Файлові:** MS Access (для невеликих БД, входить до Office), SQLite (вбудовувана, популярна для мобільних додатків).
 - **Серверні (Реляційні):** MySQL, PostgreSQL (популярні Open Source), Microsoft SQL Server, Oracle Database, IBM Db2 (комерційні, для великих корпоративних систем).
 - **NoSQL (Нереляційні):** MongoDB (документно-орієнтована), Cassandra (колонково-орієнтована), Redis (ключ-значення). Використовуються для специфічних завдань (великі обсяги неструктурованих/напівструктурованих даних, висока швидкість запису).

2. Моделі даних

Модель даних – це концептуальний каркас, який визначає логічну структуру бази даних, тобто як дані організовані та як встановлюються зв'язки між різними

частинами даних. Вибір моделі даних впливає на спосіб проектування, реалізації та роботи з БД.

- **Ієрархічна модель:**

- **Структура:** Дані організовані у вигляді дерева. Є "кореневий" запис, від якого відгалужуються "нащадки". Кожен запис-нащадок може мати лише одного "предка".
- **Зв'язки:** Реалізуються як зв'язки типу "один до багатьох" (1:N). Наприклад, один відділ може мати багато співробітників, але кожен співробітник належить лише одному відділу.
- **Переваги:** Відносно проста для розуміння, швидкий доступ до даних вздовж ієрархічних шляхів (якщо шлях відомий).
- **Недоліки:** Складно представити зв'язки "багато до багатьох" (M:N), може призводити до надлишковості даних (якщо один запис-нащадок належить до різних предків у реальному світі), негнучка – зміни в структурі вимагають значних модифікацій. Приклад: IBM IMS.

- **Мережева модель:**

- **Структура:** Розглядається як узагальнення ієрархічної моделі. Дані організовані як граф. Запис може мати *кілька* "предків".
- **Зв'язки:** Дозволяє моделювати зв'язки "багато до багатьох" (M:N) безпосередньо.
- **Переваги:** Більш гнучка, ніж ієрархічна, краще справляється з M:N зв'язками, менша надлишковість порівняно з ієрархічною при моделюванні M:N.
- **Недоліки:** Дуже складна структура та навігація по даних ("навігаційний" доступ), складність проектування та модифікації схеми. Приклад: CODASYL.

- **Реляційна модель:**

- **Запропонована:** Едгар Кодд (Edgar F. Codd) з IBM у 1970 році.
- **Структура:** Дані представляються у вигляді колекції простих, незалежних двовимірних **таблиць (відношень)**.
- **Зв'язки:** Зв'язки між таблицями встановлюються не через фізичні покажчики, а **через збіг значень у спільних полях** (за допомогою ключів). Це є ключовою відмінністю від ієрархічної та мережевої моделей, де зв'язки є більш "фізичними".
- **Переваги:**
 - **Простота:** Інтуїтивно зрозуміла концепція таблиць.
 - **Гнучкість:** Легше додавати нові дані, змінювати структуру (хоча й існують правила).
 - **Строга математична основа:** Базується на теорії множин та реляційній алгебрі/численні, що забезпечує теоретичну обґрунтованість.

- **Незалежність даних:** Структура даних може змінюватися без необхідності зміни програм, що отримують до них доступ (логічна незалежність), а фізичне зберігання даних може змінюватися без зміни логічної структури (фізична незалежність).
- **Потужна мова запитів SQL:** Standard Query Language – декларативна мова, що дозволяє ефективно вибирати, фільтрувати, об'єднувати та модифікувати дані, не вказуючи, *як* це робити, а лише *що* потрібно отримати. Це значно простіше за "навігаційний" доступ попередніх моделей.
- **Недоліки:** Може бути менш ефективною для деяких типів даних або операцій порівняно з оптимізованими для них NoSQL моделями. Потенційні проблеми продуктивності при дуже складних запитах, що вимагають об'єднання великої кількості таблиць (операція JOIN).
- **Позиція:** Реляційна модель стала **фактичним стандартом** для більшості СУБД загального призначення і домінує досі. Більшість СУБД, з якими ви стикнетесь (MySQL, PostgreSQL, SQL Server, Oracle, Access), є реляційними СУБД (РСУБД).

3. Основні поняття реляційної моделі

Реляційна модель оперує специфічними термінами, запозиченими з математичної теорії відношень, але на практиці часто використовуються більш звичні синоніми:

- **Відношення (Relation) / Таблиця (Table):** Основний елемент зберігання даних. Це двовимірна структура, що містить дані про сутності одного типу (наприклад, таблиця "Студенти", таблиця "Товари", таблиця "Замовлення"). Кожна таблиця повинна мати унікальне ім'я в межах БД.
- **Атрибут (Attribute) / Стовець (Column) / Поле (Field):** Іменована вертикальна колонка в таблиці. Представляє певну характеристику або властивість сутності, описаної таблицею (наприклад, "ПІБ", "Дата народження", "Ціна", "Кількість"). Кожен атрибут в таблиці має унікальне ім'я (в межах цієї таблиці).
- **Домен (Domain):** Множина всіх допустимих значень для певного атрибута. Визначає тип даних (число, текст, дата) та будь-які додаткові обмеження на значення (наприклад, вік має бути додатним числом, ціна – більше нуля). В сучасних СУБД концепція домену часто реалізується через **типи даних** та **обмеження цілісності** (constraints).
- **Кортеж (Tuple) / Рядок (Row) / Запис (Record):** Горизонтальний рядок у таблиці. Представляє один конкретний екземпляр сутності (наприклад, один конкретний студент, один конкретний товар). Кожен кортеж містить набір значень, по одному для кожного атрибута таблиці, що описують цей екземпляр.

За правилами реляційної моделі, всі кортежі в таблиці мають бути унікальними (не може бути двох абсолютно однакових рядків).

- **Схема відношення:** Визначає структуру таблиці: її ім'я та список імен її атрибутів разом з їх доменами (типами даних). Наприклад, Студенти (Код_студента, ПІБ, Дата_народження, Група).
- **Схема бази даних:** Сукупність схем усіх відношень (таблиць), що входять до складу конкретної бази даних, разом з визначеними зв'язками та обмеженнями між ними.
- **Ключ (Key):** Один або кілька атрибутів, значення яких мають особливе значення для ідентифікації записів та встановлення зв'язків:
 - **Надключ (Superkey):** Набір атрибутів, який однозначно ідентифікує кожен кортеж у відношенні. Може містити зайві атрибути.
 - **Можливий ключ (Candidate Key):** Надключ, який є *мінімальним*, тобто жоден з його атрибутів не може бути видалений без втрати властивості унікальної ідентифікації. У кожній таблиці може бути один або кілька можливих ключів.
 - **Первинний ключ (Primary Key - PK):** Один із можливих ключів, обраний проектувальником БД для однозначної ідентифікації *кожного* запису в таблиці. Значення первинного ключа мають бути **унікальними** для кожного запису і **не можуть бути невизначеними (NULL)**. Таблиця може мати тільки один первинний ключ (хоча він може складатися з кількох полів).
 - **Альтернативний ключ (Alternate Key):** Будь-який можливий ключ, який не був обраний як первинний.
 - **Зовнішній ключ (Foreign Key - FK):** Атрибут або набір атрибутів в одній таблиці (яку називають "дочірньою" або залежною), значення якого **посилаються на значення первинного ключа** в іншій таблиці ("батьківській" або головній). Зовнішні ключі використовуються для встановлення зв'язків між таблицями та забезпечення **посилальної цілісності** (Referential Integrity) – правила, яке гарантує, що значення зовнішнього ключа в дочірній таблиці завжди має відповідне значення первинного ключа в батьківській таблиці. Це запобігає "висячим" посиланням.
 - **Складений ключ (Composite Key):** Ключ (первинний, можливий чи зовнішній), що складається з двох або більше атрибутів. Наприклад, комбінація "Номер замовлення" та "Код товару" може бути первинним ключем у таблиці "Елементи замовлення", де номер замовлення сам по собі не унікальний (одне замовлення містить багато товарів), і код товару сам по собі не унікальний (товар може бути в багатьох замовленнях), але їхня комбінація унікальна для кожного рядка.
- **Зв'язки між таблицями (Relationships):** В реляційній моделі зв'язки між сутностями предметної області моделюються через зв'язки між таблицями за

допомогою відповідності значень Первинного та Зовнішнього ключів. Найпоширеніші типи зв'язків:

- **Один до одного (1:1):** Кожен запис в одній таблиці відповідає не більше ніж одному запису в іншій таблиці, і навпаки. Реалізується через взаємне використання РК в обох таблицях або РК в одній таблиці та FK в іншій з унікальним індексом. Використовується рідко, зазвичай для розділення таблиці на дві частини з міркувань безпеки або продуктивності.
- **Один до багатьох (1:N):** Кожен запис в одній таблиці ("один" бік, батьківська таблиця, що містить РК) може відповідати багатьом записам в іншій таблиці ("багато" бік, дочірня таблиця, що містить FK), але кожен запис у "багато" боці відповідає лише одному запису в "один" боці. **Це найпоширеніший тип зв'язку.** Реалізується шляхом додавання поля Зовнішнього ключа до таблиці на "багато" боці, яке посилається на Первинний ключ таблиці на "один" боці. Приклад: один відділ (РК в таблиці "Відділи") має багато співробітників (FK на Код_відділу в таблиці "Співробітники").
- **Багато до багатьох (M:N):** Кожен запис в одній таблиці може відповідати багатьом записам в іншій, і кожен запис в іншій таблиці може відповідати багатьом записам в першій. Прямо реалізувати M:N зв'язок між двома таблицями в реляційній моделі неможливо. Він моделюється за допомогою створення **проміжної (зв'язуючої) таблиці**, яка має Первинний ключ, що складається з Зовнішніх ключів, які посилаються на Первинні ключі обох вихідних таблиць. Приклад: багато студентів відвідують багато курсів. Створюється проміжна таблиця "Зарахування", яка містить FK на "Код_студента" (з таблиці "Студенти") та FK на "Код_курсу" (з таблиці "Курси"). Первинний ключ таблиці "Зарахування" може бути складеним з полів "Код_студента" та "Код_курсу".

4. Нормалізація даних

Нормалізація – це формальний процес проектування (або вдосконалення) структури реляційної бази даних. Він полягає у послідовному аналізі таблиць та їх атрибутів з метою усунення надлишковості даних та залежностей, що можуть призвести до проблем (аномалій) при виконанні операцій вставки, оновлення та видалення даних. Мета нормалізації – привести структуру БД до однієї з нормальних форм.

- **Аномалії даних (Anomalies):** Проблеми, що виникають у ненормалізованих таблицях:
 - **Аномалія вставки (Insertion Anomaly):** Неможливість додати інформацію про один об'єкт, не маючи інформації про інший, або необхідність дублювати дані при додаванні нового запису. Наприклад,

у таблиці, що зберігає інформацію про товари та їх постачальників в одному рядку, неможливо додати інформацію про нового постачальника, поки він не почне постачати хоча б один товар.

