

**Інститут цифровізації освіти  
Національної академії педагогічних наук України  
Державна установа  
“Науково-методичний центр вищої та фахової передвищої освіти”  
Національний технічний університет України  
"Київський політехнічний інститут імені Ігоря Сікорського"**

**ПРАКТИЧНІ КЕЙСИ**

**ШТУЧНИЙ ІНТЕЛЕКТ У ДІЯЛЬНОСТІ ПЕДАГОГА**

**ЗБІРНИК МАТЕРІАЛІВ**

**Випуск 1 (2)**



**Київ 2024**

**Інститут цифровізації освіти  
Національної академії педагогічних наук України  
Державна установа  
“Науково-методичний центр вищої та фахової передвищої освіти”**

**Національний технічний університет України  
"Київський політехнічний інститут імені Ігоря Сікорського"**

**ПРАКТИЧНІ КЕЙСИ  
ШТУЧНИЙ ІНТЕЛЕКТ У ДІЯЛЬНОСТІ ПЕДАГОГА  
ЗБІРНИК МАТЕРІАЛІВ**

**Випуск 1 (2)**

**Київ 2024**

УДК 37.091.3:004.8

**Ш87**

*Рекомендовано до друку вченою радою  
Інституту цифровізації освіти НАПН України  
(Протокол № 9 від 30 травня 2024 року)*

**Рецензенти:**

**Лаврентьєва О.О.** д-р пед. н., проф., в.о. завідувача кафедри педагогіки, Криворізький державний педагогічний університет.

**Пінчук О.П.** к. пед. н., с.н.с., заступник директора з науково-експериментальної роботи, Інститут цифровізації освіти НАПН України.

**Ш87 Штучний інтелект у діяльності педагога (практичні кейси) :**  
збірник матеріалів. Вип. 1 (2) / упоряд. Ю. М. Богачков,  
Д. Ю. Богачков, П. С. Ухань. Київ : ІЦО НАПН України, 2024.  
60 с.

**ISBN 978-617-8330-38-5**

Збірник містить дослідницькі матеріали науковців, викладачів, студентів з тематики застосування систем штучного інтелекту в науковій та освітній діяльності.

Кожна робота містить результати дослідження та практичні кейси із застосуванням штучного інтелекту. Мета видання цього збірника – надати можливість швидкого ознайомлення користувачів з функціоналом та можливостями нових сервісів на основі вивчення конкретних зразків їх застосування.

Збірник адресований науковим, науково-педагогічним працівникам, вчителям, аспірантам, докторантам, студентам закладів вищої, фахової передвищої освіти та всім, хто цікавиться використанням сучасних цифрових технологій, зокрема штучного інтелекту, у науковій та освітній діяльності.

Матеріали надруковано в авторській редакції. За стилістичне та орфографічне оформлення відповідальність несуть автори публікацій.

УДК 37.091.3:004.8

ISBN 978-617-8330-38-5

© Богачков Ю.М., Богачков Д.Ю.,  
Захарченко А.С., Павловський О.М.,  
Півторак Д.О., Ухань П.С., 2024.  
© ІЦО НАПН України, 2024.  
© КПІ ім. Ігоря Сікорського, 2024.  
© НМЦ «ВФПО», 2024.

## ЗМІСТ

<b>ВСТУП</b> .....	2
<b>1. РОЗРОБЛЕННЯ НАВЧАЛЬНИХ МАТЕРІАЛІВ З ВИКОРИСТАННЯМ ШІ (Півторак Д.О.)</b> .....	4
<b>1.1. Дослідження середовищ ШІ</b> .....	4
<b>1.2. Навчальний контент, створений за допомогою сервісів ШІ</b> .....	6
<b>1.3. Блок домашніх завдань</b> .....	13
<b>1.4. Блок контролю та самоконтролю</b> .....	15
<b>1.5. Методика застосування обраних сервісів ШІ у власній         викладацькій діяльності</b> .....	18
<b>2. ЕЛЕКТРОНІКА (Павловський О.М.)</b>	
<b>Формування презентації з застосуванням ШІ</b> .....	20
<b>3. СТВОРЕННЯ НАВЧАЛЬНОГО КОНТЕНТУ ЗА ДОПОМОГОЮ СЕРВІСІВ ШІ (Захарченко А.С.)</b> .....	34
<b>3.1. Дослідження сервісів ШІ</b> .....	34
<b>3.2. Лекція 1: основи Python</b> .....	36
<b>3.3. Лекція 2: використання Python в наукових цілях</b> .....	47
<b>3.4. Питання для самоперевірки</b> .....	58

## ВСТУП

Збірник містить матеріали робіт слухачів курсу підвищення кваліфікації Штучний інтелект в освітній діяльності викладача. Матеріали публікуються за індивідуальною згодою авторів.

Кожна стаття містить результати дослідження різних сервісів, працюючих із застосуванням штучного інтелекту. Слухачам пропонувався шаблон структури випускної роботи, але вони мають можливість широких варіацій у своїх дослідженнях та формі представлення результатів.

Мета періодичного видання збірнику – обмін досвідом і обговорення питань застосування штучного інтелекту в освіті. Це – зручний майданчик для швидкого ознайомлення користувачів з функціоналом та можливостями нових сервісів на основі вивчення конкретних зразків їх застосування.

Збірник адресований вчителям, науковим і науково-педагогічним працівникам, керівниками наукових установ НАПН України, аспірантам, студентам закладів вищої, фахової передвищої освіти та всім, хто цікавиться використанням сучасних цифрових технологій, зокрема штучного інтелекту, у педагогічній, науковій і науково-педагогічній діяльності.

У рамках курсу слухачі досліджували сервіси, засновані на штучному інтелекті, обираючи з великої бази сервісів, наданої організаторами курсу (понад 2100 найменувань). Застосована оригінальна методика аналізу та класифікації функціоналу сервісів. Методика описана в першій статті збірника. Це забезпечує швидкий пошук необхідного сервісу.

Тематика поданих у першому томі матеріалів свідчить про актуальність розроблення науково-методичного забезпечення впровадження штучного інтелекту як в систему освіти, так і в наукових дослідженнях.

## **ВИКЛАДАЦЬКА КОМАНДА КУРСУ**

1. Викладач: **Богачков Юрій** – к.т.н, С.Н.С., с.н.с. відділу технологій відкритого навчального середовища Інституту цифровізації освіти НАПН України.

2. Консультант з питань ІКТ: **Ухань Павло** – к.пед.н, с.н.с. відділу технологій відкритого навчального середовища Інституту цифровізації освіти НАПН України.

## **КООРДИНАТОР КУРСУ:**

**Івченко Марина** – начальник відділу підвищення кваліфікації, Інституту післядипломної освіти Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського" м. Київ ([pkv.ipn.kpi@gmail.com](mailto:pkv.ipn.kpi@gmail.com))

# 1. РОЗРОБЛЕННЯ НАВЧАЛЬНИХ МАТЕРІАЛІВ З ВИКОРИСТАННЯМ ШІ

*Півторак Д.О.*

## 1.1. Дослідження середовищ ШІ

Назва сервісу: **Chat GPT**

Посилання на сервіс: <https://chat.openai.com/c/a53beb60-67ef-4a28-b93e-cc8a0921fec4>

Основні функції: **Генерація тексту за текстовим описом**

Умови використання: **Безкоштовно**

Якість отриманих результатів: **Якість висока**

**Відгук про сервіс:** Дуже швидко генерує текст за запитом. Можна легко згенерувати тестові запитання. Даний сервіс може бути зручним при освітньому процесі.

Недолік полягає у копіюванні тексту. При копіюванні тексту, який містить формул - не копіюються формули. Також код виглядає набагато краще на чорному фоні на сайті, ніж у тому вигляді, який копіюється. В лекційні матеріали краще додавати скрини коду (що було зроблено в даній роботі).

Назва сервісу: **Microsoft Bing**

Посилання на сервіс: <https://www.bing.com/images/create?>

Основні функції: **Генерація зображення за текстовим описом**

Умови використання: **Безкоштовно, потрібно реєстрація**

Якість отриманих результатів: **Зображення якісні**

**Відгук про сервіс:** Сервіс дуже зручний для використання, дозволяє згенерувати зображення за текстовим описом. Сервіс дозволяє використовувати як англійську, так і українську мову.

В якості текстового опису були використані наступні фрази: “графічна візуалізація MatLab”, “plot title MatLab”, “MatLab”. Даний сервіс може бути

зручним при освітньому процесі для створення ілюстративного матеріалу, наприклад, для презентацій.

Назва сервісу: **invideo**

Посилання на сервіс: <https://ai.invideo.io/>

Основні функції: **Генерація відео за текстовим описом**

Умови використання: **Безкоштовно, потрібно реєстрація**

Якість отриманих результатів: **Відео відображається з затримкою.**

**Ваш загальний відгук про сервіс:** Сервіс дуже зручний для використання, дозволяє згенерувати відео за текстовим описом. Сервіс дозволив згенерувати відео тривалістю 2.52.

До недоліків сервісу можна віднести те, що сервіс генерує відео, але відображення його відбувається з великою затримкою. Наприклад, в даному випадку було згенеровано відео тривалістю 2.52, а для перегляду відео знадобилось 4.32. Що являється дуже не зручним для використання таких відео в освітньому процесі. Відео озвучується англійською мовою, тому для студентів, які не володіють англійською мовою це буде проблемою.

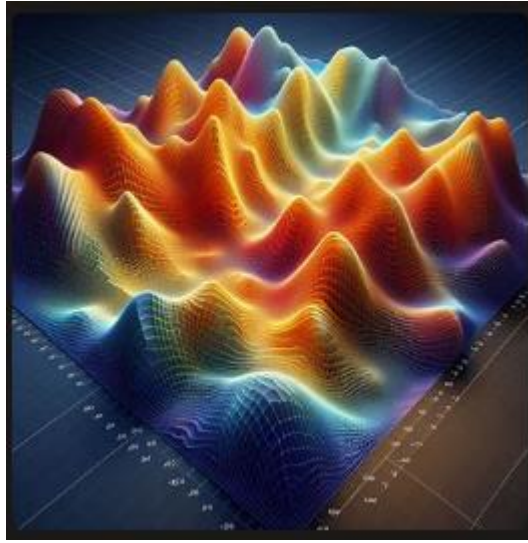
Згенероване відео можна використовувати тільки як додатковий матеріал.