- **Аномалія оновлення (Update Anomaly):** Необхідність оновлювати однакові дані в багатьох місцях. Якщо дані дублюються, і оновити їх не скрізь, виникає суперечність (неузгодженість даних). Наприклад, якщо адреса постачальника зберігається в рядках для кожного товару від цього постачальника, при зміні адреси потрібно оновити багато рядків. Якщо якийсь рядок пропущено, дані стануть неузгодженими.
- **Аномалія видалення (Deletion Anomaly):** Видалення одного запису призводить до небажаної втрати іншої, пов'язаної інформації. Наприклад, у тій самій таблиці "Товари-Постачальники", видалення запису про останній товар від певного постачальника призведе до видалення всієї інформації про цього постачальника.
- **Функціональна залежність (Functional Dependency):** Основа для розуміння нормалізації. Атрибут В функціонально залежить від атрибута А (позначається $A \rightarrow B$), якщо для кожного значення А існує тільки одне відповідне значення В. Атрибут А називається детермінантом. Нормалізація спрямована на усунення небажаних функціональних залежностей.
- **Переваги нормалізації:**
 - **Зменшення надлишковості даних:** Зменшення дублювання інформації, що економить місце зберігання та покращує цілісність.
 - **Підтримка цілісності даних:** Зміни вносяться в одному місці, що виключає аномалії оновлення. Посилальна цілісність легко підтримується.
 - **Гнучкість та спрощення модифікації:** Легше додавати нові сутності, атрибути та встановлювати нові зв'язки без значних змін у структурі.
 - **Потенційне прискорення деяких операцій:** Менші, простіші таблиці можуть оброблятися швидше.
- **Недоліки нормалізації:**
 - **Збільшення кількості таблиць:** Вищі рівні нормалізації призводять до розбиття таблиць на багато дрібніших.
 - **Ускладнення запитів (операція JOIN):** Для отримання інформації, яка раніше була в одній таблиці, тепер часто необхідно "з'єднувати" (JOIN) кілька таблиць. Операція JOIN може бути ресурсомісткою, особливо для великих таблиць, що потенційно сповільнює виконання запитів.
- **Нормальні форми (НФ):** Послідовність правил, що визначають рівні нормалізації. Кожна наступна нормальна форма є суворішою за попередню. Досягнення вищої НФ усуває певні типи аномалій. Найпоширеніші НФ, до яких прагнуть при проектуванні:
 - **Перша нормальна форма (1НФ):**

- Всі атрибути є **атомарними** (неподільними) – не можна мати списки значень в одній комірці або складні структури.
- Відсутні **групи, що повторюються** – кожен атрибут має бути в окремому стовпці, а не повторюватися в кількох стовпцях (наприклад, Товар1, Кількість1, Товар2, Кількість2).
- Кожна таблиця має **первинний ключ**.
- Друга нормальна форма (2НФ):
 - Таблиця має бути в 1НФ.
 - Всі **неключові атрибути** (ті, що не входять до складу первинного ключа) мають **повністю функціонально залежати від УСЬОГО первинного ключа**. Це правило важливе, коли первинний ключ є складеним. Якщо неключовий атрибут залежить лише від *частини* складеного первинного ключа, це порушення 2НФ.
- Третя нормальна форма (3НФ):
 - Таблиця має бути в 2НФ.
 - Відсутні **транзитивні залежності неключових атрибутів від первинного ключа**. Це означає, що неключовий атрибут не повинен функціонально залежати від *іншого неключового атрибута*. Якщо $A \rightarrow B$ і $B \rightarrow C$ (де C – неключовий атрибут, що залежить від неключового B), то існує транзитивна залежність.
- **Нормальна форма Бойса-Кодда (НФБК або BCNF):** Більш строга версія 3НФ. Таблиця в BCNF, якщо **кожен детермінант** (атрибут або група атрибутів, що функціонально визначає інший атрибут) є **можливим ключем**.
- **Вищі нормальні форми (4НФ, 5НФ, ДКНФ):** Вирішують більш специфічні та складні типи залежностей (багатозначні залежності, залежності з'єднання) і рідше досягаються на практиці, якщо це не викликано специфічними вимогами.
- **Процес нормалізації:** Полягає у послідовному застосуванні правил нормальних форм. Якщо таблиця не відповідає вимогам певної НФ, вона розбивається (декомпозиція) на дві або більше таблиць. Ці нові таблиці зв'язуються за допомогою зовнішніх ключів, і процес нормалізації продовжується для нових таблиць.
- **Компроміс: Нормалізація vs денормалізація:** Хоча досягнення високого рівня нормалізації (як правило, 3НФ або BCNF) є метою для забезпечення цілісності даних та усунення аномалій, на практиці іноді застосовують **денормалізацію**. Це свідоме додавання контрольованої надлишковості до нормалізованої структури БД (наприклад, додавання обчислюваного поля або дублювання деяких даних у таблиці, де вони часто потрібні) з метою **оптимізації продуктивності** найчастіших запитів, особливо тих, що вимагають складних JOIN-операцій. Рішення про денормалізацію має бути обґрунтованим і ґрунтуватися на аналізі реальних потреб та навантаження на БД.

ТЕМА 2.5. СУБД MS ACCESS: ОСНОВИ РОБОТИ

Мета: Ознайомити з СУБД MS Access як настільною реляційною системою, навчити створювати базу даних та проектувати таблиці, правильно вибирати типи даних та налаштовувати властивості полів, а також розуміти та використовувати ключі для зв'язування таблиць та забезпечення цілісності даних.

План:

1. Призначення та інтерфейс MS Access.
2. Створення бази даних та робота з таблицями.
3. Типи даних та властивості полів.
4. Ключі та їх використання (первинний, зовнішній).
5. Введення та редагування даних.

1. Призначення та інтерфейс MS Access

MS Access – це СУБД, яка орієнтована переважно на індивідуальних користувачів або невеликі робочі групи. Вона ідеально підходить для:

- Створення та управління базами даних відносно невеликого та середнього розміру.
- Швидкої розробки простих додатків баз даних (з формами та звітами) без необхідності знання мов програмування (хоча підтримує VBA).
- Збору, організації та представлення інформації для аналізу.
- Створення прототипів баз даних перед розробкою на більш потужних серверних СУБД.

Access використовує **реляційну модель даних** і зберігає всю базу даних (таблиці, запити, форми, звіти, макроси, модулі) в одному файлі (.accdb для сучасних версій, .mdb для старих). Це спрощує поширення, але обмежує масштабованість та кількість одночасних користувачів порівняно з серверними СУБД, де дані зберігаються на окремому сервері.

Інтерфейс Access має багато спільного з іншими програмами Microsoft Office:

- **Стрічка (Ribbon):** Знаходиться вгорі вікна, містить вкладки:
 - "Файл": Область Backstage (операції з файлом БД).
 - "Головна": Основні команди для роботи з об'єктами (перегляди, буфер обміну, сортування, фільтрація, пошук).
 - "Створення": Для створення нових об'єктів БД (таблиць, запитів, форм, звітів тощо).
 - "Зовнішні дані": Для імпорту, експорту або зв'язування з даними з інших джерел (Excel, текстові файли, інші БД, SharePoint).

- "Знаряддя бази даних": Інструменти для управління БД (схема даних, макроси, VBA, оптимізація, аналіз).
- Контекстні вкладки: З'являються при роботі з певним об'єктом у певному режимі (наприклад, "Конструктор таблиць", "Знаряддя макета форми").
- **Подання Backstage (Вкладка "Файл"):** Керування файлом бази даних (.accdb): створення нової БД, відкриття існуючої, збереження (хоча зміни зберігаються автоматично), збереження об'єкта під іншим ім'ям, друк, інформація про БД (стиснення, шифрування), параметри програми.
- **Область переходів (Navigation Pane):** Розташована зліва. Надає зручний спосіб перегляду, відкриття та організації всіх об'єктів вашої бази даних. Об'єкти можуть бути згруповані за типом (Таблиці, Запити, Форми, Звіти тощо) або іншими критеріями. Можна приховувати або розгортати цю панель.
- **Вікно об'єкта:** Основна робоча область справа, де відображається вміст або структура вибраного об'єкта. Кожен відкритий об'єкт відображається у своїй вкладці в цьому вікні.

2. Створення бази даних та робота з таблицями

Робота в Access починається зі створення файлу бази даних та визначення її структури, яка представлена таблицями.

- **Створення бази даних:**
 - Запустіть Access.
 - Виберіть "Файл" -> "Створити".
 - Можна обрати "Пуста база даних" для створення БД з нуля або вибрати один з **шаблонів** (Templates) для типових завдань (Управління проектами, Облік активів, Контакти тощо). Шаблони вже містять готові таблиці, форми та звіти, що значно прискорює початок роботи.
 - При створенні пустої БД, необхідно вказати **ім'я файлу** бази даних та **місце** її збереження на диску. Розширення файлу буде .accdb.
- **Об'єкти БД Access:** Access оперує не тільки таблицями, але й іншими об'єктами, які допомагають взаємодіяти з даними та представляти їх:
 - **Таблиці (Tables):** Основне місце зберігання всіх даних. Визначають структуру даних (стовпці/поля, рядки/записи).
 - **Запити (Queries):** Для вибірки (отримання), фільтрації, сортування, обчислення, узагальнення даних з однієї або кількох таблиць. Також використовуються для зміни даних (запити дії).
 - **Форми (Forms):** Інтерфейс для зручного введення, перегляду та редагування даних у таблицях або запитах. Імітують паперові бланки.
 - **Звіти (Reports):** Для форматування, узагальнення та представлення даних у вигляді друкованих документів.

- **Макроси (Macros):** Набір команд, які можна використовувати для автоматизації типових завдань (наприклад, відкриття форми, виконання запиту, виведення повідомлення).
- **Модулі (Modules):** Для написання програмного коду мовою VBA (Visual Basic for Applications) для реалізації більш складної логіки та функціональності.
- **Створення таблиць:** Таблиці є фундаментом бази даних. Без таблиць не може бути інших об'єктів, оскільки вони працюють з даними, що зберігаються в таблицях. Способи створення таблиць в Access:
 - **Режим таблиці (Datasheet View):** Швидкий спосіб створити просту таблицю. Після створення пустої БД Access автоматично відкриває нову пусту таблицю в цьому режимі. Ви можете просто вводити дані в колонки, і Access спробує автоматично визначити тип даних для кожного стовпця на основі введених даних. Можна двічі клацнути на заголовку стовпця "Клацніть, щоб додати поле", щоб перейменувати його, і обрати тип даних зі списку. Це зручно для швидкого початку, але не дає повного контролю над структурою.
 - **Режим конструктора (Design View): Основний та рекомендований режим** для професійного проектування структури таблиці. У цьому режимі ви спочатку визначаєте структуру таблиці, а потім вже вводите дані. У режимі конструктора ви для кожного поля задаєте:
 - **Ім'я поля (Field Name):** Унікальне ім'я (до 64 символів, без крапок ., знаків !, [], пробілів на початку/в кінці, керуючих символів; краще використовувати англійські літери, цифри та нижнє підкреслення).
 - **Тип даних (Data Type):** Вибираєте зі списку доступних типів даних (див. п. 3).
 - **Опис (Description):** (Необов'язково) Додатковий текст для пояснення призначення поля.
 - **Властивості поля (Field Properties):** Детальні налаштування для вибраного поля (див. п. 3).
 - **Первинний ключ (Primary Key):** Встановлюється в цьому режимі (див. п. 4).
 - **Майстер таблиць (Table Wizard):** Допомогає створити таблицю на основі готових, стандартних шаблонів для типових предметних областей (наприклад, дані про клієнтів, замовлення, продукти, співробітників). Ви вибираєте шаблон, потрібні поля з нього, Access створює структуру таблиці.
 - **Створення таблиці за допомогою SharePoint Lists:** Можна імпортувати дані зі списку SharePoint або зв'язати таблицю з ним.

Хоча режим таблиці зручний для швидкого введення, режим конструктора є

необхідним для створення надійної та правильно структурованої бази даних з визначеними типами даних, властивостями та ключами. Перемикається між режимами можна за допомогою кнопок у групі "Подання" на вкладці "Головна" або клацанням правою кнопкою миші на ярличку об'єкта та вибором режиму.

3. Типи даних та властивості полів

Вибір правильного типу даних для кожного поля в таблиці є одним з найважливіших етапів проектування. Він впливає на:

- **Тип даних, які можна зберігати** в полі.
- **Обсяг пам'яті**, який займає поле.
- **Операції**, які можна виконувати над даними цього поля (наприклад, математичні операції тільки над числовими типами).
- **Швидкість** доступу до даних та виконання запитів.
- **Цілісність даних**.

Access пропонує стандартний набір типів даних:

- **Короткий текст (Short Text):** Для зберігання текстових або числових даних, які не використовуються в обчисленнях (імена, адреси, поштові індекси, телефонні номери). Обмеження: до 255 символів.
- **Довгий текст (Long Text або Мемо у старих версіях):** Для великих обсягів тексту (примітки, описи). Не має обмеження на кількість символів на рівні зберігання (до 1 ГБ), але елементи керування відображають лише перші 64000 символів.
- **Число (Number):** Для числових даних, що будуть використовуватися в математичних обчисленнях. У властивостях поля можна вибрати більш специфічний тип числа (Розмір поля):
 - **Байт:** Цілі числа від 0 до 255.
 - **Ціле (Integer):** Цілі числа від -32,768 до 32,767.
 - **Довге ціле (Long Integer):** Цілі числа від -2,147,483,648 до 2,147,483,647. Найпоширеніший для цілих.
 - **Одинарної точності (Single):** Числа з плаваючою комою, займає 4 байти.
 - **Подвійної точності (Double):** Числа з плаваючою комою, займає 8 байтів. Найпоширеніший для дробових чисел.
 - **Десяткове (Decimal):** Для чисел з фіксованою точністю, займає 12 байтів. Використовується, коли потрібна висока точність без помилок округлення, характерних для плаваючої коми (наприклад, для фінансових розрахунків).
- **Велике число (Large Number):** Для дуже великих цілих чисел, що виходять за діапазон Long Integer (тільки у форматі .accdB). Займає 8 байтів.
- **Дата й час (Date/Time):** Для зберігання значень дати та часу. Займає 8 байтів. У властивості "Формат" можна вибрати спосіб відображення.

- **Дата й час подовжено (Date/Time Extended):** (Тільки у форматі .accdb) Надає більший діапазон дат та вищу точність для часу порівняно зі стандартним Date/Time. Займає 42 байти у рядку.
- **Грошова одиниця (Currency):** Спеціальний тип даних для грошових значень. Займає 8 байтів, зберігає число з фіксованою точністю до 4 знаків після коми. Призначений для уникнення помилок округлення у фінансових обчисленнях.
- **Автономерація (AutoNumber):** Автоматично присвоює унікальне послідовне число або випадковий GUID (глобальний унікальний ідентифікатор) кожному новому запису. Це значення не може бути змінено користувачем. Ідеально підходить для використання як Первинний ключ, оскільки гарантує унікальність.
- **Так/Ні (Yes/No):** Для зберігання логічних значень – Істина (Yes) або Хибність (No). Зберігається як 1 байт (-1 для Yes, 0 для No). Може відображатися як прапорець, перемикач або поле зі списком "Так/Ні".
- **Об'єкт OLE (OLE Object):** Для вбудовування або зв'язування з об'єктами, створеними в інших програмах, що підтримують технологію OLE (Object Linking and Embedding) – наприклад, зображення, документи Word, електронні таблиці Excel. Не рекомендується для зберігання великої кількості зображень через неефективне використання простору та потенційні проблеми з продуктивністю.
- **Гіперпосилання (Hyperlink):** Для зберігання веб-адрес, електронних адрес, шляхів до файлів або папок. При клацанні на посиланні відкривається відповідний ресурс.
- **Вкладення (Attachment):** (Тільки у форматі .accdb) Дозволяє зберігати один або кілька файлів (будь-якого типу) безпосередньо в записі бази даних. Зручно для приєднання документів, зображень, PDF-файлів до конкретного запису.
- **Обчислюваний (Calculated):** (Тільки у форматі .accdb) Створює поле, значення якого автоматично обчислюється за формулою, визначеною користувачем, на основі значень в інших полях *цієї ж таблиці*. Це поле не зберігає дані фізично, його значення обчислюється при перегляді запису. Формула може використовувати оператори, посилання на інші поля та функції. Наприклад, $\text{=[Кількість]} * \text{[Ціна]}$.
- **Майстер підстановок (Lookup Wizard):** Технічно це не окремий тип даних, а інструмент, що допомагає створити поле зі списком значень. Значення можуть братися зі списку, який ви вводите вручну, або (що частіше) з іншого поля в *іншій таблиці*. Майстер підстановок насправді створює поле з типом даних, який відповідає типу даних поля, з якого беруться значення підстановки (зазвичай Число або Короткий текст), і автоматично налаштовує властивості поля та зв'язок. Це візуально спрощує введення даних (вибір зі списку), але може ускладнити розуміння структури БД, оскільки приховує реальний механізм зовнішнього ключа.

Властивості полів (Field Properties): У режимі конструктора таблиці, вибравши поле, в нижній частині вікна можна налаштувати його властивості. Ці властивості визначають, як Access обробляє, зберігає та відображає дані в цьому полі:

- **Загальні властивості:**

- **Розмір поля (Field Size):** (Для Короткого тексту, Числа, Великого числа, Автонумерації) Обмежує максимальну кількість символів або діапазон чисел, що можуть зберігатися. Правильне налаштування економить місце та підвищує продуктивність.
- **Формат (Format):** (Для всіх типів) Визначає спосіб відображення даних без зміни їхнього фактичного значення (наприклад, "Загальна дата", "Довгий час", "Грошовий", "Відсоток", "0.00").
- **Маска вводу (Input Mask):** (Для Короткого тексту, Числа, Дати/часу, Грошової одиниці) Визначає шаблон для введення даних, забезпечуючи їхню відповідність певному формату (наприклад, (999) 000-0000 для номера телефону). Допомагає уникнути помилок введення. Можна створити за допомогою Майстра масок вводу.
- **Підпис (Caption):** Текст, який відображається як заголовок стовпця в режимі таблиці за замовчуванням, а також як підпис елемента керування для цього поля у формах та звітах. Якщо Підпис не вказано, використовується Ім'я поля. Зручно використовувати більш зрозумілі підписи, ніж технічні імена полів.
- **Значення за замовчуванням (Default Value):** Значення, яке автоматично вставляється в це поле при створенні нового запису, якщо користувач не вводить інше значення.
- **Умова на значення (Validation Rule):** Логічний вираз, який перевіряє коректність даних, введених у поле. Якщо умова не виконується, Access не дозволить зберегти запис. Наприклад, <100 для поля "Кількість" (не більше 100) або Is Not Null (поле не може бути пустим).
- **Повідомлення про помилку (Validation Text):** Текст повідомлення, який відображається користувачеві, якщо введені дані не відповідають Умові на значення. Робить повідомлення про помилку зрозумілішим.
- **Обов'язкове поле (Required):** Властивість Так/Ні. Якщо встановлено Так, поле не може залишатися пустим (NULL). Це гарантує наявність даних у цьому полі для кожного запису. **Відмінність від "Дозволити пусті рядки":** "Required: Так" означає, що поле не може бути NULL. "Дозволити пусті рядки: Ні" (для текстових полів) означає, що поле не може містити пустий рядок (""), але може бути NULL, якщо "Required: Ні". Для числових полів та дат NULL означає відсутність значення, а 0 або пустий рядок – певне значення.

- **Дозволити пусті рядки (Allow Zero Length):** (Тільки для Короткого тексту, Довгого тексту, Гіперпосилання) Властивість Так/Ні. Якщо встановлено Ні, користувач не зможе ввести пустий рядок "" в це поле. Це корисно, якщо ви хочете, щоб текстове поле або містило текст, або було NULL (якщо Required=Ні), але не було просто порожнім рядком.
- **Індексоване поле (Indexed):** Властивість Так/Ні/Збіги не допускаються. Створення індексу для поля прискорює пошук, сортування та виконання запитів, що використовують це поле в умовах або зв'язках. "Індексоване: Так (Збіги допускаються)" дозволяє дублювання значень. "Індексоване: Так (Збіги не допускаються)" створює унікальний індекс, який забороняє введення дубльованих значень у це поле. Останнє використовується для полів-кандидатів на первинний ключ або альтернативних ключів.
- **Властивості для підстановок:** З'являються, якщо для поля використовується Майстер підстановок. Дозволяють налаштувати відображення списку підстановки.

4. Ключі та їх використання

Ключі є фундаментальною концепцією реляційної моделі, що забезпечує структуру, унікальність записів та встановлення зв'язків між таблицями.

● **Первинний ключ (Primary Key - PK):**

- **Призначення:** Унікально ідентифікує *кожен окремий запис* (рядок) у таблиці. Це означає, що значення РК ніколи не повторюється в різних рядках однієї таблиці.
- **Вимоги:**
 - **Унікальність:** Значення РК мають бути унікальними для кожного запису.
 - **Не NULL:** Значення РК не можуть бути порожніми (NULL).
- **Типи полів, що використовуються як РК в Access:**
 - **Автонумерація (AutoNumber):** Найкращий вибір для більшості випадків, оскільки Access автоматично генерує унікальні значення і гарантує відсутність NULL. Це **штучний ключ**, який не несе смислового навантаження, але ідеально виконує функцію ідентифікації.
 - **Простий ключ:** Одне поле з іншим типом даних (наприклад, Короткий текст або Число), якщо ви абсолютно впевнені в унікальності та непорожності його значень для кожного запису (наприклад, номер паспорта, код ЄДРПОУ для юридичної особи).
 - **Складений ключ (Composite Key):** Комбінація двох або більше полів. Унікальність забезпечується лише унікальністю *комбінації* значень цих полів. Використовується, коли одного поля

недостатньо для унікальної ідентифікації (наприклад, у зв'язуючій таблиці M:N зв'язку).

- **Встановлення Первинного ключа в Access:** У режимі конструктора таблиці виділіть поле або кілька полів (утримуючи Ctrl) та натисніть кнопку "Ключове поле" (Primary Key) на вкладці "Конструктор таблиць". Access відобразить значок ключа ліворуч від імені поля і автоматично створить для нього унікальний індекс. Якщо ви не встановите РК вручну, Access при збереженні таблиці запропонує створити поле "Код" з типом Автономерація та встановити його як РК.
- **Вибір РК:** Намагайтеся уникати використання як первинного ключа полів, значення яких можуть з часом змінюватися (наприклад, ПІБ, адреса, номер телефону) або які можуть дублюватися (наприклад, ім'я). Штучний ключ (Автономерація) є найнадійнішим вибором.
- **Зовнішній ключ (Foreign Key - FK):**
 - **Призначення:** Поле або набір полів в одній таблиці (дочірній), значення якого відповідає значенню Первинного ключа в іншій таблиці (батьківській). Зовнішній ключ встановлює **логічний зв'язок** між таблицями та забезпечує **посилальну цілісність**.
 - **Посилальна цілісність (Referential Integrity):** Правило, яке гарантує, що ви не зможете ввести значення у поле Зовнішнього ключа дочірньої таблиці, якщо такого значення не існує у відповідному полі Первинного ключа батьківської таблиці. Це запобігає "висячим" посиланням. Також при ввімкненій посилальній цілісності Access може автоматично оновлювати або видаляти залежні записи при зміні/видаленні запису в батьківській таблиці (каскадні операції).
 - **Встановлення зв'язків та FK в Access:** Зв'язки між таблицями визначаються у вікні "Схема даних" (Database Tools -> Relationships). Ви перетягуєте поле Первинного ключа з батьківської таблиці на відповідне поле Зовнішнього ключа в дочірній таблиці. У діалоговому вікні "Змінення зв'язків" встановлюєте прапорець "Забезпечення цілісності даних". Access візуально відобразить зв'язок між таблицями лінією, на якій можуть бути позначки "1" та символу нескінченності "∞", що вказують на зв'язок "один до багатьох" (1:N).
- **Індексування:** Крім автоматичного індексування Первинного ключа, можна вручну створювати індекси для інших полів, які часто використовуються для пошуку, сортування або умов у запитах. Це значно прискорює ці операції, але може сповільнювати додавання та оновлення даних, оскільки індекси також потрібно оновлювати. У режимі конструктора таблиці у властивостях поля "Індексоване" вибирається "Так (Збіги допускаються)" або "Так (Збіги не допускаються)".

5. Введення та редагування даних

Після того, як структура таблиць визначена та збережена, можна переходити до наповнення їх даними. Це зазвичай робиться в режимі таблиці (Datasheet View) або за допомогою форм.

- **Введення даних у режимі таблиці:**

- Відкрийте таблицю (двічі клацнувши на ній в Області переходів). Вона відкриється в режимі таблиці.
- Останній рядок таблиці позначений зірочкою (*) – це рядок для **додавання нового запису**. Клацніть в першій комірці цього рядка та почніть вводити дані.
- Для переміщення між комірками в поточному записі використовуйте клавіші **Tab** (вперед) та **Shift+Tab** (назад).
- Після введення даних в останнє поле запису, натисніть **Enter** або **Tab** – курсор автоматично переміститься до першого поля наступного нового запису (*). Access автоматично зберігає введені дані при переході на інший рядок або закритті таблиці.
- Клавіші керування курсором (стрілки) дозволяють переміщатися між комірками. Home/End – переміщення на початок/кінець рядка. Page Up/Page Down – переміщення на екран вгору/вниз.

- **Редагування даних:**

- Клацніть в комірці, яку потрібно відредагувати. Текстовий курсор з'явиться в комірці.
- Внесіть зміни.
- Натисніть Enter або перейдіть до іншої комірки/запису для збереження змін.
- Клавіша **F2** переводить комірку в режим редагування, дозволяючи редагувати весь вміст.

- **Навігація між записами:**

- Використовуйте смуги прокручування для переміщення по рядках.
- Використовуйте **кнопки навігації записами** внизу вікна (Перший запис, Попередній запис, Номер поточного запису, Наступний запис, Останній запис, Новий (пустий) запис).

- **Видалення записів:**

- Виділіть запис (рядок), клацнувши на сірому маркері виділення зліва від запису.
- Натисніть клавішу **Delete** на клавіатурі.
- Access запитає підтвердження видалення. Будьте обережні, видалені дані не можна легко відновити, якщо не було резервної копії.
- Якщо видалення запису порушує посилальну цілісність (наприклад, ви намагаєтеся видалити відділ, в якому є співробітники, і для зв'язку не дозволено каскадне видалення), Access повідомить про помилку.