## 1.2. Навчальний контент, створений за допомогою сервісів ШІ

Лекційний блок.

Лекція: Основи графічної візуалізації в середовищі MATLAB



Графічна візуалізація в MATLAB - це потужний інструмент для вивчення та подання даних. Вона дозволяє аналізувати та відображати інформацію у вигляді графіків, діаграм, зображень та інших візуальних елементів. Сьогодні ми розглянемо основні концепції та команди для графічної візуалізації в середовищі MATLAB.

### 1. Основні типи графіків

#### 1.1 Графік функції:

Графік функції - це графічне представлення відношення між змінною  $x$  і відповідними значеннями функції  $f(x)$ . Графік може бути побудований на площині, де одна вісь відображає значення аргументу  $x$ , а інша вісь відображає відповідні значення функції  $f(x)$ .

У математиці і графічному представленні функцій використовуються різні типи графіків, такі як лінійні графіки, параболі, експоненціальні графіки, логарифмічні графіки і т. д. Графік допомагає візуально зрозуміти залежність

між вхідними та вихідними значеннями функції, а також виявляти особливості та характеристики функції, такі як максимуми, мінімуми, точки перегину і т. д.

```
matlab Copy code  
  
x = linspace(0, 2*pi, 100);  
y = sin(x);  
plot(x, y);  
title('Графік функції sin(x)');  
xlabel('Вісь X');  
ylabel('Вісь Y');  
grid on;
```

## 1.2 Точковий графік:

Визначення точкового графіку можна сформулювати так:

Точковий графік функції - це множина точок у координатній площині, де кожна точка має координати  $(x,y)$ , де  $x$  - аргумент функції, а  $y$  - відповідне значення функції  $f(x)$ . Графік відображає залежність між вхідними значеннями і вихідними значеннями функції, визначаючи, як змінюються значення функції залежно від змінного аргументу.

Множина усіх точок графіку відображає форму та характер функції. Точковий графік може включати лінії, криві та інші геометричні елементи, які відображають зміни значень функції на площині.

```
matlab Copy code  
  
x = linspace(0, 2*pi, 100);  
y = sin(x);  
scatter(x, y);  
title('Точковий графік sin(x)');  
xlabel('Вісь X');  
ylabel('Вісь Y');  
grid on;
```

## 2. Створення діаграм

### 2.1 Кругова діаграма:

Кругова діаграма - це графічний метод відображення даних у вигляді кола, розділеного на сегменти, що відповідають різним часткам або категоріям. Цей тип діаграми часто використовується для представлення часток у відсотках чи пропорціях.

Визначення кругової діаграми:

Кругова діаграма - це вид графіку, який використовується для представлення часток чого-небудь цілого. Коло поділяється на сегменти, пропорційні відсоткам чи часткам, які вони представляють. Кожен сегмент позначає окрему категорію чи частку, а сума всіх сегментів складає 100%.

Кругові діаграми часто використовуються для ілюстрації розподілу даних у відсотках та вказівки на те, яка частина відноситься до загальної цілі. Це ефективний метод для візуалізації структури та порівняння часток у цілому.

```
matlab Copy code  
  
data = [25, 40, 15, 20];  
labels = {'Категорія 1', 'Категорія 2', 'Категорія 3', 'Категорія 4'};  
pie(data, labels);  
title('Кругова діаграма');
```

### 2.2 Гістограма:

Гістограма - це графічний метод представлення розподілу даних, що складається з прямокутних стовпців, де ширина кожного стовпця пропорційна діапазону значень, а висота відображає частоту (або відсоток) даних в цьому діапазоні.

Визначення гістограми:

Гістограма - це графік, який використовується для візуалізації розподілу числових даних в різних діапазонах або інтервалах. Гістограма складається з прямокутних стовпців, де кожний стовпець представляє певний діапазон

значень, а його висота відображає кількість чи частоту даних, які потрапили в цей діапазон.

Гістограми дозволяють швидко оцінити розподіл даних, виявити основні тенденції та виокремити можливі аномалії чи викиди. Цей вид графіку особливо корисний при аналізі великих обсягів числових даних.

```
matlab Copy code  
  
data = randn(1, 1000);  
histogram(data, 'BinWidth', 0.1);  
title('Гістограма випадкових чисел');  
xlabel('Значення');  
ylabel('Частота');
```

### 3. Зображення даних

#### 3.1 Зображення матриці:

Матриця - це організований спосіб представлення даних у вигляді таблиці, складеної з числових чи символічних елементів, розташованих у рядках та стовпцях. Кожен елемент матриці ідентифікується за його розташуванням у рядку та стовпці.

Зазвичай матриці позначають великими літерами, наприклад,  $A$ , і елементи матриці позначаються малими літерами та індексами. Наприклад, елемент у рядку  $i$  та стовпці  $j$  матриці  $A$  можна позначити як  $a_{ij}$

Матриці використовуються в різних областях математики та науки, таких як лінійна алгебра, теорія ймовірностей, комп'ютерна наука та інші. Матриці важливі для вирішення систем лінійних рівнянь, оптимізації, обробки сигналів, графічного програмування та багатьох інших областей.

matlab

Copy code

```
A = peaks(30);  
imagesc(A);  
title('Зображення матриці');  
colorbar;
```

### 3.2 3D графік:

3D графік - це графічне представлення об'єктів та їх взаємодій у тривимірному просторі, де зображення може мати три осі: ось  $x$ , ось  $y$  і ось  $z$ . У таких графіках кожен об'єкт може мати координати в трьох напрямках, що дозволяє відтворювати реалістичний об'ємний образ.

Визначення 3D графіку:

3D графік - це метод візуалізації об'єктів та просторових взаємодій, який використовує тривимірну координатну систему. Кожен об'єкт у тривимірному просторі описується своїми координатами  $(x, y, z)$ , де  $x$ ,  $y$  та  $z$  - це відстані вздовж відповідних осей.

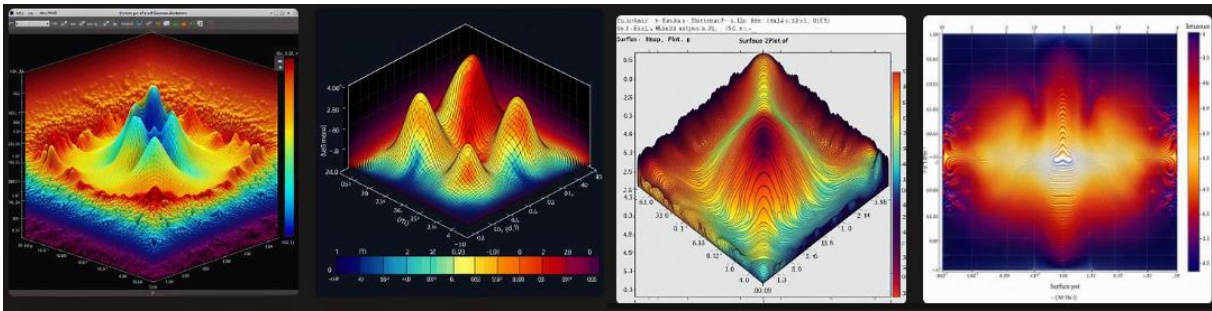
3D графіка застосовується в різних галузях, таких як комп'ютерна графіка, моделювання, наукові дослідження, дизайн і архітектура. Вона дозволяє створювати більш реалістичні та деталізовані відображення об'єктів та сцен, а також вивчати їхню просторову структуру.

matlab

Copy code

```
[X, Y] = meshgrid(-2:0.1:2, -2:0.1:2);  
Z = X .* exp(-X.^2 - Y.^2);  
surf(X, Y, Z);  
title('3D графік X*exp(-X^2 - Y^2)');  
xlabel('Вісь X');  
ylabel('Вісь Y');  
zlabel('Вісь Z');
```

## Приклади побудови графіків:



## 4. Анімація

### 4.1 Проста анімація:

Анімація - це процес створення візуального ефекту руху за допомогою послідовності зображень, кадрів чи об'єктів. У своєму суттєвому вигляді, анімація передає відчуття руху або зміни в часі.

Визначення анімації:

Анімація - це техніка створення ілюзії руху або живлення зображень, об'єктів чи сцен. Вона включає в себе створення послідовностей кадрів, які відтворюються з певною швидкістю (кадровою частотою), щоб створити враження руху. Анімація може бути створена як у ручному режимі (малювання кадрів один за одним) так і за допомогою спеціалізованих комп'ютерних програм.

Анімація широко використовується в різних галузях, включаючи кіно, телебачення, веб-дизайн, ігрову індустрію, навчання та багато інших сфер. Це ефективний спосіб виразу інформації та створення захопливого візуального враження для глядачів.

```
t = linspace(0, 2*pi, 100);
for i = 1:length(t)
    plot(t(1:i), sin(t(1:i)), 'LineWidth', 2);
    title('Анімація sin(t)');
    xlabel('Вісь X');
    ylabel('Вісь Y');
    grid on;
    drawnow;
end
```

Отже, Графічна візуалізація в MATLAB – це потужний інструмент для аналізу та інтерпретації даних. Ми розглянули лише базові приклади, але в MATLAB є безліч можливостей для створення різноманітних інтерактивних та ілюстративних візуалізацій для покращення розуміння ваших даних.

### 1.3. Блок домашніх завдань

#### Домашнє завдання: Основи графічної візуалізації в середовищі MATLAB

Завдання 1: Графіки функцій

Створіть MATLAB-скрипт, який буде містити код для візуалізації графіків наступних функцій:

1.1.  $y = 2x^2 - 3x + 1$  на інтервалі  $[0, 5]$ .

1.2.  $y = \sin(x)$  та  $y = \cos(x)$  на одному графіку.

1.3. Точковий графік для функції  $y = x^2$  з використанням масиву  $x$  з 50 випадкових чисел.

Завдання 2: Діаграми

Створіть MATLAB-скрипт для візуалізації наступних типів діаграм:

2.1. Кругова діаграма, що відображає відсотковий розподіл вашого часу на навчання, роботу, відпочинок і т.д. (використайте свої оцінки).