- **Пошук та Заміна:**

- Виберіть поле, в якому потрібно виконати пошук (або всю таблицю).
- На вкладці "Головна" в групі "Пошук" натисніть кнопку "Знайти" (або Ctrl+F). Введіть шуканий текст/значення. Можна налаштувати параметри пошуку (де шукати, з урахуванням регістру, за збігом).
- Кнопка "Замінити" (або Ctrl+H) дозволяє не тільки знайти, але й автоматично замінити знайдені значення на інші.
- **Просте сортування та фільтрація:**
 - **Сортування:** Клацніть правою кнопкою миші на заголовку стовпця і виберіть "Сортування від А до Я" або "Сортування від Я до А".
 - **Фільтрація:** Клацніть на кнопці зі стрілкою в заголовку стовпця. З'явиться список унікальних значень. Можна зняти/поставити прапорці, щоб показати/приховати записи з цими значеннями. Також доступні текстові, числові фільтри, фільтри за датами. Щоб скасувати фільтр, клацніть на тій самій кнопці та виберіть "Очистити фільтр". Кнопка "Фільтр" на стрічці дозволяє застосувати більш складні фільтри "за виділенням" або "за формою".

ТЕМА 2.6. СУБД MS ACCESS: ЗАПИТИ, ФОРМИ, ЗВІТИ

Мета: Навчити створювати та ефективно використовувати ключові об'єкти бази даних MS Access – запити для вибірки та маніпулювання даними, форми для зручного користувацького інтерфейсу вводу/редагування, та звіти для узагальнення та представлення даних.

План:

1. Створення запитів на вибірку (QBE та конструктор).
2. Розробка форм для введення та перегляду даних.
3. Створення звітів для представлення даних.

Після того, як ви створили таблиці та визначили зв'язки між ними, ви маєте сховище даних. Але щоб отримати з цих даних корисну інформацію, взаємодіяти з ними зручним способом та представляти їх у форматованому вигляді, вам знадобляться інші об'єкти Access: запити, форми та звіти.

1. Створення запитів на вибірку (QBE)

Запити є основним інструментом для видобутку, фільтрації, сортування, обчислення та узагальнення даних з однієї або кількох пов'язаних таблиць. Запити на вибірку (Select Queries) дозволяють отримати підмножину даних, що відповідає певним критеріям, у вигляді тимчасової таблиці-результату.

- **Способи створення запиту на вибірку (вкладка "Створення" -> група "Запити"):**
 - **Майстер запитів (Query Wizard):** Покроковий помічник, який веде вас через процес створення запиту. Найчастіше використовується "Майстер простих запитів" для вибору полів з однієї або кількох пов'язаних таблиць. Зручно для початківців або простих запитів.
 - **Конструктор запитів (Query Design):** Найпотужніший та гнучкий інструмент для створення запитів будь-якої складності. Використовує принцип QBE (Query By Example - запит за зразком), де ви візуально задаєте структуру та умови запиту, а Access генерує відповідний SQL-код.
 - **SQL-подання (SQL View):** Дозволяє переглядати та редагувати запит безпосередньо мовою SQL (Standard Query Language). Режими Конструктора та SQL-подання синхронізовані – зміни в одному режимі відображаються в іншому.
- **Робота в Конструкторі запитів:**
 - **Додавання таблиць або інших запитів:** При відкритті Конструктора запитів з'являється діалогове вікно "Відобразити таблицю". Виберіть таблиці та/або інші запити, що містять потрібні вам дані, та натисніть "Додати". Додані об'єкти з'являться у верхній частині вікна конструктора з відображенням зв'язків між ними.
 - **Вибір полів:** Щоб включити поле до результату запиту, двічі клацніть на його імені у списку полів таблиці у верхній частині вікна, або перетягніть його ім'я з таблиці у нижню сітку бланка запиту. Кожне поле відображається в окремому стовпці сітки.
 - **Налаштування сітки запиту (QBE Grid):** Нижня частина вікна конструктора – це сітка, де ви налаштовуєте запит для кожного поля:
 - **Поле (Field):** Ім'я поля.
 - **Таблиця (Table):** Назва таблиці, з якої взято поле.
 - **Сортування (Sort):** Виберіть порядок сортування результату за цим полем (За зростанням, За спаданням, Відсутнє). Можна сортувати за кількома полями – порядок стовпців у сітці визначає пріоритет сортування (зліва направо).
 - **Відображення (Show):** Прапорець. Якщо встановлено, поле буде відображатися в результаті запиту. Якщо знято, поле використовується для умов відбору або сортування, але не показується користувачеві.
 - **Умови відбору (Criteria):** Введіть вирази для фільтрації записів. Тільки записи, що відповідають цим умовам, будуть включені до результату.
 - Рівність: = "Київ" або просто "Київ".
 - Нерівність: <> "Закінчено".

- Порівняння: > 100, <= #2024-12-31# (дати беруться в #).
- Діапазон: Between 100 And 200, Between #2025-01-01# And #2025-01-31#.
- Шаблон (для тексту): Like "Іван*" (знайти всіх, чиє ім'я починається на Іван), Like "?????" (знайти текст рівно з 4 символів).
- Пусте/не пусте: Is Null, Is Not Null.
- **Умови "або:" (Or):** Додаткові рядки під "Умови відбору" дозволяють задавати альтернативні умови, об'єднані логічним оператором OR. Умови в одному рядку об'єднуються за And.
- **Обчислювані поля:** Можна створити нове поле, значення якого обчислюється на основі виразу. У вільному стовпці сітки в рядку "Поле" введіть: НовеІм'яПоля: Вираз. Наприклад, ЗагальнаВартість: [КількістьЗамовлено] * [Ціна]. Імена полів у виразах беруться в квадратні дужки [].
- **Підсумкові запити (Totals Query):** Якщо потрібно згрупувати дані та виконати агрегатні обчислення (сума, кількість, середнє тощо) для кожної групи, натисніть кнопку "Підсумки" (Σ) на вкладці "Конструктор запитів". У сітці з'явиться рядок "Підсумки". Для полів, за якими потрібно групувати, виберіть "Групування за" (Group By). Для полів, за якими потрібно обчислити підсумок, виберіть потрібну агрегатну функцію (Sum, Avg, Count, Max, Min, StDev, Var, First, Last, Expression, Where). Наприклад, щоб підрахувати загальну кількість товарів кожного типу, згрупуйте за полем "Тип товару" та застосуйте функцію "Sum" до поля "Кількість".
- **Зв'язки в запитах (JOINS):** Якщо запит використовує дані з кількох таблиць, які пов'язані у Схемі даних, Access автоматично відображає ці зв'язки. Якщо зв'язки не визначено у Схемі даних, їх можна тимчасово створити у вікні конструктора запитів, перетягнувши поле з однієї таблиці на відповідне поле в іншій. Типи з'єднань (JOIN types) впливають на те, які рядки будуть включені до результату, коли є записи без відповідності в іншій таблиці (внутрішнє з'єднання - Inner Join (за замовчуванням), ліве/праве зовнішнє з'єднання - Left/Right Outer Join).
- **Типи запитів за призначенням (вкладка "Конструктор запитів" -> група "Тип запиту"):**
 - **Запит на вибірку (Select):** Основний тип, отримує дані.
 - **Запити дії (Action Queries):** Змінюють дані в таблицях. **Будьте обережні при їх використанні!**
 - **Запит на створення таблиці (Make-Table Query):** Створює нову таблицю на основі результату запиту на вибірку.

- **Запит на оновлення (Update Query):** Змінює значення в існуючих записах, що відповідають умовам.
- **Запит на додавання (Append Query):** Додає записи з однієї таблиці (або результату запиту) до кінця іншої таблиці.
- **Запит на видалення (Delete Query):** Видаляє записи з однієї або кількох таблиць, що відповідають умовам.
- **Виконання та збереження запиту:**
 - Щоб побачити результат запиту, натисніть кнопку "Запуск" (!) на вкладці "Конструктор запитів" або перейдіть у режим "Таблиця" (Datasheet View).
 - Щоб зберегти структуру запиту для подальшого використання, натисніть кнопку "Зберегти" або закрийте запит і погодьтеся на збереження, вказавши ім'я. Збережений запит з'явиться в Області переходів.

2. Розробка форм для введення та перегляду даних

Форми в Access – це об'єкти, що надають зручний, часто більш естетичний та контрольований спосіб для роботи користувача з даними таблиць або запитів. Вони дозволяють представляти дані одного запису (або пов'язаних записів) одночасно, що зручніше, ніж перегляд великої таблиці.

- **Призначення форм:**
 - Спрощення введення нових даних.
 - Зручне редагування існуючих даних.
 - Наочний перегляд даних.
 - Реалізація бізнес-логіки (наприклад, автоматичні обчислення у полях).
 - Керування іншими об'єктами БД (наприклад, кнопки для відкриття звітів, виконання запитів).
 - Підвищення зручності та захисту даних для кінцевих користувачів.
- **Способи створення форм (вкладка "Створення" -> група "Форми"):**
 - **Форма (Form):** Найшвидший спосіб. Створює просту форму в один стовпець на основі таблиці або запиту, який був виділений в Області переходів. Відразу відкриває форму в режимі макета, де можна легко змінити розташування елементів.
 - **Майстер форм (Form Wizard):** Покроковий інструмент. Дозволяє вибрати поля з однієї або кількох пов'язаних таблиць/запитів, визначити макет форми (в один стовпець, стрічковий - кілька записів одночасно, табличний - як таблиця, вирівняний), вибрати стиль оформлення та вказати ім'я форми.
 - **Розділена форма (Split Form):** Створює форму, яка одночасно відображає дані в режимі форми (для поточного запису) у верхній або нижній частині вікна та в режимі таблиці (для всіх записів) в іншій

частині. Дуже зручно для навігації та пошуку записів у табличній частині та детального перегляду/редагування в частині форми.

- **Кілька елементів (Multiple Items):** Створює форму, яка відображає кілька записів одночасно в стрічковому макеті, схожому на таблицю, але з більшими можливостями керування дизайном.
- **Пуста форма (Blank Form):** Створює нову пусту форму в режимі макета, де ви можете вручну додавати потрібні поля та елементи керування.
- **Конструктор форм (Form Design):** Надає повний контроль над дизайном та функціональністю форми. Відкриває пусте вікно форми в режимі конструктора, де ви можете вручну розміщувати елементи керування з панелі інструментів або зі списку полів джерела даних.
- **Джерело даних форми (Record Source):** Форма завжди відображає дані, що беруться з певного джерела. Цим джерелом може бути **таблиця** або **запит**. Вказується у властивості форми "Джерело записів". Використання запиту як джерела даних є дуже гнучким, оскільки дозволяє формі працювати тільки з попередньо відфільтрованими, відсортованими, обчисленими або об'єднаними даними.
- **Елементи керування форми (Controls):** Об'єкти, що розміщуються на формі для відображення даних або взаємодії з користувачем.
 - **Пов'язані елементи керування (Bound Controls):** Відображають дані з певного поля джерела записів (наприклад, текстове поле, яке відображає ПІБ зі стовпця "ПІБ").
 - **Непов'язані елементи керування (Unbound Controls):** Не пов'язані з полем джерела даних. Використовуються для відображення інформації (написи, зображення), кнопок (виконати дію), обчислюваних значень (які обчислюються на формі за виразом).
 - **Обчислювані елементи керування:** Непов'язані елементи, що відображають результат обчислення за виразом, введеним у властивості "Джерело елемента керування". Наприклад, $= [\text{Кількість}] * [\text{Ціна}]$ для розрахунку суми в рядку форми.
- **Режими роботи форми:**
 - **Режим форми (Form View):** Основний режим для кінцевих користувачів. Дозволяє переглядати, вводити та редагувати дані.
 - **Режим макета (Layout View):** Дозволяє змінювати дизайн форми (розмір, положення елементів керування) безпосередньо під час перегляду реальних даних. Зручно для швидкого налаштування вигляду.
 - **Режим конструктора (Design View):** Надає повний контроль над структурою форми. Дозволяє додавати/видаляти розділи форми (Заголовок форми, Область даних, Примітка форми), додавати елементи керування, налаштовувати їх властивості, додавати код VBA.

- **Властивості елементів керування та форми:** Кожен елемент керування та сама форма мають великий набір властивостей, що визначають їхній вигляд (колір, шрифт, розмір, видимість), поведінку (доступність, можливість редагування) та події (що має статися при натисканні кнопки, зміні значення поля тощо). Доступні через вікно "Відомість властивостей" (Property Sheet), що відкривається кнопкою на стрічці або правою кнопкою миші.