2.2. Гістограму, яка відображає розподіл оцінок студентів вашого курсу. Згенеруйте випадковий набір оцінок від 1 до 10 для 100 студентів.

Завдання 3: Зображення даних

Створіть MATLAB-скрипт, який використовує зображення для відображення наступних даних:

3.1. Зображення теплової карти для матриці  $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ .

3.2. 3D графік для функції  $z = e^{-x^2-y^2}$  на інтервалі  $[-2, 2]$  для  $x$  і  $y$ .

Завдання 4: Анімація

Створіть MATLAB-скрипт для створення анімації:

4.1. Анімація, яка показує рух точки вздовж графіка функції  $y = \sin(x)$  при зміні часу.



4.2. Анімація, що демонструє рух графіку функції  $y = x^2$  при зміні параметра  $a$  в рівнянні  $y = ax^2$ , де  $a$  змінюється від 1 до 5.

Завдання 5: Додаткове завдання

Оберіть тему самостійно та створіть власне графічне представлення даних у MATLAB. Наприклад, можливо, ви хочете візуалізувати зміни погоди, фінансові показники або будь-які інші дані, які вас цікавлять.

Вимоги до задачі завдання:

1. Завантажте всі MATLAB-скрипти та файли з даними у вказаний термін на вашому електронному ресурсі.
2. Забезпечте коментарі та пояснення для кожного рядка коду.  
У роботі має бути чітко вказано, яке завдання ви вирішили для додаткового завдання.
3. Якщо ви використовуєте додаткові бібліотеки або взаємодієте з іншими інструментами, зазначте це в коментарях.

## 1.4. Блок контролю та самоконтролю

Звісно, ось 10 тестових запитань з 5 варіантами відповідей на тему "Основи графічної візуалізації в середовище MATLAB":

1. Питання: Яка функція в MATLAB використовується для створення графіка функції?

- a) ``drawplot``
- b) ``createGraph``
- c) ``plot``
- d) ``generateChart``
- e) ``visualize``

Відповідь: c) ``plot``

2. Питання: Як відобразити дві функції на одному графіку в MATLAB?

- a) ``showDoublePlot``
- b) ``plot2Functions``
- c) ``multipleGraphs``
- d) ``hold on;``
- e) ``doubleChart``

Відповідь: d) ``hold on;``

3. Питання: Як створити кругову діаграму в MATLAB?

- a) ``createCircleChart``
- b) ``plotPie``
- c) ``circleDiagram``
- d) ``pie``
- e) ``drawDonut``

Відповідь: d) ``pie``

4. Питання: Яка функція в MATLAB використовується для створення гістограми?

- a) ``barGraph``
- b) ``createHistogram``
- c) ``plotHistogram``
- d) ``hist``
- e) ``drawBar``

Відповідь: d) ``hist``

5. Питання: Як відобразити зображення матриці як теплову карту в MATLAB?

- a) ``drawHeatmap``
- b) ``imageMatrix``
- c) ``plotHeatmap``
- d) ``heatmap``
- e) ``matrixImage``

Відповідь: d) ``heatmap``

6. Питання: Яка функція в MATLAB використовується для створення 3D графіку?

- a) ``plot3D``
- b) ``create3DGraph``
- c) ``draw3D``
- d) ``surf``
- e) ``visualize3D``

Відповідь: d) ``surf``

7. Питання: Як використовувати ``scatter`` в MATLAB?

- a) ``createScatterPlot``
- b) ``plotScatter``
- c) ``scatterPlot``
- d) ``drawPoints``
- e) ``scatter``

Відповідь: e) ``scatter``

8. Питання: Яка функція в MATLAB дозволяє відобразити анімацію?

- a) ``animateGraph``
- b) ``createAnimation``
- c) ``plotAnimation``
- d) ``animation``
- e) ``animate``

Відповідь: d) ``animation``

9. Питання: Які команди в MATLAB допомагають встановити заголовок для графіку та підписи для вісей?

- a) ``title``, ``xlabel``, ``ylabel``
- b) ``setGraphTitle``, ``setXLabel``, ``setYLabel``

- c) ``graphTitle`, `xAxisLabel`, `yAxisLabel``
  - d) ``plotTitle`, `axisXLabel`, `axisYLabel``
  - e) ``setChartTitle`, `setXAxisLabel`, `setYAxisLabel``
- Відповідь: a) ``title`, `xlabel`, `ylabel``

10. Питання: Яка функція в MATLAB дозволяє зберігати графік у зображенні?

- a) ``saveImage``
- b) ``exportGraph``
- c) ``savePlot``
- d) ``printImage``
- e) ``saveas``

Відповідь: e) ``saveas``

## 1.5. Методика застосування обраних сервісів ШІ у власній викладацькій діяльності

В процесі роботи були використанні сервіс **Chat GPT**. Сервіс дозволяє генерувати текст лекцій, домашнього завдання та тестові запитання з варіантами відповідями.

Спочатку при запиті “Основи графічної візуалізації в середовище MATLAB” сервіс видав у вигляді класифікація типи графіків з кодом програми. Потім були запити, наприклад, “визначення графіків функції”, “визначення 3D графіки”, “визначення гістограми” та т.д. (дані дії були зроблені для того, щоб детальніше розглянути лекційний матеріал).

В сервісі **Microsoft Bing** були згенеровані різний ілюстративний матеріал.

Сервіс **invideo** дозволив згенерувати відео. Але, на жаль, дане відео можна використовувати тільки як додатковий матеріал, так як озвучування відбувається англійською мовою та на відео присутні водні знаки (у безкоштовній версії). Також до недоліку згенерованого відео можна віднести: відео відображається з затримкою. Таким чином, обраний сервіс не являється зручним, через велику кількість недоліків.

Зміст роботи викладача	Сервіси ШІ та їх функції які застосовувались для реалізації
Підготовка лекційних матеріалів	Chat GPT - генерація тексту за текстовим описом
Підготовка тестових запитань	Chat GPT - генерація тексту за текстовим описом Conker - генерація тексту за текстовим описом (генерує тести англійською мовою за текстовим описом (англійською або українською мовою). Може використовуватись для генерації тестів для студентів-іноземців)

<p>Підготовка ілюстративного матеріалу</p>	<p>Microsoft Bing - генерація зображень за текстовим описом          Deep Dream Generator - генерація зображень за текстовим описом (сервіс дозволить згенерувати велику кількість зображень, з яких в подальшому можна буде обрати те, що більш підходить для ілюстративних матеріалів)</p>
<p>Підготовка відео-матеріалів (англійською мовою)</p>	<p>invideo - генерація відео за текстовим описом          runwayml - генерація коротких відео (4 секунди) за текстовим описом</p>

## 2. ЕЛЕКТРОНІКА

### Формування презентації з застосуванням ШІ

*Павловський О.М.*



*КПІ ім. Ігоря Сікорського  
Приладобудівний факультет  
Кафедра комп'ютерно-інтегрованих  
оптичних та навігаційних систем*



# Електроніка

**Вступ. Історичні аспекти**

Лектор: к.т.н., доцент  
*Павловський Олексій Михайлович*

2

## Умовна класифікація електронних пристроїв

В залежності від застосовуваної елементної бази виділяють чотири основних покоління розвитку електроніки:

- ❑ **Перше покоління (1904-1950 рр.):** В цей період основними компонентами елементної бази електронних пристроїв були електровакуумні та газорозрядні прилади, такі як електронні лампи, електронно-вакуумні трубки, газорозрядні індикатори тощо.
- ❑ **Друге покоління (1950 - початок 60-х рр.):** У цей період відзначився застосуванням дискретних напівпровідникових приладів, таких як діоди, транзистори, тиристори та інші.
- ❑ **Третє покоління (1960-1980 рр.):** Цей етап пов'язаний із стрімким розвитком мікроелектроніки та створенням інтегральних схем різного рівня інтеграції, а також мікросбірок. Електронні пристрої на цьому етапі відзначаються значним підвищенням надійності, зменшенням розмірів, маси та споживання енергії.
- ❑ **Четверте покоління (з 1980 р. по теперішній час):** Характеризується подальшою мініатюризацією електронних пристроїв за допомогою великих і

Слайд 2 перефразований за допомогою Чату GPT

3

Яке покоління електронних пристроїв характеризується застосуванням дискретних напівпровідникових приладів?

- A. перше покоління
- Б. друге покоління
- В. третє покоління
- Г. четверте покоління

Answer

Correct answer: B



Слайд 3

Тести зроблені за допомогою

Class Point

<https://www.classpoint.io/>

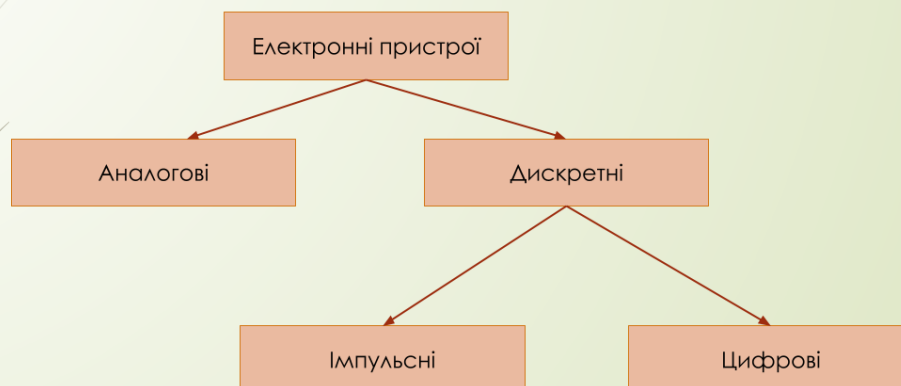
Слайд 3 Картинки згенеровані Stable Diffusion та Lexica Aperture

<https://lexica.art/aperture>

<https://stablediffusionweb.com/#ai-image-generator>

4

Умовна класифікація електронних пристроїв

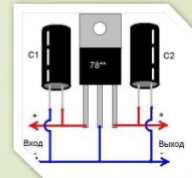
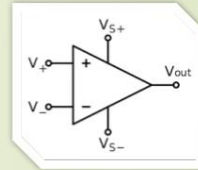




5

## Аналогові електронні пристрої

- Аналогові електронні пристрої використовуються для отримання, зміни та передачі сигналів, які змінюються згідно з неперервною (аналоговою) функцією. Ці пристрої відрізняються простотою та високою швидкістю, але їхні параметри можуть бути нестабільними та мають низьку стійкість до впливу зовнішніх факторів, таких як температура, вологість, час тощо.
- До аналогових електронних пристроїв відносяться електронні підсилювачі, операційні підсилювачі, комутатори, компаратори, стабілізатори напруги та інші.




Слайд 5 Частково згенерований, частково перефразований із старої лекції з використанням Чату GPT

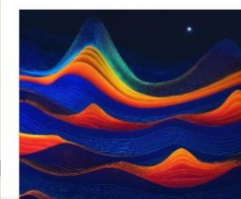
6

Аналогові електронні пристрої використовуються для прийому, перетворення і передачі сигналів, які змінюються за законом якої функції?

- A. Лінійної функції
- Дискретної функції
- Безперервної (аналогової) функції
- Комбінованої функції

 Answer

Correct answer: C



Слайд 6 Тести зроблені за допомогою Class Point

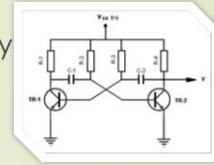
<https://www.classpoint.io/>

Картинки згенеровані Lexica Aperture

7

## Дискретні електронні пристрої


- Дискретні електронні пристрої призначені для прийому, перетворення і передачі електричних сигналів, представлених в дискретній формі. Такі пристрої відрізняються високою надійністю, невеликою споживаною потужністю і вартістю.
- Імпульсні електронні пристрої формують імпульсну послідовність сигналів. Процес перетворення аналогової інформації в послідовність імпульсів називається імпульсною модуляцією.  
На практиці широко використовується амплітудна, широтно-імпульсна і фазоімпульсна модуляції.
- До імпульсних електронних пристроїв відносяться: мультивібратори, блокувальні генератори, функціональні перетворювачі, генератори пилоподібного сигналу.



8

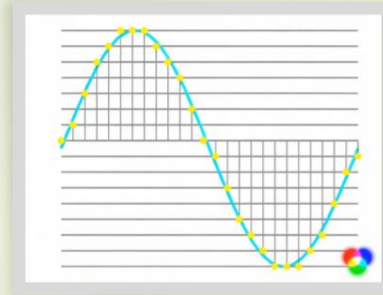
## Що означає імпульсна модуляція?



 Answer

Suggested answer:

Процес перетворення аналогової інформації в послідовність імпульсів



Слайд 8 Запитання згенероване за допомогою **Class Point**

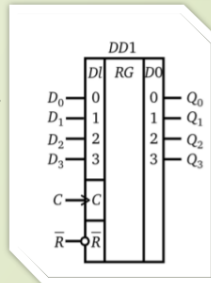
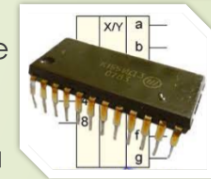
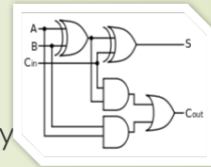
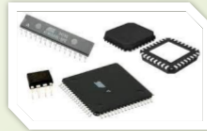
Картинки згенеровані **Lexica Aperture**

Анімація створена в **Runwayml**

<https://app.runwayml.com/>

## Дискретні електронні пристрої

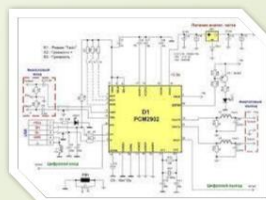
- В **цифрових електронних пристроях** відбувається кодування сигналу, тобто перетворення його в певну послідовність однотипних імпульсів.
- Цифрові електронні пристрої зараз отримали дуже широке поширення завдяки високій надійності та завадостійкості, можливості тривалого зберігання інформації без її втрати, енергетичній сумісності та технологічності інтегрованої елементної бази.
- До цифрових електронних пристроїв відносяться: логічні елементи, тригери, регістри, дешифратори, шифратори, мультиплексори, демультимплексори, суматори і т.д.



Слайди 9-14 рійрайт за допомогою Чату GPT

## Комбіновані пристрої

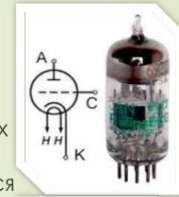
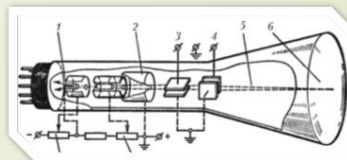
- Деякі електронні пристрої використовують як аналогову так і цифрову інформацію. Такі пристрої відносяться до **комбінованих електронних пристроїв**.
  - Це аналого-цифрові перетворювачі (АЦП) та цифро-аналогові перетворювачі (ЦАП).



## Елементна база електронних пристроїв

### Електровакуумні прилади

- Принцип дії електровакуумних приладів заснований на створенні електронного потоку і управлінні цим потоком вільних електронів за допомогою електричних або електромагнітних полів. Для отримання потоку вільних електронів використовується спеціальний металевий або напівпровідниковий електрод - катод.
- До електровакуумних приладів відносяться електронні лампи, електронно-променевої трубки, електровакуумні фотоелектронні прилади, вакуумні люмінесцентні індикатори та ін.

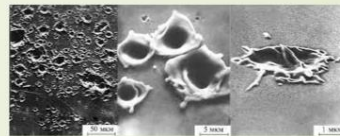
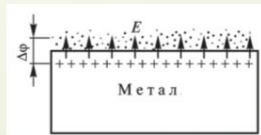


## Електронна емісія

- Процес вивільнення електронів з катода отримав назву "електронна емісія". Для подолання гальмівної дії подвійного електричного шару електронів катода потрібно забезпечити додаткову енергію. Залежно від методу передачі цієї додаткової енергії існують різновиди електронної емісії:
  - **Термоелектронна емісія:** Додаткова енергія передається електронам при нагріванні катода.
  - **Вторинна електронна емісія:** Виникає при облученні катода потоком електронів або іонів, які рухаються з великою швидкістю.
  - **Фотоелектронна емісія:** Подія, при якій електромагнітне випромінювання впливає на поверхню катода, викликаючи вивільнення електронів.
  - **Електростатична емісія:** Вихід електронів з катода забезпечується сильним електричним полем на його поверхні.
- У масово використовуваних ЕП широкого поширення набули катоди з термоелектронної емісією прямого і відносного накалу.

## Електронна емісія

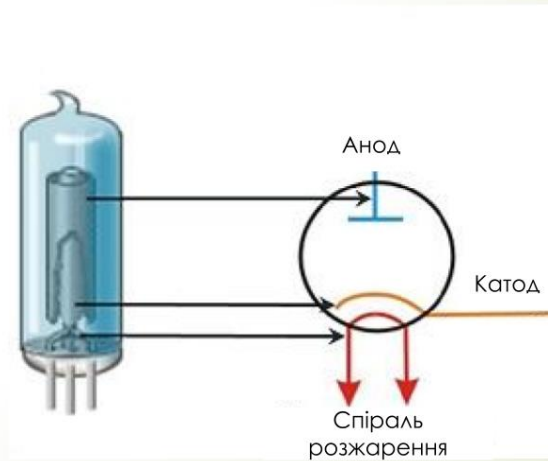
- У катодах прямого накалу струм проходить безпосередньо по катоду, нагріваючи його. Зазвичай вони виготовляються з вольфрамового дроту і нагріваються до температури понад 2000 °С.
- Катоди прямого розжарення відрізняються великим рівнем шумів, що призводить до порушення нормального режиму роботи електровакуумних приладів. Тому частіше використовуються катоди непрямого накалу, які представляють собою циліндр, усередині якого розташовується спіраль нагрівача, ізольована від катода. Зовнішня поверхня циліндра покрита активним шаром.



## Електровакуумний діод

- Залежно від кількості електродів розрізняють різновиди електровакуумних ламп: двохелектродні - діоди, трьохелектродні - тріоди, чотирьохелектродні - тетроди, п'ятиелектродні - пентоди і так далі. У певних випадках можуть використовуватися корпуси, в яких розташовані дві лампи, наприклад, тріод і пентод.
- **Електровакуумний діод** представляє собою балон із скла або металу, в якому створений високий вакуум. Усередині балона розміщуються два електроди: анод і катод. Під час нагрівання катода енергія електронів збільшується, і частина з них виходить з поверхні катода. Якщо до анода прикласти додатню напругу, між анодом і катодом утворюється прискорююче електричне поле, і через лампу протікає електричний струм. Змінюючи полярність напруги, можна створити гальмівне електричне поле, що призводить до припинення електричного струму через діод.

## Електровакуумний діод



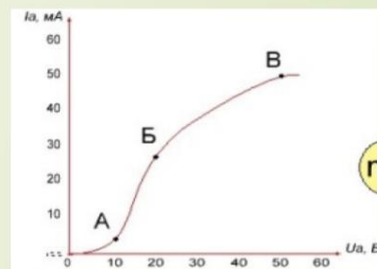
## Електровакуумний діод

На самому початку, коли на аноді відсутня напруга, навколо катода, в наслідок його нагріву, формується електронна хмара.

Коли на аноді виникає додатня невелика напруга, найшвидші електрони, що входять в електронну хмару катода, починають рухатися до анода. В результаті можна реєструвати анодний струм невеликої величини (точка А).

У ситуації, коли анодна напруга буде продовжувати збільшуватися, з електронної хмари все більше число електронів будуть перетікати до анода, до повного розсіювання електронної хмари. Цей стан відповідає точці В на графіку, наведеному вище. Така напруга означає, що всі електрони, що емітуються з катода будуть негайно притягатися до анода.

Подальше наростання анодного струму при збереженні величини напруги не відбуваються.