3. Створення звітів для представлення даних

Звіти в Access – це об'єкти, призначені для представлення даних у форматovanому вигляді, переважно для друку або перегляду на екрані. Вони дозволяють групувати дані, обчислювати підсумки, створювати підсумкові таблиці та представляти інформацію у професійному вигляді. Дані у звітах зазвичай не редагуються.

- **Призначення звітів:**

- Форматований вивід даних для друку (рахунки, накладні, звіти про продажі).
- Узагальнення даних (підрахунок сум, середніх значень, кількості за групами).
- Представлення даних у вигляді списків або таблиць з красивим оформленням.
- Створення етикеток, поштових конвертів.

- **Способи створення звітів (вкладка "Створення" -> група "Звіти"):**

- **Звіт (Report):** Швидке створення простого звіту на основі вибраної таблиці або запиту.
- **Майстер звітів (Report Wizard):** Найпопулярніший спосіб. Покроковий інструмент, що дозволяє вибрати поля, визначити рівні групування (наприклад, за менеджером, за датою), визначити поля для сортування, вибрати макет та орієнтацію звіту, вибрати стиль оформлення та вказати ім'я звіту.
- **Пустий звіт (Blank Report):** Створює новий пустий звіт у режимі макета, де ви вручну додаєте потрібні поля та елементи керування.
- **Конструктор звітів (Report Design):** Надає повний контроль над структурою та дизайном звіту. Дозволяє вручну додавати елементи керування, налаштовувати їх властивості, використовувати розділи звіту.
- **Етикетки (Labels):** Спеціальний майстер для створення звітів у форматі поштових або інших етикеток на основі стандартних або користувацьких розмірів.

- **Джерело даних звіту (Record Source):** Як і форми, звіти базуються на **таблиці** або **запиті**. Використання запиту дозволяє включити до звіту лише потрібні дані, які вже відфільтровані, відсортовані або містять необхідні обчислювані поля. Це рекомендований підхід.

- **Структура звіту (в режимі Конструктора):** Звіт розділений на **розділи**, які визначають, коли і як відображаються елементи керування:
 - **Заголовок звіту (Report Header):** Відображається один раз на початку звіту (на першій сторінці). Використовується для розміщення загальної назви звіту, логотипу.
 - **Верхній колонтитул сторінки (Page Header):** Відображається вгорі кожної сторінки. Використовується для розміщення заголовків стовпців даних, назви звіту (повторюється), номера сторінки, дати.
 - **Заголовок групи (Group Header):** З'являється на початку кожної нової групи записів (якщо у звіті визначено групування). Використовується для відображення значення поля групування (наприклад, ім'я менеджера) та, можливо, проміжних підсумків перед групою.
 - **Область даних (Detail Section):** Відображається один раз для *кожного запису* джерела даних, що відповідає умовам запити/таблиці. Це основна частина звіту, де виводяться дані окремих полів.
 - **Примітка групи (Group Footer):** З'являється в кінці кожної групи записів. Використовується для відображення проміжних підсумків для цієї групи (наприклад, сума продажів цього менеджера).
 - **Нижній колонтитул сторінки (Page Footer):** Відображається внизу кожної сторінки. Використовується для номера сторінки, дати, приміток.
 - **Примітка звіту (Report Footer):** Відображається один раз в кінці звіту (як правило, на останній сторінці, під останнім нижнім колонтитулом сторінки). Використовується для відображення загальних підсумків для всього звіту.
- **Групування та сортування:** Визначаються при створенні звіту Майстром або вручну через панель "Групування, сортування та підсумки" в режимі конструктора/макета. Групування дозволяє об'єднувати записи з однаковим значенням у полі групування.
- **Обчислення в звітах:** Можна додавати текстові поля з виразами для обчислення підсумків. Наприклад, поле з виразом `=Sum([ЗагальнаВартість])` у Примітці групи "Менеджер" обчислить суму загальної вартості для всіх записів у цій групі. Функції Sum, Avg, Count, Max, Min можуть використовуватися з посиланнями на поля. Загальні підсумки обчислюються в Примітці звіту.
- **Режими роботи звіту:**
 - **Режим звіту (Report View):** Інтерактивний попередній перегляд звіту на екрані. Дозволяє швидко переглядати дані, але може виглядати дещо інакше, ніж при друці.
 - **Попередній перегляд (Print Preview):** Показує, як звіт виглядатиме на друці. Точне відображення сторінок, полів, колонтитулів.

- **Режим макета (Layout View):** Дозволяє змінювати дизайн звіту під час перегляду реальних даних. Зручно для налаштування ширини стовпців, вирівнювання, форматування.
- **Режим конструктора (Design View):** Надає повний контроль над структурою звіту та властивостями всіх елементів керування та розділів.

Таким чином, Access надає набір інтегрованих об'єктів, які взаємодіють один з одним: таблиці зберігають сирі дані, запити їх обробляють, форми надають зручний інтерфейс для вводу/редагування, а звіти формують та узагальнюють дані для представлення. Ефективне проектування бази даних в Access включає не тільки створення добре нормалізованих таблиць, але й розробку відповідних запитів, форм та звітів.

ТЕМА 2.7. ОСНОВИ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Мета лекції: Розкрити фундаментальні поняття інформаційної безпеки, ознайомити з основними типами загроз та їхніми проявами, а також вивчити ключові технічні та організаційні методи і засоби захисту інформаційних активів, включаючи роль політики інформаційної безпеки.

План:

1. Поняття та загрози інформаційній безпеці.
2. Принципи та модель інформаційної безпеки.
3. Методи захисту інформації (технічні та організаційні).
4. Політика інформаційної безпеки.
5. Управління ризиками в інформаційній безпеці.

1. Поняття та загрози інформаційній безпеці

Інформаційна безпека (ІБ) – це багатогранна дисципліна, що займається захистом інформації та інформаційних систем від несанкціонованого доступу, використання, розголошення, порушення, зміни або знищення. Вона спрямована на мінімізацію ризиків, пов'язаних із загрозами безпеці, та забезпечення безперервності бізнес-процесів.

Основною моделлю, що визначає фундаментальні властивості інформації, які має захищати ІБ, є **триада CIA (Confidentiality, Integrity, Availability)**:

- **Конфіденційність (Confidentiality):** Забезпечення того, що інформація доступна лише авторизованим особам. Це означає захист даних від несанкціонованого перегляду або розголошення. Приклади порушення

конфіденційності: витік даних клієнтів, перехоплення мережевого трафіку, несанкціонований доступ до секретних документів.

- **Цілісність (Integrity):** Гарантія точності, повноти та достовірності інформації, а також методів її обробки. Це захист від несанкціонованої зміни або знищення даних. Приклади порушення цілісності: модифікація фінансових звітів, пошкодження файлів шкідливим ПЗ, некоректне внесення змін у базу даних.
- **Доступність (Availability):** Забезпечення можливості доступу до інформації та пов'язаних з нею ресурсів (систем, мереж) авторизованими користувачами тоді, коли це необхідно. Приклади порушення доступності: DoS/DDoS-атаки, збої обладнання, стихійні лиха, атаки програм-вимагачів, що блокують доступ до даних.

Іноді до тріади CIA додають інші принципи, формуючи розширені моделі, наприклад, CIAA (з додаванням Аутентичності - Authenticity) або Parkerian Hexad (додаючи Володіння - Possession, та Корисність - Utility). Однак CIA є базовою та найчастіше використовуваною моделлю.

Загрози інформаційній безпеці – це будь-які обставини або події, які можуть призвести до небажаних наслідків для інформаційних активів, використовуючи їхні вразливості. Загрози можуть бути випадковими (збій обладнання, помилка оператора) або навмисними (зловмисні дії).

Основні типи навмисних загроз:

- **Шкідливе програмне забезпечення (Malware):** Широка категорія програм, створених зі зловмисними цілями.
 - **Віруси:** Програми, що можуть розмножуватися та поширюватися, вбудовуючи свій код в інші програми.
 - **Хробаки (Worms):** Самостійні програми, що розмножуються та поширюються через мережі без втручання користувача.
 - **Троянські програми (Trojans):** Програми, що маскуються під корисне ПЗ, але виконують зловмисні дії (наприклад, надають віддалений доступ).
 - **Шпигунське ПЗ (Spyware):** Збирає інформацію про користувача (натискання клавіш, відвідувані сайти) без його згоди.
 - **Програми-вимагачі (Ransomware):** Шифрують дані користувача або блокують доступ до системи, вимагаючи викуп за відновлення доступу.
 - **Рекламне ПЗ (Adware):** Показує небажану рекламу, часто збирає дані про користувача.
- **Фішинг (Phishing):** Метод соціальної інженерії, заснований на масовому розсиланні електронних листів або повідомлень, що імітують офіційні джерела (банки, платіжні системи, відомі компанії), з метою виманити конфіденційну інформацію.

- **Spear Phishing:** Цільовий фішинг, спрямований на конкретну особу або невелику групу осіб, з використанням персоналізованої інформації для підвищення довіри.
 - **Whale Phishing (Whaling):** Фішинг, спрямований на високопоставлених осіб в організації ("велику рибу").
 - **Smishing:** Фішинг через SMS-повідомлення.
 - **Vishing:** Фішинг через голосові дзвінки.
- **Соціальна інженерія:** Використання психологічних маніпуляцій для обману людей та отримання доступу до інформації чи систем. Націлена на "найслабшу ланку" – людину.
 - **Pretexting:** Створення вигаданої ситуації або сценарію, щоб змусити жертву надати інформацію.
 - **Baiting:** Використання приманки (наприклад, зараженого USB-накопичувача з інтригуючою назвою файлу), щоб змусити жертву виконати дію.
 - **Quid Pro Quo:** Пропозиція чогось (наприклад, "технічної допомоги") в обмін на інформацію (наприклад, дані облікового запису).
 - **Tailgating / Piggybacking:** Несанкціоноване слідування за авторизованою особою для отримання фізичного доступу до захищеної зони.
- **Атаки на відмову в обслуговуванні (DoS - Denial of Service / DDoS - Distributed Denial of Service):** Спрямовані на виведення з ладу сервера, сервісу або мережевого каналу шляхом надсилання величезної кількості запитів або некоректних даних. DDoS атаки здійснюються з багатьох джерел одночасно.
- **Несанкціонований доступ (Unauthorized Access):** Отримання доступу до ресурсів (даних, систем, мереж) без належних дозволів. Це може бути здійснено шляхом підбору пароля, використання вразливостей ПЗ, отримання доступу через скомпрометований обліковий запис.
- **Витік даних (Data Breach):** Розкриття або втрата конфіденційної, приватної або захищеної інформації особам, які не мають права доступу до неї. Може бути результатом хакерських атак, помилок співробітників, втрати пристроїв.
- **Інтернет-шахрайство (Online Fraud):** Широкий спектр шахрайських дій в Інтернеті з метою отримання фінансової вигоди (наприклад, шахрайство з електронними платежами, фейкові онлайн-магазини, інвестиційні афери).
- **Внутрішні загрози (Insider Threats):** Загрози, що походять від співробітників, підрядників або інших осіб, які мають легітимний доступ до систем та даних. Можуть бути навмисними (зловмисні дії) або випадковими (помилки, недбалість).

2. Принципи та модель інформаційної безпеки

Для ефективного захисту інформації необхідно дотримуватися певних принципів та будувати модель безпеки.

- **Принципи інформаційної безпеки:**

- **Комплексність:** Захист має охоплювати всі компоненти інформаційної системи (обладнання, ПЗ, дані, мережі, користувачі) та використовувати сукупність різних методів і засобів захисту (технічних, організаційних, правових).
 - **Безперервність:** Захист повинен функціонувати постійно на всіх етапах життєвого циклу інформації (створення, зберігання, обробка, передача, знищення) та функціонування системи.
 - **Розумна достатність:** Рівень захисту та витрати на його забезпечення мають бути адекватними актуальним загрозам та цінності інформації, що захищається. Не варто будувати надто дорогі та складні системи захисту для низькоцінної інформації.
 - **Персональна відповідальність:** Кожен користувач системи несе відповідальність за свої дії відповідно до наданих йому повноважень та встановлених правил безпеки.
 - **Мінімізація повноважень (Principle of Least Privilege):** Кожному користувачеві та системному процесу надається мінімально необхідний набір прав доступу до ресурсів, достатній лише для виконання їхніх функцій. Це обмежує можливу шкоду від компрометації облікового запису або помилки.
 - **Гнучкість (Flexibility):** Система захисту має бути спроможною адаптуватися до змін у системі, що захищається, та до появи нових загроз.
 - **Обов'язковість контролю (Accountability):** У системі мають бути реалізовані механізми реєстрації дій користувачів та системних подій (журнали аудиту), що дозволяють відстежити, хто і які дії виконував.
- **Модель безпеки:** Визначення того, які саме правила доступу застосовуються в системі.
 - **Дискреційна модель (Discretionary Access Control - DAC):** Доступ до ресурсу визначається власником цього ресурсу. Власник може самостійно надавати або відкликати права доступу іншим користувачам (наприклад, як у файлових системах Windows, де власник папки може встановлювати дозволи для інших користувачів).
 - **Мандатна модель (Mandatory Access Control - MAC):** Доступ визначається не власником ресурсу, а централізованим адміністратором безпеки на основі міток безпеки (рівнів секретності, категорій). Кожен суб'єкт (користувач, процес) та об'єкт (файл, ресурс) мають свої мітки. Доступ дозволяється лише якщо мітки суб'єкта та об'єкта відповідають встановленим правилам (наприклад, користувач з рівнем "Секретно" може читати документи "Секретно" або "Для службового користування", але не може записувати дані з нижчим

рівнем секретності в документи з вищим рівнем). Використовується у системах з високими вимогами до безпеки (військові, державні).