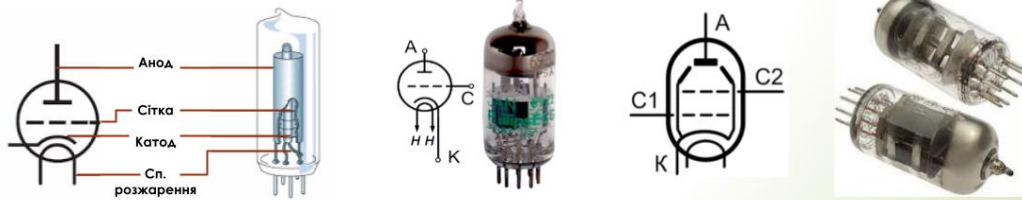
Вольт-амперна характеристика електровакуумного діода

Слайди 16-17 пояснення згенеровані за допомогою Чату GPT

## Електровакуумний триод та тетрод

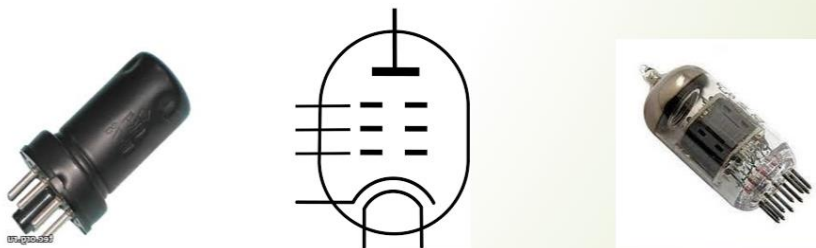
**Електровакуумні триоди** мають в своєму складі додаткову керуючу сітку, яка розташовується поблизу катода. Змінюючи напругу  $U_c$  на керуючій сітці, можна управляти анодним струмом. Тому триод можна використовувати в якості підсилювального елемента.

Для підвищення коефіцієнта підсилення, до складу ламп вводяться додаткові сітки. Лампа, що має дві сітки - керуючу і екрануючу, отримала назву **тетрод**. Екрануюча сітка розташовується між анодом і керуючою сіткою і виконується у вигляді спіралі. Екрануюча сітка ефективно послаблює вплив анодної напруги на електрони, що вилітають з катода, і при цьому значно підвищується коефіцієнт підсилення сигналу.

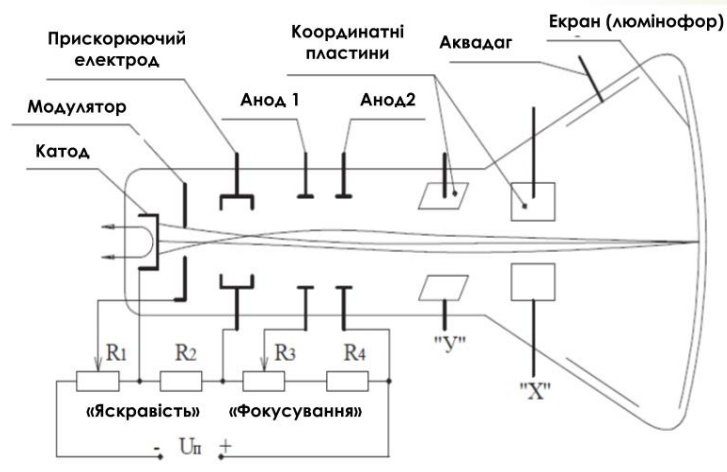


## Електровакуумний пентод

- Недоліком тетродів є можливість появи вторинної емісії з анода на екрануючу сітку. Для виключення вторинної емісії між анодом і екрануючою сіткою встановлюють додаткову сітку антидинатронну (захисну), на яку подають напругу  $U = 0$ . Така лампа отримала назву **пентод**. Можуть працювати із високими частотами (близько ГГц).



## Електронно-променева трубка




## Електронно-променева трубка (ЕПТ)

- Система електродів ЕПТ розміщена у скляному балоні, який є вакуумним. Катод непрямого розжарення розташований в торці балона і призначений для генерації електронного потоку. Послідовно слідує модулятор, який представляє собою циліндр із отвором для викидання електронів. На модулятор подається напруга, що змінює інтенсивність електронного потоку і, відповідно, "яскравість" зображення на екрані. Прискорюючий електрод, перший і другий аноди, формують електронну лінзу, яка фокусує електронний промінь на екрані.
- Для відхилення променя використовуються дві пари пластин для відхилення по "Х" та "У". Напруга на цих пластинах створює електричне поле, яке відхиляє електронний промінь. Щоб уникнути скупчення електронів біля екрану і спотворення зображення, стінки балона поблизу екрану покриті графітовим шаром (аквадагом). Електрони з аквадагу стікають на другий анод.



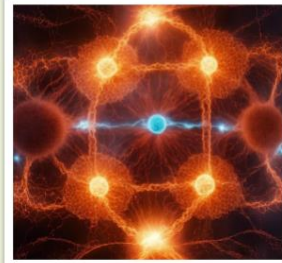
21

## Чому корпус електронно-променевої трубки зроблений із скла?

 Answer

Suggested answer:

Світіння спостерігається з протилежного боку електронного бомбардування.



Слайди 21-22 Тести та питання зроблені за допомогою **Class Point**

Слайд 21 Картинки згенеровані **Stable Diffusion** та **Lexica Aperture**

Слайди 21-22 Проведена сумаризація відео з Ютуб:

<https://www.youtube.com/watch?v=IoDoW5kykkw>

[https://www.youtube.com/watch?v=\\_tLuDJeh8yU](https://www.youtube.com/watch?v=_tLuDJeh8yU)

[https://www.youtube.com/watch?v=s3vpH3A\\_eTA](https://www.youtube.com/watch?v=s3vpH3A_eTA)

<https://www.youtube.com/watch?v=Rb6MguN0Uj4>

<https://www.youtube.com/watch?v=GzMh4q-2HjM>

<https://www.youtube.com/watch?v=Gbo7m0FAOqc>

22

## Яка роль модулятора в електронно-променевій трубці (ЕПТ)?

- A. Формування електронного потоку  
Утворення електронної лінзи  
Відхилення променя від осей X і Y  
Збудження атомів люмінофора

 Answer

Correct answer: A




## Маркування електровакуумних ламп

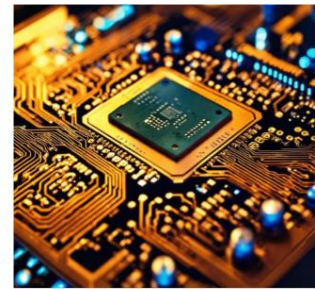
- **Маркування ламп** включає кілька цифрових і буквених елементів.
- Перший елемент - число, яке вказує напругу накалу в вольтах (заокруглено).
- Другий елемент - буква, що характеризує тип лампи:
  - ❖ "Д" - діод;
  - ❖ "Ц" - кенотрон (випрямний діод);
  - ❖ "Н" - триод;
  - ❖ "П", "Ж" - пентод;
  - ❖ "Ф" - триод-пентод
- Третій елемент - порядковий номер даного типу лампи.
- Четвертий елемент - буква, що характеризує конструктивне оформлення лампи:
  - ❖ "С" - в скляному балоні, діаметром більше 22,5 мм;
  - ❖ "П" - пальчикові;
  - ❖ "Б" - надмініатюрні з діаметром від 6 до 10,5 мм.
- Наприклад, 6ППП- пентод пальчиковий з напругою накалу ~ 6,3.

## Що означає перший елемент маркування лампи?

- A. Кількість елементів у лампі  
 Напругу накалу лампи  
 Тип лампи  
 Конструктивне оформлення лампи

 Answer

Correct answer: B



## Питання для самоконтролю(Д/з):

### □ Блок 1:

1. Що характеризує перше покоління електроніки?
2. Які прилади відносяться до другого покоління?
3. Які зміни відбулися в електронних пристроях третього покоління?
4. Яка основна особливість четвертого покоління електроніки?

### □ Блок 2:

1. Які елементи складають елементну базу першого покоління електроніки?
2. Які напівпровідникові прилади використовувалися в другому поколінні?
3. Що характеризує третє покоління за використанням елементної бази?
4. Які технологічні зміни відбулися у четвертому поколінні електроніки?

### □ Блок 3:

1. Яка відмінність між МОП-структурами та транзисторами?
2. Які пристрої часто виготовляються за технологією МОП?
3. Як МОП-структури впливають на роботу полевих транзисторів?

## Питання для самоконтролю:

### □ Блок 4:

1. Як можна іншими словами описати розвиток електроніки в різні періоди?
2. Які основні характеристики другого покоління електроніки?
3. Як змінилася надійність та розмір електронних пристроїв в третьому поколінні?
4. Як відрізняється четверте покоління від попередніх за мініатюризацією?

### □ Блок 5:

1. Як виникає електронна хмара в електровакуумному діоді?
2. Які функції виконують модулятор та прискорюючий електрод в ЕПТ?
3. Як відбувається відхилення електронного променя в ЕПТ?
4. Як запобігається скупченню електронів поблизу екрану в ЕПТ?

Використано сервіс **Skipit.Ai** <https://skipit.ai/>

Анотації до відео перекладені і разом з текстом презентації опрацьовані **Чатом GPT**. Сформовано 5 блоків запитань для самоконтролю/домашнього завдання

Сервіси є в таблицях Notion. З сервісами **Stable Diffusion**, **Lexica Aperture**, **Runwayml**, **Skipit.Ai** працював я (в графі автор вказане моє Прізвище). **Class Point** використав по рекомендації Сергія Сироти після його презентації.

Із недоліків зазначених сервісів:

**Class Point** дає можливість згенерувати всього 20 запитань, після цього треба купувати.

**Stable Diffusion** – генерує довго, тяжко правильно підібрати промт на технічний малюнок, плюс багато фільтрів які незрозуміло як працюють, вцілому після аналізу презентації одразу видно, що генерувалося ШІ. Відносно довго генерує зображення.

**Lexica Aperture** – обмеження в 100 зображень на тиждень.

**Runwayml** – обмеження в ~100с відео, за цей час важкувато підібрати вдалий запит. Краще «анімує» картинки ніж текстові запити.

**Skipit.Ai** – працює лише із англomовним відео, не всі відео підходять для сумаризації, не вистачає якихось тегів.

Посилання на сервіси:

<https://lexica.art/aperture>

<https://stablediffusionweb.com/#ai-image-generator>

<https://app.runwayml.com/>

<https://skipit.ai/>

### 3. СТВОРЕННЯ НАВЧАЛЬНОГО КОНТЕНТУ ЗА ДОПОМОГОЮ СЕРВІСІВ ШІ (Захарченко)

*Захарченко А.С.*

#### 3.1. Дослідження сервісів ШІ

##### 1. Tome AI

**Посилання на сервіс:** <https://tome.app/>

**Короткий опис:** Середовище Tome створене, візуалізації і узагальнення своїх ідей. Це інтелектуальний, гнучкий інструмент, який дозволяє створювати будь-що: презентації, односторінкові сторінки, цільові сторінки, портфоліо, дошки настрою та інший вміст.

**Функції:** візуалізація, генерація тексту, генерація зображень, віддалений доступ, створення презентацій.

**Умови використання:** платний, але даються початкові “кредити” для тесту середовища

**Якість отриманих результатів:** якість згенерованого контенту достатньо висока, структурована та зрозуміла. Потребує мінімального коректування.

**Ваш загальний відгук про сервіс:** за потреби генерування значного об’єму презентацій для навчальної дисципліни та можливості оплати PRO версії може бути зручним інструментом для генерації «сирого» матеріалу, як відправної точки для подальшої персоналізації та актуалізації викладачем.

##### 2. Diagramming AI

**Посилання на сервіс:** <https://diagrammingai.com/>

**Короткий опис:** створює діаграми за допомогою ШІ на основі GPT відповідно до введеного тексту. Дозволяє отримувати діаграми в різних форматах, наприклад блок-схеми, діаграми послідовності, діаграми Ганта тощо.

**Функції:** візуалізація, генерація схем та діаграм, використання Mermaid і PlantUML, редагування отриманих результатів

**Умови використання:** платний, але даються початкові “кредити” для тесту середовища.

**Якість отриманих результатів:** отримано результат достатньо високої якості.

**Ваш загальний відгук про сервіс:** дуже зручно за рахунок того, що він заснований на основі чату GPT можливо відразу вказувати загальну тематику діаграм та схем, інформація до яких буде отримана автоматично. Тим не менше, схеми мають достатньо просту будову, тому хоч даний сервіс і подає гарні ідеї по візуалізації інформації, що стосується запиту, є достатньо безкоштовних сервісів, що дозволяють виконувати ті ж операції, хоч і вручну.

### **3. DeepL**

**Посилання на сервіс:** <https://www.deepl.com/>

**Короткий опис:** Сучасний перекладач на основі штучного інтелекту. Нейронні мережі DeepL здатні вловлювати навіть найменші нюанси та відтворювати їх у перекладі на відміну від будь-якого іншого сервісу. Для оцінки якості моделей машинного перекладу, регулярно проводяться сліпі тести. Під час сліпих тестів професійні перекладачі вибирають найточніший переклад, не знаючи, яка компанія його виготовила. DeepL перевершує конкурентів у співвідношенні 3:1.

**Функції:** переклад, покращення письма, динамічна корекція тексту, переклад файлів

**Умови використання:** є Pro версія, але більшість корисних функцій доступні безкоштовно.

**Якість отриманих результатів:** дуже висока якість перекладу.

**Ваш загальний відгук про сервіс:** DeepL є одним з найкращих перекладачів на сьогоднішній день. Але його найкорисніша функція полягає в можливості динамічно вносити корективи в текст, що автоматично коректує наступні слова, для збереження граматичних структур речень. Особливо корисно використовувати при роботі з технічними текстами та термінами.

## 3.2. Лекція 1: основи Python

### Тема 1: Вступ

#### 1.1. Визначення Python та його популярність в наукових галузях.

Python - це високорівнева мова програмування загального призначення, яка здатна ефективно вирішувати різні завдання. Вона володіє синтаксисом, який дуже лаконічний та простий для вивчення, що робить її популярним вибором серед програмістів початківців. Python має широкі можливості і кожен може використовувати його для своїх потреб. Однак, основна сила Python полягає в його використанні у багатьох наукових галузях. Це пов'язано з наявністю багатьох бібліотек та інструментів, спеціально розроблених для наукових обчислень і аналізу даних. Один з найпопулярніших інструментів для наукових обчислень, який використовує Python, - це бібліотека NumPy. Вона надає широкі можливості для роботи з масивами та математичними функціями і часто використовується для наукових обчислень. Ще одна важлива бібліотека - це SciPy, яка надає функції для чисельного інтегрування, оптимізації, обробки сигналів та багато іншого. Крім того, для аналізу даних популярно використовувати бібліотеку Pandas, яка забезпечує інструменти для маніпулювання та аналізу даних у зручній табличній формі. Також варто згадати про бібліотеку Matplotlib, яка забезпечує можливості побудови графіків і діаграм. Python широко використовується в таких наукових галузях як фізика, астрономія, біоінформатика, машинне навчання та багато інших. Його популярність пояснюється зручністю використання, багатим екосистемою бібліотек та активним співтовариством розробників, що сприяє постійному розвитку мови та її інструментарію.

### Тема 2: Основи Python

#### 2.1. Змінні та типи даних.

Python - це мова програмування, яка підтримує різні типи даних. Основні типи даних в Python включають:

1. Числа: тут входять цілі числа (integers), дробові числа (floats) і комплексні числа (complex numbers).

2. Рядки: це послідовності символів, які представляють текст. Рядки можна визначати за допомогою одинарних (' '), подвійних (" ") або потрійних ("\"") або \"\"\" \"\"\"") лапок.

3. Булеві значення: це лише два можливих значення - True (істина) і False (хиба). Булеві значення використовуються для умовних виразів та логічних операцій.

4. Списки: це змінні, які можуть містити послідовності інших об'єктів. Списки використовуються для зберігання даних, які можуть змінюватися.

5. Кортежі: це схожі на списки змінні, але вони є незмінними, тобто зміст кортежу не може бути змінений після його створення.

6. Словники: це асоціативні структури даних, які містять пари ключ-значення. Ключі використовуються для доступу до відповідних значень.

Це лише декілька основних типів даних в Python. Мова також має інші типи, такі як множини, файли, класи та об'єкти, і багато інших, що забезпечують розширені можливості для програмування.

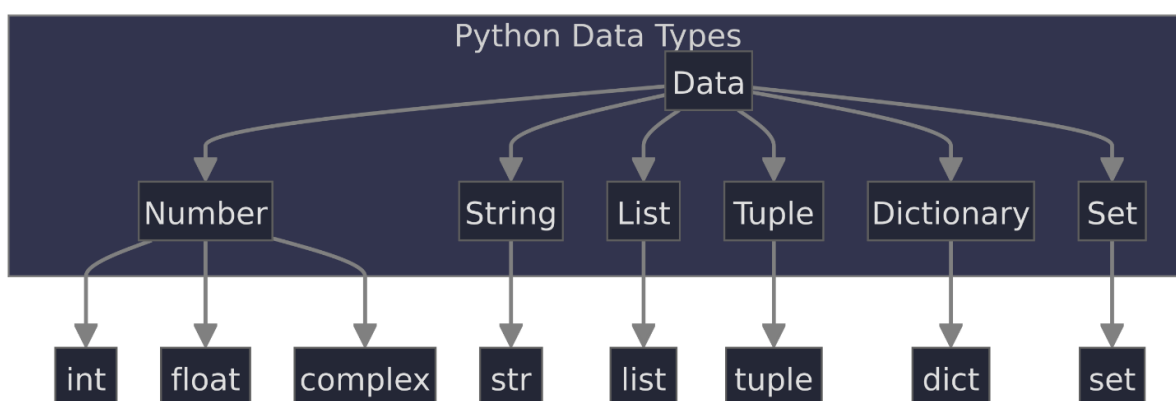


Рис. Запит: «python data type diagram»

## 2.2. Оператори та вирази.

Оператори та вирази Python є важливими елементами мови програмування Python. Оператори використовуються для виконання дій над операндами, які можуть бути змінними, константами або виразами. Вирази складаються з операторів та операндів і виконують певні обчислення.

Деякі основні оператори Python включають:

1. Арифметичні оператори: `+` (додавання), `-` (віднімання), `*` (множення), `/` (ділення), `%` (остача від ділення), `//` (цілочисельне ділення) та `**` (піднесення до степеня).

2. Оператори порівняння: `==` (рівність), `!=` (нерівність), `<` (менше), `>` (більше), `<=` (менше або рівне), `>=` (більше або рівне).

3. Логічні оператори: `and` (логічне І), `or` (логічне АБО), `not` (логічне НЕ). Вони використовуються для комбінування умов.



4. Оператори присвоєння: `=` (присвоєння значення змінній), `+=` (додати і присвоїти), `-=` (відняти і присвоїти), `\*=` (помножити і присвоїти), `/=` (поділити і присвоїти) та ін.

5. Оператори умовного вибору: `if`, `else`, `elif` (використовується, коли потрібно перевірити кілька умов).

6. Циклічні оператори: `for` (цикл for), `while` (цикл while). Вони використовуються для повторного виконання фрагменту коду.

7. Оператор доступу до елементів: `[]` (для доступу до елементів списків, рядків, кортежів) та `.` (для доступу до атрибутів об'єктів).

Це лише невеликий список основних операторів та виразів Python. У мові програмування Python доступно багато інших операторів, які можуть бути використані для виконання різних завдань.

### 2.3. Структури даних: списки, кортежі, словники.

У веб-розробці й програмуванні Python, є кілька основних структур даних, таких як списки, кортежі та словники.

**Списки (Lists):** Список є змінюваним, упорядкованим та мутабельним типом даних в Python. Він може містити елементи будь-якого типу даних, і кожен елемент може бути доступний за його індексом. Списки можуть бути змінені, додавані елементи, видаляні елементи, а також розширювані за потребою.

Приклад:

```
my_list = [1, 2, 3, 'four', 'five']
```

**Кортежі (Tuples):** Кортеж є незмінюваним, упорядкованим та імутабельним типом даних в Python. Він схожий на список, за винятком того, що кортежі не можуть бути змінені після свого створення (тобто елементи не можуть бути додані або видалені). Кортежі зазвичай використовуються для представлення колекцій, де елементи не повинні змінюватись.