- **Рольова модель (Role-Based Access Control - RBAC):** Доступ надається не конкретним користувачам, а ролям. Користувачеві призначається одна або кілька ролей, і він отримує права доступу, асоційовані з цими ролями (наприклад, роль "Менеджер" має права на перегляд звітів, роль "Бухгалтер" – на редагування фінансових даних). Це значно спрощує управління доступом у великих системах.

3. Методи захисту інформації (технічні та організаційні)

Ефективна система інформаційної безпеки поєднує технічні засоби захисту, що реалізуються за допомогою програмного та апаратного забезпечення, та організаційні заходи, пов'язані з управлінням, процедурами та людським фактором.

Технічні методи та засоби:

- **Антивірусне та антишкідливе ПЗ:** Програми для виявлення, блокування, лікування та видалення шкідливого програмного забезпечення. Працюють на основі сигнатурного аналізу (порівняння з відомими зразками) та евристичного/поведінкового аналізу (виявлення підозрілих дій). **Важливо:** регулярне оновлення антивірусних баз та самого програмного забезпечення, а також періодичне повне сканування системи.
- **Брандмауери (Фаєрволи) та Міжмережеві екрани:** Програмні або апаратні засоби, що встановлюються на межі мережі або на окремих комп'ютерах. Вони контролюють мережевий трафік, аналізуючи пакети даних за певними правилами (джерело, призначення, порт, протокол) та дозволяючи або забороняючи їх проходження. Захищають від несанкціонованого доступу ззовні та контролюють вихідний трафік.
- **Шифрування (Encryption):** Процес перетворення даних з читабельного формату (відкритий текст) у нечитабельний (шифротекст) за допомогою криптографічного алгоритму та ключа. Дешифрування повертає дані у початковий вигляд за допомогою відповідного ключа.
 - **Шифрування даних під час зберігання (Data at Rest):** Шифрування файлів на дисках, базах даних. Наприклад, шифрування всього системного диска (BitLocker у Windows, FileVault у macOS) захищає дані у випадку втрати або крадіжки пристрою.
 - **Шифрування даних під час передачі (Data in Transit):** Захист даних, що передаються через мережі. Використовується в протоколах HTTPS (для веб-з'єднань), SSL/TLS (для захищених з'єднань), VPN (Virtual Private Network - створення захищеного "тунелю" через незахищену мережу, наприклад, Інтернет).
- **Системи виявлення вторгнень (IDS - Intrusion Detection Systems) та Системи запобігання вторгненням (IPS - Intrusion Prevention Systems):**

- **IDS:** Моніторять мережеву активність або файли журналів на предмет підозрілої або шкідливої активності, сигналізуючи про виявлені загрози. Не блокують трафік.
- **IPS:** Виконують ті самі функції, що й IDS, але також можуть автоматично вживати заходів для блокування або припинення виявленої шкідливої активності.
- **Системи керування доступом:**
 - **Аутентифікація (Authentication):** Процес перевірки особистості користувача або іншого суб'єкта. Найпоширеніші фактори аутентифікації:
 - Щось, що користувач знає (пароль, PIN-код).
 - Щось, що користувач має (смарт-картка, токен безпеки, мобільний телефон для SMS-коду).
 - Щось, чим користувач є (біометрія – відбиток пальця, сканування райдужної оболонки, розпізнавання обличчя).
 - **Багатофакторна автентифікація (MFA - Multi-Factor Authentication) / Двофакторна автентифікація (2FA - Two-Factor Authentication):** Вимога надати щонайменше два різні фактори аутентифікації для підтвердження особи. Значно підвищує безпеку облікових записів, оскільки компрометація одного фактора (наприклад, викрадення пароля) не дає зловмиснику повного доступу.
 - **Авторизація (Authorization):** Процес надання або відмови в доступі авторизованому користувачеві до певних ресурсів або функцій відповідно до його прав та повноважень (див. моделі безпеки: дискреційна, мандатна, рольова).
- **Системи резервного копіювання та відновлення даних:** Регулярне створення копій важливих даних на окремих носіях або в хмарних сховищах. Це дозволяє відновити втрачену або пошкоджену інформацію у випадку збоїв, атак програм-вимагачів, помилок користувача або стихійних лих. **Важливо:** Регулярно тестувати процедури відновлення, щоб бути впевненим у працездатності резервних копій. Дотримуватися правила "3-2-1": 3 копії даних, на 2 різних типах носіїв, 1 з яких зберігається віддалено.
- **Системи моніторингу та аудиту безпеки (SIEM - Security Information and Event Management):** Збирають та аналізують журнали подій безпеки з різних систем та пристроїв для виявлення підозрілої активності та інцидентів безпеки.

Організаційні методи та заходи:

- **Політика інформаційної безпеки (ПІБ):** Набір документів, що визначають правила, процедури та стандарти безпечної роботи (див. п. 4).
- **Навчання та підвищення обізнаності користувачів (Security Awareness Training):** Проведення регулярних тренінгів для співробітників з питань інформаційної безпеки, ознайомлення з актуальними загрозами (фішинг,

соціальна інженерія), правилами створення надійних паролів, безпечного використання Інтернету, поводження з конфіденційною інформацією. Людський фактор є однією з найслабших ланок у системі безпеки, і його важливо зміцнювати.

- **Планування реагування на інциденти безпеки (Incident Response Plan):** Розробка чітких процедур дій у випадку виникнення інциденту безпеки (наприклад, виявлення шкідливого ПЗ, витік даних, DoS-атака). План має включати кроки з виявлення, стримування, ліквідації наслідків інциденту та відновлення нормальної роботи.
- **Розмежування обов'язків (Separation of Duties):** Розподіл критично важливих завдань між кількома співробітниками, щоб жодна особа не мала повного контролю над усім процесом, що може призвести до зловживань або помилок.
- **Управління патчами (Patch Management):** Процес регулярного встановлення оновлень безпеки (патчів) для операційних систем, програмного забезпечення та обладнання. Виробники ПЗ випускають патчі для закриття виявлених вразливостей, якими можуть скористатися зловмисники. Своєчасне встановлення патчів є критично важливим.
- **Управління вразливостями (Vulnerability Management):** Процес ідентифікації, оцінки та усунення вразливостей в інформаційних системах. Включає сканування на наявність вразливостей та їхнє пріоритетне усунення.
- **Фізична безпека:** Захист серверних кімнат, офісних приміщень, комп'ютерів та мережевого обладнання від несанкціонованого фізичного доступу (контроль доступу, відеоспостереження, охорона).
- **Планування безперервності бізнесу (Business Continuity Planning) та аварійного відновлення (Disaster Recovery Planning):** Розробка планів дій для забезпечення функціонування критично важливих бізнес-процесів та відновлення роботи ІТ-систем після серйозних збоїв, аварій або катастроф.

4. Політика інформаційної безпеки

Політика інформаційної безпеки (ПІБ) – це наріжний камінь системи ІБ будь-якої організації. Це формалізований, затверджений керівництвом документ, що встановлює загальні правила, принципи та обов'язки щодо захисту інформаційних активів. ПІБ є основою для побудови всіх інших процедур та інструкцій з безпеки.

- **Призначення та роль ПІБ:**

- Встановлення єдиних вимог та стандартів безпеки для всіх співробітників, підрядників та користувачів, які мають доступ до інформаційних ресурсів організації.
- Регламентація процесів управління інформаційною безпекою.
- Підвищення обізнаності персоналу щодо їхніх обов'язків у сфері ІБ.
- Забезпечення відповідності вимогам законодавства, регуляторних органів, стандартів та договірних зобов'язань.

- Слугує основою для прийняття рішень у ситуаціях, пов'язаних з безпекою.
- Визначає відповідальність за порушення правил безпеки.
- **Принципи розробки та зміст ПІБ:**
 - **Чіткість та зрозумілість:** Повинна бути написана мовою, зрозумілою для всіх співробітників.
 - **Повнота:** Охоплювати всі критичні аспекти безпеки.
 - **Актуальність:** Регулярно переглядатися та оновлюватися відповідно до змін у технологіях, загрозах, структурі організації та законодавстві.
 - **Підтримка керівництва:** Мати офіційне затвердження та підтримку вищого керівництва організації.
- **Основні компоненти типової ПІБ (або набору пов'язаних документів):**
 - **Визначення цілей та завдань ІБ:** Що саме захищається і навіщо.
 - **Сфера застосування:** На кого поширюється дія політики (усі співробітники, підрядники, системи).
 - **Визначення ролей та відповідальності:** Хто відповідає за розробку, впровадження, моніторинг та виконання політики.
 - **Правила використання інформаційних активів:** Класифікація інформації, правила поводження з конфіденційною інформацією, правила резервного копіювання.
 - **Політика керування доступом:** Правила надання/відкликання прав доступу, політика паролів (вимоги до складності, терміну дії), правила використання багатофакторної автентифікації.
 - **Правила використання мережевих ресурсів та Інтернету:** Дозволені та заборонені дії, правила використання електронної пошти, безпека бездротових мереж.
 - **Захист від шкідливого програмного забезпечення:** Правила використання антивірусного ПЗ, дії у випадку виявлення зараження.
 - **Безпека обладнання:** Правила використання особистих пристроїв (BYOD - Bring Your Own Device), безпека мобільних пристроїв, поводження зі знімними носіями інформації.
 - **Реагування на інциденти безпеки:** Загальні процедури повідомлення та реагування на інциденти.
 - **Навчання та обізнаність:** Вимоги до обов'язкового навчання персоналу.
 - **Контроль та аудит:** Як перевіряється виконання політики.
 - **Відповідальність за порушення:** Наслідки недотримання правил політики.

ПІБ повинна бути доступною для всіх співробітників, і кожен співробітник має бути ознайомлений з її положеннями під підпис. Вона є живим документом і потребує періодичного перегляду та актуалізації.

5. Управління ризиками в інформаційній безпеці

Забезпечення інформаційної безпеки – це по суті управління ризиками. **Ризик** в ІБ – це ймовірність того, що певна загроза використає певну вразливість і завдасть шкоди інформаційним активам. **Управління ризиками** – це систематичний процес ідентифікації, оцінки та обробки ризиків з метою зменшення їхнього впливу до прийняттого рівня.

Етапи управління ризиками:

- **Ідентифікація ризиків:** Виявлення потенційних загроз, вразливостей та інформаційних активів.
- **Оцінка ризиків:** Визначення ймовірності реалізації кожної загрози та потенційного впливу (шкоди) у випадку її реалізації. Ризик часто розраховується як Добуток Ймовірності на Вплив ($Risk = Probability \times Impact$).
- **Обробка ризиків (Risk Treatment):** Прийняття рішення щодо кожного ідентифікованого ризику:
 - **Уникнення ризику:** Припинення діяльності, що генерує ризик.
 - **Зменшення ризику (Mitigation):** Впровадження заходів безпеки (контролів) для зниження ймовірності або впливу ризику (наприклад, встановлення антивірусу, шифрування даних, навчання персоналу).
 - **Передача ризику (Transfer):** Перекладання відповідальності за ризик на третю сторону (наприклад, страхування кіберризиків, передача даних на обробку провайдеру хмарних послуг з відповідними гарантіями безпеки).
 - **Прийняття ризику (Acceptance):** Усвідомлене рішення не вживати заходів щодо ризику, якщо його рівень вважається прийнятним або вартість заходів перевищує потенційну шкоду.
- **Моніторинг та перегляд ризиків:** Регулярне відстеження ідентифікованих ризиків, оцінка ефективності впроваджених заходів та ідентифікація нових ризиків.

Управління ризиками є безперервним процесом, який допомагає організації приймати обґрунтовані рішення щодо інвестицій в інформаційну безпеку та пріоритезації заходів захисту.