Приклад:

```
my_tuple = (1, 2, 3, 'four', 'five')
```

**Словники (Dictionaries):** Словник є змінюваним, неупорядкованим та імутабельним типом даних в Python. Він складається з ключів та значень, де кожен ключ пов'язаний зі своїм значенням. Словники можуть бути змінені, додані нові пари ключ-значення, видалені записи або змінені значення за ключем.

Приклад:

```
my_dict = {'name': 'John', 'age': 25, 'city': 'New York'}
```

Ці структури даних дуже потужні та надають широкі можливості для зручної роботи з інформацією в Python. Вони використовуються в багатьох програмних завданнях для зберігання та обробки даних.

#### 2.4. Умовні оператори та цикли.

Умовні оператори та цикли є важливими конструкціями в програмуванні Python, які дозволяють керувати поведінкою програми залежно від певних умов.

Умовні оператори включаються за допомогою ключових слів ``if``, ``elif`` (скорочення для "else if") та ``else``. Синтаксис має наступний вигляд:

```
if умова_1:
    # код, що виконується, якщо умова_1 істинна
elif умова_2:
    # код, що виконується, якщо умова_2 істинна
else:
    # код, що виконується, якщо жодна з умов не є істинною
```

Цикли в Python можуть бути реалізовані за допомогою наступних конструкцій: ``for`` та ``while``.

``for`` - це цикл, який виконується певну кількість разів, виконуючи певні дії. Синтаксис ``for`` має наступний вигляд:

```
for елемент in ітерований_об'єкт:
    # код, який виконується для кожного елемента ітерованого об'єкта
```

``while`` - це цикл, який виконується, поки певна умова є істинною. Синтаксис ``while`` має наступний вигляд:

```
while умова:
    # код, який виконується, поки умова є істинною
```

Умовні оператори та цикли є потужними інструментами для контролю поведінки програми в залежності від умов та виконання певних дій багаторазово. Наведені конструкції застосовуються в широкому спектрі задач програмування.

## Тема 3: Функції

### 1.1. Визначення та використання функцій.

Python - це високорівнева мова програмування загального призначення. Вона була створена в кінці 1980-х і початку 1990-х років Гвідо ван Россумом і вироблена для бути простою у вивченні та читанні.

У Python є багато корисних функцій, що полегшують роботу з програмами. Основні функції використовуються для виконання певних завдань та можуть бути викликані у коді програми.

Основні приклади використання функцій Python:

1. Вбудовані функції: Python має багато вбудованих функцій, таких як `print()`, `len()`, `input()`, `range()` та інші. Вони використовуються для виконання різних завдань, наприклад, виведення даних на екран, отримання довжини рядка, отримання введених даних від користувача тощо.
2. Створення власних функцій: Python дає можливість створювати власні функції зі спеціальними інструкціями та повертати значення. Вони можуть бути повторно використані у різних місцях програми, що полегшує розробку та управління кодом.
3. Модулі та пакети: Python має багато модулів та пакетів, які містять набір функцій, які можна імпортувати та використовувати у своїх програмах. Наприклад, модуль `math` містить різні математичні функції, модуль `datetime` - функції для роботи з датами та часом.
4. Лямбда-функції: Лямбда-функції - це анонімні функції, які можуть мати лише один вираз. Вони зазвичай використовуються там, де потрібна невелика функція без необхідності оголошення окремої функції.
5. Обробка винятків: Python має можливості обробки винятків з допомогою конструкції `try-except`. Функції `try` виконують код, а `except` ловить винятки та дозволяє виконувати необхідні дії, коли виникає помилка.

Функції Python дозволяють розбити складні завдання на менші модулі, полегшуючи тим самим розробку та управління кодом. Вони є потужним інструментом для програмування та вирішення різних задач.

### 1.2. Параметри та аргументи.

Python мова програмування, яка дозволяє передавати параметри та аргументи до функцій. Нижче наведено основні параметри та аргументи, які можна використовувати у Python:

1. Параметри за позицією: Параметри можна передавати у функцію за їхньою позицією. Наприклад:

```
def add_numbers(a, b):  
    return a + b  
result = add_numbers(5, 10)
```

У цьому прикладі `a` та `b` є параметрами функції `add\_numbers`, які приймають значення `5` та `10` відповідно.

2. Параметри за замовчуванням: Параметри можуть мати значення за замовчуванням, які використовуються, якщо вони не передаються в функцію. Наприклад:

```
def greet(name='Guest'):  
    print(f"Hello, {name}!")  
greet() # Виведе "Hello, Guest!"  
greet("John") # Виведе "Hello, John!"
```

У цьому прикладі `name` є параметром функції `greet`, який має значення за замовчуванням "Guest".

4. Параметри з ключовим словом: Параметри можуть бути передані вказуючи їхнє ім'я. Це дозволяє передавати аргументи у будь-якому порядку. Наприклад:

```
def greet(first_name, last_name):  
    print(f"Hello, {first_name} {last_name}!")  
greet(last_name="Doe", first_name="John") # Виведе "Hello, John Doe!"
```

У цьому прикладі параметри `first\_name` та `last\_name` передані з ключовими словами, що дозволяє вказувати їх у різному порядку.

Це лише декілька основних прикладів параметрів та аргументів у Python. Мова має багато інших можливостей, пов'язаних з параметрами, які допомагають у створенні більш гнучких та повторно використовуваних функцій.

## **Тема 4: Введення/виведення та обробка файлів**

### **4.1. Робота зі зчитуванням/записом у файли.**

У Python є кілька способів працювати з файлами. Давайте розглянемо основні операції з файлами: зчитування та запис.

Запис у файл:

Для запису даних у файл використовуйте функцію `write()` або `writelines()`. Ось приклад:

```
# Відкриття файлу у режимі запису
file = open("file.txt", "w")
# Запис даних у файл
file.write("Привіт, світ!")
file.write("Це новий рядок.")
# Закриття файлу
file.close()
```

У цьому прикладі ми створюємо файл з ім'ям `"file.txt"` та записуємо два рядки в цей файл. Перед закриттям файлу обов'язково виконайте цю операцію за допомогою `close()`.

Зчитування з файлу:

Для зчитування даних з файлу використовуйте функцію `read()` або `readlines()`. Ось приклад:

```
# Відкриття файлу у режимі зчитування
file = open("file.txt", "r")
# Зчитування даних з файлу
data = file.read()
print(data)
# Закриття файлу
file.close()
```

У цьому прикладі ми відкриваємо файл з ім'ям `"file.txt"`, зчитуємо його вміст за допомогою `read()` та виводимо на екран. Потім обов'язково закриємо файл.

Утім, кращим підходом є використання ключового слова `with`, яке автоматично закриє файл у кінці його використання:

```
# Запис у файл
with open("file.txt", "w") as file:
    file.write("Привіт, світ!")
# Зчитування з файлу
with open("file.txt", "r") as file:
    data = file.read()
    print(data)
```

Цей підхід забезпечує автоматичне закриття файлу, незалежно від того, що станеться в коді всередині блоку `with`.

Надіюся, ця інформація була корисною для роботи з файлами у Python!

## 4.2. Основи обробки текстових файлів.

Основи обробки текстових файлів в Python включають наступні кроки:

1. Відкриття файлу: Використовуйте функцію `open()` для відкриття текстового файлу у вказаному режимі. Наприклад, `file = open("file.txt", "r")` відкриє файл з назвою "file.txt" у режимі читання ("r").

2. Читання файлу: Використовуйте метод `read()` для зчитування всього вмісту файлу або `readline()` для зчитування одного рядка в кожному виклику. Наприклад, `content = file.read()` зчитає весь вміст файлу у змінну "content".

3. Запис у файл: Використовуйте методи `write()` або `writelines()` для запису даних у файл. Наприклад, `file.write("Hello, World!")` запише рядок "Hello, World!" до файлу.

4. Закриття файлу: Використовуйте метод `close()` для закриття файлу після виконання операцій. Наприклад, `file.close()` закриє файл, який був відкритий раніше.

5. Обробка помилок: Рекомендується використовувати блоки `try-except` для обробки можливих помилок при роботі з файлами. Наприклад:

```
try:
    file = open("file.txt", "r")
    content = file.read()
    # Виконання необхідних операцій зі зчитаним вмістом файлу
except FileNotFoundError:
    print("Файл не знайдено")
finally:
    file.close()
```

Це лише основи обробки текстових файлів в Python. Ви можете використовувати різноманітні методи та функції для виконання більш складних операцій з файлами.

Питання для самоперевірки:

1. Які є основні типи даних в Python?
2. Що таке змінна і як створити змінну в Python?
3. Як зчитати введені дані з клавіатури в Python?
4. Як вивести результат на екран в Python?
5. Як визначити довжину рядка (строки) в Python?
6. Які є основні умовні конструкції в Python?
7. Як створити цикл в Python?
8. Що таке список (list) і як його створити в Python?
9. Як отримати доступ до елементів списку в Python?
10. Як сортувати список в Python?

Тести для перевірки знань студентів:

1. Яка різниця між локальними та глобальними змінними у Python?

a) Локальні змінні визначаються всередині функцій і доступні лише у межах цих функцій, тоді як глобальні змінні визначаються на найвищому рівні програми і доступні у всьому коді.

b) Локальні змінні використовуються у циклах, тоді як глобальні змінні використовуються у умовних конструкціях.

c) Локальні змінні використовуються для збереження даних, тоді як глобальні змінні використовуються для вводу-виводу.

2. Що таке списки у Python?

a) Списки - це не змінний тип даних, який містить у собі послідовність елементів.

b) Списки - це набір різнотипних даних, які можна модифікувати.

c) Списки - це структури даних, що містять у собі відсортовану послідовність елементів.

3. Як можна додати новий елемент до списку у Python?

a) Використовуючи метод add().

- b) Змінивши розмір списку та присвоївши нове значення елементу.
- c) Використовуючи метод `append()`.

4. Що станеться, якщо спробувати використовувати індекс, що виходить за межі списку?

- a) Буде викинуто помилку `IndexError`.
- b) Елемент буде доданий в кінець списку.
- c) Елемент буде видалений зі списку.