ТЕМА 2.8. ПРАВОВІ ТА ЕТИЧНІ АСПЕКТИ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Мета: Ознайомити здобувачів вищої освіти з ключовими правовими нормами, що регулюють використання інформаційних технологій в Україні, включаючи авторське право на програмне забезпечення, різні типи ліцензій,

законодавство про захист персональних даних та кримінальну відповідальність за кіберзлочини, а також розглянути основні етичні принципи, що стосуються професійної діяльності в ІТ-сфері.

План:

1. Авторське право на програмне забезпечення.
2. Ліцензії на програмне забезпечення.
3. Захист персональних даних (українське законодавство та GDPR).
4. Кримінальна відповідальність за кіберзлочини в Україні.
5. Етика в інформаційних технологіях.

Використання інформаційних технологій, їх розробка та розповсюдження відбувається не у вакуумі, а в чітко визначеному правовому полі та в контексті певних етичних норм. Розуміння цих аспектів є важливим для кожного, хто працює або активно користується ІТ.

1. Авторське право на програмне забезпечення

Авторське право – це юридичний інститут, що надає авторам (творцям) оригінальних творів (літературних, мистецьких, наукових) виключні права на використання та розповсюдження своїх творів. Комп'ютерні програми в Україні та більшості країн світу охороняються саме авторським правом, а не патентним правом (як винаходи) або торговельними марками (як бренди).

- **Правове регулювання в Україні:** Основний закон – Закон України "Про авторське право і суміжні права". Чинна редакція була суттєво оновлена у грудні 2022 року. Цей закон чітко визначає комп'ютерні програми як об'єкти авторського права, прирівнюючи їх до літературних творів. Посилання на актуальний текст закону: <https://zakon.rada.gov.ua/laws/show/2811-20#Text>.
- **Об'єкт охорони авторським правом:** Охороняється **форма вираження** комп'ютерної програми. Це, перш за все, її **вихідний код (Source Code)** та **об'єктний код (Object Code)**, за умови, що вони є оригінальними, тобто результатом власної інтелектуальної творчості автора. Авторське право захищає *спосіб вираження ідеї*, а не саму ідею.
- **Що НЕ охороняється авторським правом на ПЗ:** Авторським правом на комп'ютерну програму **не охороняються**:
 - Ідеї та принципи, що лежать в основі програми.
 - Її функціональність.
 - Алгоритми, що використовуються.
 - Логічні схеми.
 - Мови програмування.
 - Концепції користувацького інтерфейсу, якщо вони не є оригінальною графічною роботою.

- Формат файлів даних.
- Технічні засоби та системи, в яких використовується програма. Це означає, що інший розробник може створити програму, яка виконує ті самі функції або використовує той самий алгоритм, але написавши власний оригінальний код.
- **Права автора комп'ютерної програми:** Автор має два види прав:
 - **Особисті немайнові права:** Не можуть бути передані (відчужені) іншій особі і діють безстроково. Включають:
 - Право вимагати визнання свого авторства (право на ім'я).
 - Право забороняти згадування свого імені, якщо автор бажає залишитися анонімним (право на псевдонім).
 - Право обирати псевдонім.
 - Право на недоторканність твору (забороняти спотворення, калічення твору, що може зашкодити честі та репутації автора).
 - **Майнові права:** Надають автору виключне право використовувати твір будь-яким способом та виключне право дозволяти або забороняти таке використання іншим особам. Ці права є відчужуваними (можуть бути передані за договором) та мають обмежений строк дії. Основні способи використання ПЗ, на які поширюються майнові права:
 - Відтворення (копіювання) програми.
 - Розповсюдження екземплярів програми (продаж, оренда, передача).
 - Імпорт екземплярів програми.
 - Публічний показ або публічне сповіщення.
 - Переробка, адаптація, аранжування, інші подібні зміни програми.
 - Доведення програми до загального відома публіки таким чином, що її представники можуть отримати доступ до твору з будь-якого місця і в будь-який час за власним вибором (надання доступу онлайн).
 - **Термін дії майнових прав:** За загальним правилом, майнові права на твір діють протягом життя автора та **70 років після його смерті**. Для творів, створених у співавторстві, 70 років відраховуються після смерті останнього співавтора.
- **Службовий твір:** Якщо комп'ютерна програма створена працівником під час виконання своїх трудових обов'язків за трудовим договором, майнові права на неї, як правило, належать **роботодавцю**, якщо інше не встановлено договором. При цьому особисті немайнові права завжди залишаються за працівником-автором. Працівник, який створив службовий твір, має право на винагороду за його використання (якщо інше не передбачено договором). Аналогічні правила застосовуються до творів, створених за гіг-контрактами у рамках правового режиму Дія.City.

- **Виникнення та реєстрація авторського права:** Авторське право виникає **автоматично** з моменту створення твору (з моменту надання йому об'єктивної форми) і **не потребує обов'язкової реєстрації**, спеціального оформлення чи дотримання будь-яких формальностей. Проте, в Україні можлива **добровільна державна реєстрація** авторського права на комп'ютерну програму. Свідоцтво про реєстрацію може слугувати додатковим доказом авторства та дати створення твору у разі виникнення судових спорів.
- **Порушення авторського права (Піратство ПЗ):** Незаконне використання комп'ютерної програми (наприклад, копіювання, встановлення на більшу кількість комп'ютерів, ніж дозволено ліцензією, розповсюдження) без дозволу правовласника є порушенням авторського права і тягне за собою юридичну відповідальність (цивільну, адміністративну, кримінальну).

2. Ліцензії на програмне забезпечення

Оскільки авторське право надає правовласнику виключне право використовувати ПЗ, для легального використання програми іншими особами необхідний дозвіл від правовласника. Такий дозвіл надається у формі **ліцензії на програмне забезпечення**. Ліцензія – це, по суті, договір між правовласником (ліцензіаром) та користувачем (ліцензіатом), що визначає умови, на яких користувач може використовувати ПЗ.

- **Ключова ідея:** Користувач програмного забезпечення, як правило, **не купує саму програму** у власність (як річ), а отримує **право використовувати її** відповідно до умов ліцензії.
- **Типи ліцензій:** Існує безліч видів ліцензій, які можна класифікувати за різними критеріями:
 - **За способом оплати / доступу:**
 - **Комерційні (Commercial / Proprietary Software):** Програми, за використання яких потрібно сплатити. Як правило, розповсюджуються за пропрієтарними ліцензіями.
 - **Freeware (Безкоштовне ПЗ):** Дозволяє використовувати програму безкоштовно протягом необмеженого часу. Проте правовласник зазвичай зберігає всі інші права, обмежує модифікацію та розповсюдження, вихідний код закритий.
 - **Shareware (Умовно-безкоштовне ПЗ):** Надається для безкоштовного ознайомлення (пробний період) або з обмеженим функціоналом. Для повноцінного або подальшого використання необхідно придбати ліцензію.
 - **Adware (Ad-supported Software):** Безкоштовне ПЗ, використання якого "оплачується" переглядом реклами, інтегрованої в програму.
 - **За доступом до вихідного коду та правами на модифікацію / розповсюдження:**

- **Пропрієтарні (Proprietary / Closed Source):** Надають користувачеві лише право використовувати програму (зазвичай, на одному або кількох пристроях). Вихідний код є закритим, користувачу забороняється копіювати, модифікувати, декомпілювати, розповсюджувати програму без прямого дозволу правовласника. Більшість комерційного ПЗ розповсюджується за такими ліцензіями (наприклад, Microsoft Windows, Adobe Creative Cloud).
- **Відкриті (Open Source):** Надають користувачам ширші права, включаючи доступ до вихідного коду. Дозволяють використовувати, вивчати, змінювати та розповсюджувати програму (як оригінальну, так і модифіковану версію) на певних умовах. Існує багато різних ліцензій відкритого коду з різними умовами.
- **Дозвільні (Permissive Licenses):** Накладають мінімальні обмеження. Зазвичай вимагають лише збереження інформації про авторство та текст ліцензії. Дозволяють використовувати код у пропрієтарних проектах. Приклади: MIT License, BSD License, Apache License.
- **Копілефт (Copyleft Licenses):** Більш обмежувальні. Вимагають, щоб будь-які похідні роботи (модифіковані версії або програми, що використовують код під такою ліцензією) також розповсюджувалися під цією ж ліцензією або сумісною з нею. Це забезпечує "вірусне" поширення вільного статусу коду. Приклади: GNU General Public License (GPL), GNU Lesser General Public License (LGPL).
- **Ліцензія кінцевого користувача (End-User License Agreement - EULA):** Це типова форма ліцензійного договору, що супроводжує пропрієтарне ПЗ. Користувач, як правило, приймає умови EULA під час встановлення програми. EULA деталізує права та обмеження для кінцевого користувача.

Розуміння умов ліцензії є обов'язковим для легального використання будь-якого програмного забезпечення. Порушення умов ліцензії (наприклад, використання Freeware у комерційних цілях, якщо це заборонено, або розповсюдження ПЗ під пропрієтарною ліцензією) є порушенням авторського права.

3. Захист персональних даних

В епоху інформаційних технологій збирання, зберігання, обробка та передача персональних даних здійснюються у величезних масштабах. Захист цих даних та забезпечення права людини на приватність є критично важливим і регулюється спеціальним законодавством.

- **Законодавство України:** Основний нормативний акт – **Закон України "Про захист персональних даних"**. Посилання: <https://zakon.rada.gov.ua/laws/show/2297-17#Text>. Цей закон визначає правові основи захисту і обробки персональних даних.
 - **Персональні дані:** Будь-які відомості або сукупність відомостей про фізичну особу, яка ідентифікована або може бути конкретно ідентифікована. Приклади: прізвище, ім'я, по батькові, адреса, номер телефону, електронна пошта, дата народження, ідентифікаційний код, дані про місцезнаходження, IP-адреса, дані про стан здоров'я, біометричні дані.
 - **Суб'єкт персональних даних:** Фізична особа, стосовно якої здійснюється обробка її персональних даних.
 - **Володілець персональних даних:** Фізична або юридична особа, яка визначає мету та способи обробки персональних даних.
 - **Розпорядник персональних даних:** Фізична або юридична особа, якій володільцем або законом надано право обробляти ці дані від імені володільця (наприклад, хмарний провайдер).
 - **Принципи обробки персональних даних (відповідно до Закону України):**
 - **Законність:** Обробка здійснюється лише на законних підставах.
 - **Визначена мета:** Дані збираються для конкретних, чітко визначених та законних цілей і не обробляються надмірно.
 - **Відповідність та адекватність:** Обсяг даних є мінімально необхідним та відповідає меті обробки.
 - **Точність:** Дані мають бути точними та актуальними.
 - **Обмеження строку зберігання:** Дані зберігаються не довше, ніж це необхідно для мети обробки.
 - **Цілісність та конфіденційність:** Забезпечення захисту даних від незаконної обробки, випадкової втрати, знищення, пошкодження, несанкціонованого доступу.
 - **Підстави для обробки:** Обробка персональних даних дозволяється, як правило, за згодою суб'єкта персональних даних. Інші законні підстави включають виконання договору, здійснення повноважень, передбачених законом, захист життєво важливих інтересів суб'єкта, захист законних інтересів володільця (якщо це не порушує права суб'єкта).
 - **Чутливі персональні дані:** Закон виділяє "особливі категорії" даних (расове або етнічне походження, політичні, релігійні або світоглядні переконання, членство в політичних партіях та професійних спілках, стан здоров'я, статеве життя, біометричні та генетичні дані). Обробка таких даних, як правило, **заборонена**, за винятком окремих випадків,

чітко визначених законом (наприклад, явна згода суб'єкта, життєво важливі інтереси).

- **Права суб'єкта персональних даних:** Суб'єкт даних має значний перелік прав, включаючи: право знати про джерела збору та мету обробки своїх даних; право на доступ до своїх даних; право отримувати інформацію про умови надання доступу; право вимагати виправлення, зміни або знищення неточних або незаконно оброблених даних; право на заперечення проти обробки; право відкликати згоду на обробку; право на судовий захист. Доступ до своїх даних та інформації про їх обробку надається безоплатно.
- **Обов'язки власника даних:** Власник зобов'язаний вживати всіх необхідних заходів для забезпечення захисту персональних даних від незаконної обробки, повідомляти суб'єкта даних про його права, а також про третіх осіб, яким передаються його дані (якщо це вимагається законом або згодою).
- **Уповноважений ВРУ з прав людини:** В Україні функцію органу з контролю за дотриманням законодавства про захист персональних даних виконує Уповноважений Верховної Ради України з прав людини.
- **GDPR (General Data Protection Regulation): Загальний регламент про захист даних Європейського Союзу.** Набув чинності у 2018 році. Встановлює суворіші правила захисту даних порівняно з попередніми нормами ЄС і має **екстериторіальну дію**. Це означає, що GDPR може застосовуватися до компаній за межами ЄС (включаючи українські), якщо вони:
 - Пропонують товари або послуги резидентам ЄС (незалежно від того, чи вимагається оплата).
 - Моніторять поведінку осіб, які перебувають в ЄС (наприклад, відстежують користувачів веб-сайтів за допомогою файлів cookie або інших інструментів аналітики).
 - Мають представництво в ЄС.
 - **Ключові відмінності GDPR від попередніх норм (та часто від національного законодавства, що не гармонізоване з GDPR):** Більш жорсткі вимоги до отримання "явної згоди" на обробку даних; розширені права суб'єктів даних (право на забуття/стирання, право на переносимість даних, право на обмеження обробки); обов'язок повідомляти про витоки даних; вимога призначати відповідального за захист даних (Data Protection Officer - DPO) у певних випадках; суттєві штрафи за порушення (до 20 млн євро або 4% світового річного обороту). Українське законодавство поступово гармонізується з нормами GDPR.
- **Значення для ІТ-фахівців:** Розробники програмного забезпечення, веб-сайтів, мобільних додатків повинні враховувати вимоги законодавства про захист персональних даних при проектуванні систем (принцип Privacy by Design),

збирати лише мінімально необхідні дані, забезпечувати їхній належний захист та реалізовувати механізми, що дозволяють суб'єктам даних реалізувати свої права.

4. Кримінальна відповідальність за кіберзлочини

Кіберзлочини – це суспільно небезпечні діяння, що вчиняються у сфері використання електронно-обчислювальних машин (комп'ютерів), систем та комп'ютерних мереж і мереж електрозв'язку, та передбачені Кримінальним кодексом України (КК України) як злочини.

- **Розділ XVI КК України:** Більшість злочинів у сфері використання комп'ютерів та мереж зібрані у Розділі XVI "Кримінальні правопорушення у сфері використання електронно-обчислювальних машин (комп'ютерів), систем та комп'ютерних мереж і мереж електрозв'язку" (статті 361-363-1).
 - **Стаття 361: Несанкціоноване втручання в роботу інформаційних (автоматизованих), електронних комунікаційних, інформаційно-комунікаційних систем, електронних комунікаційних мереж.** Це, по суті, стаття, що охоплює "хакінг" – незаконне проникнення в системи, зміну маршрутизації інформації тощо.
 - **Стаття 361-1: Створення з метою протиправного використання, розповсюдження або збуту шкідливих програмних чи технічних засобів, а також їх розповсюдження або збут.** Відповідальність за створення, розповсюдження та збут вірусів, троянів, програм-шпигунів та інших шкідливих інструментів.
 - **Стаття 361-2: Несанкціоновані збут або розповсюдження інформації з обмеженим доступом, яка зберігається в електронно-обчислювальних машинах (комп'ютерах), автоматизованих системах, комп'ютерних мережах або на носіях такої інформації.** Відповідальність за незаконне розголошення або продаж конфіденційної інформації (наприклад, комерційної таємниці, банківської таємниці, персональних даних), отриманої з комп'ютерних систем.
 - **Стаття 362: Несанкціоновані дії з інформацією, яка оброблюється в електронно-обчислювальних машинах (комп'ютерах), автоматизованих системах, комп'ютерних мережах або зберігається на носіях такої інформації, вчинені особою, яка має право доступу до неї.** Це злочини, вчинені інсайдерами – особами, які мають легальний доступ, але перевищують свої повноваження або діють зі зловмисним наміром (наприклад, незаконне видалення, зміна, блокування або копіювання даних працівником).
 - **Стаття 363: Порушення правил експлуатації електронно-обчислювальних машин (комп'ютерів), автоматизованих систем,**

комп'ютерних мереж чи мереж електрозв'язку або правил захисту інформації, що вчинені особою, яка відповідає за їх експлуатацію, якщо це заподіяло значну шкоду. Відповідальність для системних адміністраторів, інженерів за порушення правил, що призвели до серйозних наслідків.

- **Стаття 363-1: Перешкоджання роботі електронно-обчислювальних машин (комп'ютерів), автоматизованих систем, комп'ютерних мереж чи мереж електрозв'язку шляхом масового розповсюдження повідомлень електрозв'язку, вчинене без попередньої згоди адресатів (спам).** Відповідальність за розповсюдження спаму, що перешкоджає роботі систем. Сюди також можуть підпадати деякі види DoS/DDoS-атак.
- **Інші статті КК України:** Деякі кіберзлочини можуть підпадати під дію інших статей Кримінального кодексу, наприклад:
 - **Стаття 190 (Шахрайство):** Частини 3 та 4 передбачають відповідальність за шахрайство, вчинене шляхом незаконних операцій з використанням електронно-обчислювальної техніки.
 - **Стаття 182 (Порушення недоторканності приватного життя):** Охоплює незаконне збирання, зберігання, використання, поширення конфіденційної інформації про особу без її згоди, в тому числі отриманої з використанням ІТ.
 - **Стаття 176 (Порушення авторського права і суміжних прав):** Кримінальна відповідальність за незаконне відтворення, розповсюдження творів (включаючи ПЗ), якщо це завдало значної матеріальної шкоди.
 - **Стаття 120 (Доведення до самогубства):** Може застосовуватися, якщо кібербулінг або інші онлайн-переслідування призвели до таких трагічних наслідків.
- **Відповідальність:** Залежно від конкретного складу злочину (статті), наявності обтяжуючих обставин (наприклад, повторність, вчинення групою осіб, заподіяння значної/великої/особливо великої шкоди), відповідальність може варіювати від штрафів, позбавлення права обіймати певні посади чи займатися певною діяльністю до обмеження волі або позбавлення волі на значні терміни.
- **Міжнародне співробітництво:** Україна є стороною Конвенції Ради Європи про кіберзлочинність (Будапештська конвенція), що сприяє міжнародному співробітництву у боротьбі з кіберзлочинністю та гармонізації національного законодавства.
- **Актуальність:** Законодавство у сфері кіберзлочинів постійно розвивається, реагуючи на появу нових видів злочинів та технологій.

5. Етика в інформаційних технологіях

Окрім правових норм, професійна діяльність в ІТ-сфері та використання технологій підпадають під дію **етичних принципів** – системи моральних стандартів та правил поведінки, що визначають, що є правильним або неправильним у контексті технологій. З огляду на значний вплив ІТ на суспільство, етичні питання є надзвичайно важливими.

- **Чому етика важлива в ІТ:**

- **Потужність технологій:** Технології можуть мати значний позитивний або негативний вплив на життя людей та суспільство.
- **Відсутність чіткого правового регулювання:** Право часто не встигає за швидким розвитком технологій, залишаючи "сірі зони", де етичні принципи стають основним орієнтиром.
- **Глобальність:** ІТ не мають кордонів, створюючи складні етичні питання, пов'язані з культурними відмінностями та різним законодавством.
- **Професійна відповідальність:** ІТ-фахівці мають професійну відповідальність перед своїми клієнтами, роботодавцями, колегами та суспільством загалом.

- **Основні етичні принципи в ІТ:** Багато професійних організацій (наприклад, ACM - Association for Computing Machinery) розробили кодекси етики для фахівців у сфері ІТ. Ці кодекси зазвичай включають наступні принципи:

- **Не нашкодь (Do No Harm):** Першочерговий принцип. Не використовувати комп'ютери, системи або дані для завдання шкоди іншим людям (фізичної, моральної, фінансової). Уникати дій, що можуть призвести до втрати даних, доступу до приватної інформації без згоди, створення шкідливого ПЗ.
- **Чесність та справедливість (Honesty and Fairness):** Бути чесним у всіх професійних відносинах. Не вводити в оману користувачів щодо можливостей або обмежень продуктів. Уникати дискримінації та несправедливості у розробці та застосуванні технологій (наприклад, уникати упередженості в алгоритмах ШІ).
- **Конфіденційність та приватність (Confidentiality and Privacy):** Поважати право користувачів на приватність. Захищати конфіденційну інформацію та персональні дані. Збирати та використовувати дані лише з належною згодою та для визначених цілей.
- **Повага до інтелектуальної власності (Respect for Intellectual Property):** Дотримуватися авторських прав, патентів, торговельних марок та ліцензійних угод. Не використовувати крадене або неліцензійне програмне забезпечення. Не привласнювати результати чужої праці.

- **Відповідальність (Accountability):** Брати на себе відповідальність за власну роботу та її можливі наслідки. Проектувати системи таким чином, щоб було зрозуміло, хто відповідає за певні дії або збої. Вчасно виправляти помилки у розробленому ПЗ.
- **Професійна компетентність (Professional Competence):** Підтримувати та постійно підвищувати свій рівень знань та навичок. Не братися за завдання, для виконання яких не вистачає компетенції, або заздалегідь повідомити про обмеження.
- **Якість продуктів та послуг (Quality of Products and Services):** Прагнути до створення високоякісних, надійних та безпечних продуктів і послуг.
- **Сприяння суспільному благу (Contributing to Society and Human Well-being):** Використовувати свої навички для створення технологій, що приносять користь суспільству та покращують життя людей.
- **Об'єктивність та прозорість (Objectivity and Transparency):** Бути об'єктивним у технічних оцінках. Надавати правдиву інформацію. Бути прозорим щодо використання технологій, особливо тих, що можуть мати значний вплив (наприклад, пояснення роботи алгоритмів ШІ, якщо це можливо).
- **Етичні дилеми:** ІТ-фахівці часто стикаються з етичними дилемами – ситуаціями, коли різні етичні принципи або інтереси конфліктують, і немає очевидно "правильного" рішення. Наприклад:
 - Чи варто розробляти ПЗ для стеження або цензури на замовлення уряду, навіть якщо це легально, але суперечить праву на приватність?
 - Як балансувати між збором даних для покращення сервісу та захистом приватності користувачів?
 - Що робити, якщо ви виявили вразливість у системі клієнта, яка може бути використана зловмисниками, але клієнт не хоче її виправляти?
 - Як забезпечити справедливість алгоритмів прийняття рішень (наприклад, при прийомі на роботу або видачі кредитів), щоб вони не дискримінували певні групи людей?

Вміння розпізнавати етичні проблеми, аналізувати їх з точки зору етичних принципів та приймати відповідальні рішення є невід'ємною частиною професіоналізму в сфері інформаційних технологій.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Word - допомога та навчання - Microsoft Support, <https://support.microsoft.com/uk-ua/word>
2. Допомога та навчання з Access - Microsoft Support, <https://support.microsoft.com/uk-ua/access>
3. Інформатика — Теорія — Блок-схеми алгоритму, <https://yevshan.com.ua/info/006/content/content3.html>
4. Лекція №1 - Кафедра програмного забезпечення, <https://pz.vntu.edu.ua/media/uploads/metod/multi/%D0%9F%D0%BE%D1%81%D1%96%D0%B1%D0%BD%D0%B8%D0%BA.pdf>
5. Комп'ютерні інформаційні технології - Запорізький національний університет, <https://files.znu.edu.ua/files/Bibliobooks/Inshi67/0049483.pdf>
6. Класифікація та загальна характеристика програмного забезпечення, операційна система, <https://www.kievoit.ippo.kubg.edu.ua/kievoit/2013/56/56.html>
7. Програмне забезпечення ПК, <https://www.kievoit.ippo.kubg.edu.ua/kievoit/2013/57/57.html>
8. Основи інформаційних технологій - Електронна бібліотека ІМЗО, https://lib.imzo.gov.ua/wa-data/public/site/books2/posibnyky-prof-tech/Osnovy_inform_tehnologiy.pdf
9. Вступ до інформатики | PPT - SlideShare, <https://www.slideshare.net/slideshow/ss-16268258/16268258>
10. Інформатика - Інформаційно-обчислювальний центр, дата последнего обращения: апреля 29, 2025, http://web.kpi.kharkov.ua/sp/wp-content/uploads/sites/95/2020/12/NK_Informatika_1.pdf

Навчальне видання

ІНФОРМАТИКА ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Конспект лекцій

Укладачі:

Тищенко Світлана Іванівна
Співак Вадим Вікторович

Формат 60x84 1/16. Ум. друк. арк. 7,0.

Наклад 50 прим. Зам. № _____

Надруковано у видавничому відділі
Миколаївського національного аграрного університету
54020, м. Миколаїв, вул. Георгія Гонгадзе, 9

Свідоцтво суб'єкта видавничої справи ДК № 4490 від 20.02.2013 р.