5. Яка різниця між операторами `"=="` та `"is"` для порівняння об'єктів у Python?

- a) `"=="` перевіряє, чи об'єкти рівні за значенням, тоді як `"is"` перевіряє, чи об'єкти посилаються на одне місце в пам'яті.
- b) `"=="` перевіряє, чи об'єкти рівні за типом даних, тоді як `"is"` перевіряє, чи об'єкти мають однаковий розмір.
- c) `"=="` перевіряє, чи об'єкти однакові, тоді як `"is"` перевіряє, чи об'єкти виконують одну й ту саму операцію.

### Домашні завдання

1. Завдання для початківців:

- Напишіть програму, яка виводить на екран "Привіт, світ!".
- Створіть змінну з вашим ім'ям та виведіть її значення на екран.
- Розв'яжіть задачу: обчисліть площу прямокутника за його довжиною та шириною (значення вводяться з клавіатури).

2. Завдання середнього рівня:

- Напишіть програму, яка перетворює введену температуру з градусів Цельсія в градуси Фаренгейта та виводить результат.
- Реалізуйте програму, що знаходить суму всіх парних чисел в заданому діапазоні (включаючи границі).
- Напишіть функцію, яка визначає, чи є задане слово паліндромом (слово, яке читається однаково зліва направо та справа наліво).



3. Завдання для більш досвідчених студентів:

- Реалізуйте програму, яка визначає, чи є задане число простим.
- Напишіть функцію, яка сортує список чисел за зростанням за допомогою алгоритму сортування вибором.
- Реалізуйте програму, яка знаходить всі прості числа в заданому діапазоні (включаючи границі) і виводить їх.

### 3.3. Лекція 2: використання Python в наукових цілях

#### Тема 1: NumPy та SciPy для наукових обчислень

##### 1.1. Огляд бібліотек NumPy та SciPy.

NumPy і SciPy є двома основними бібліотеками для наукових обчислень у Python. Вони часто використовуються разом, оскільки взаємодіють між собою та доповнюють одна одну. Давайте можемо прикрати дуже коротко опишемо кожен бібліотеку:

##### **NumPy:**

NumPy є основною бібліотекою числових обчислень у Python. Вона надає масиви N-розмірності, які дозволяють виконувати ефективні операції з числами, масивами та матрицями. З NumPy ви можете виконувати різні математичні операції, статистичні обчислення, роботу з лінійною алгеброю та генерувати псевдовипадкові числа. Ця бібліотека є основою для багатьох інших наукових бібліотек у Python.

##### **SciPy:**

SciPy є бібліотекою, яка розширює функціональність NumPy. Вона надає велику кількість функцій для розв'язання різноманітних наукових та інженерних задач. SciPy включає модулі для оптимізації, обробки образів, обробки сигналів, інтерполяції, алгоритмів лінійної алгебри, статистики та інших операцій. Вона розширює можливості NumPy, дозволяючи вам легко розв'язувати складні наукові задачі за допомогою мінімального написання коду.

Ці дві бібліотеки поєднуються, щоб створити потужні інструменти для наукових обчислень у Python. Вони є незамінними для багатьох наукових досліджень, обробки даних, машинного навчання та багатьох інших областей, де потрібні числові обчислення.

## 1.2. Матричні операції та обчислення.

Матричні операції та обчислення важливі в багатьох областях, включаючи науку про дані, машинне навчання та обробку зображень. У Python існують декілька бібліотек, які надають потужні інструменти для таких обчислень. Декілька з них включають NumPy, SciPy та TensorFlow. NumPy є одним з найпопулярніших пакетів для наукових обчислень у Python. Він надає широкий набір функцій для обробки масивів, включаючи матричні обчислення. З NumPy ви можете виконувати операції над матрицями, як-от додавання, віднімання, множення та ділення. Цей пакет також має функції для обчислення власних значень та власних векторів матриць. SciPy є іншою потужною бібліотекою для наукових обчислень, яка побудована на основі NumPy. Вона надає функції для більш складних математичних обчислень, включаючи обчислення детермінанта матриці, розв'язку лінійних систем рівнянь та екстраполяції. TensorFlow є бібліотекою для машинного навчання, але вона також має потужні функції для обчислення з масивами та матрицями. TensorFlow дозволяє виконувати широкий спектр операцій, включаючи транспонування, згорткові операції та матричне множення. Він також забезпечує можливість для обчислення на графічних процесорах (GPU), що дозволяє прискорити обчислення при великих об'ємах даних. Загалом, ці бібліотеки роблять Python потужним інструментом для матричних обчислень і дозволяють зручно працювати з матрицями та виконувати різноманітні математичні операції.

## **Тема 2: Візуалізація даних**

### 2.1. Використання бібліотеки Matplotlib для візуалізації даних.

Бібліотека Matplotlib є потужним інструментом для візуалізації даних в мові програмування Python. Вона надає широкі можливості для створення різноманітних графіків, діаграм, гістограм, розсіювальних графіків та інших візуалізаційних елементів.

Для початку, потрібно імпортувати бібліотеку Matplotlib у вашому коді:

```
import matplotlib.pyplot as plt
```

Тепер ви можете використовувати різні методи та функції Matplotlib для візуалізації даних. Основними методами є `plot()`, `scatter()`, `bar()`, `hist()`, `pie()` та інші.

Наприклад, для створення лінійного графіку, ви можете використовувати метод `plot()`:

```
x = [1, 2, 3, 4, 5]
```

```
y = [10, 8, 6, 4, 2]
```

```
plt.plot(x, y)
```

```
plt.xlabel('X-axis')
```

```
plt.ylabel('Y-axis')
```

```
plt.title('Line Graph')
```

```
plt.show()
```

Це створить вам графік, що відображається у вікні.

Аналогічно, ви можете використовувати інші методи та функції для створення різноманітних видів графіків та діаграм.

Зверніть увагу, що Matplotlib має багато налаштувань та параметрів для зміни вигляду графіків, таких як колір ліній, шрифт, розміри осей, легенди та багато іншого. Ви можете детальніше ознайомитись з цими можливостями у [документації Matplotlib](<https://matplotlib.org/stable/contents.html>).

Загалом, за допомогою бібліотеки Matplotlib ви зможете створювати ефективні та професійні візуалізації даних для вашого Python-проекту.

## 2.2. Побудова графіків та діаграм.

Для побудови графіків та діаграм у Python можна використовувати різні бібліотеки, такі як Matplotlib, Seaborn, Plotly, та інші.

Ось приклад використання бібліотеки Matplotlib для побудови простого графіку:

```
import matplotlib.pyplot as plt
# Зразки даних
x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]
# Побудова графіку
plt.plot(x, y)
# Налаштування вісей та заголовка
plt.xlabel('x')
plt.ylabel('y')
plt.title('Графік залежності y від x')
# Виведення графіку
plt.show()
```

Цей код створить простий лінійний графік залежності y від x. Ви можете налаштувати його за допомогою різних методів та параметрів бібліотеки Matplotlib.

Для побудови діаграм, наприклад, кругової діаграми, можна використати наступний код з використанням бібліотеки Matplotlib:

```
import matplotlib.pyplot as plt
# Зразки даних для діаграми
labels = ['A', 'B', 'C', 'D']
sizes = [15, 30, 45, 10]
```

```
# Побудова кругової діаграми
plt.pie(sizes, labels=labels, autopct='%1.1f%%')
# Виведення діаграми
plt.show()
```

Цей код побудує кругову діаграму з вказаними мітками та розмірами сегментів. Процентне значення кожного сегмента відобразатиметься автоматично завдяки параметру 'autopct'.

Не забудьте встановити необхідні бібліотеки перед виконанням коду. Ви можете використовувати `pip` для встановлення бібліотек, наприклад:

```
pip install matplotlib
```

### **Тема 3: Pandas для роботи з даними**

#### 3.1. Огляд бібліотеки Pandas.

Бібліотека Pandas - це потужний інструмент надходження та аналізу даних у мові програмування Python. Вона надає широкий набір функцій і методів для ефективної роботи з табличними даними.

Основні можливості бібліотеки Pandas включають:

1. Структури даних: Pandas пропонує дві основні структури даних - Series і DataFrame. Series представляє одномірний масив з індексами, тоді як DataFrame - двовимірна таблична структура даних, схожа на звичайну таблицю бази даних. Ці структури дозволяють зручно виконувати операції над даними.

2. Зчитування та запис даних: Pandas надає можливість зчитувати дані з різних джерел, таких як CSV-файли, Excel-файли, бази даних та інші. Вона також дозволяє зберігати дані у цих форматах. Це дає зручність при роботі з реальними наборами даних.

3. Маніпуляції з даними: Pandas надає різноманітні методи для маніпуляцій з даними, таких як фільтрація, сортування, групування, агрегування,

об'єднання та розподіл даних. Ці функції допомагають виконувати різні операції з даними без необхідності складного коду.

4. Обробка пропущених даних: Pandas надає засоби для роботи з пропущеними значеннями у наборах даних. Вона дозволяє заповнювати пропуски або видаляти рядки з пропущеними значеннями, щоб забезпечити чистоту та консистентність даних.

5. Візуалізація даних: Pandas співпрацює з іншою популярною бібліотекою для візуалізації, такою як Matplotlib, щоб допомогти створювати графіки, діаграми та інші візуальні зображення з даними.

Бібліотека Pandas є незамінним інструментом для аналізу та обробки даних у Python. Вона дозволяє виконувати складні операції з легкістю та забезпечує гнучкість при роботі з різними типами даних. Рекомендую ознайомитись з офіційною документацією Pandas, щоб детальніше дослідити всі можливості цієї бібліотеки.

### 1.3. Зчитування та обробка даних у форматі DataFrame.

Обробка даних у форматі DataFrame в Python може бути здійснена за допомогою бібліотеки pandas. Для початку потрібно імпортувати цю бібліотеку:

```
import pandas as pd
```

Потім можна створити DataFrame з різних джерел даних, таких як списки, словники або CSV-файли. Наприклад, якщо у вас є списки з даними, ви можете створити DataFrame наступним чином:

```
data = {'Name': ['John', 'Alice', 'Bob'],  
       'Age': [25, 28, 30],  
       'City': ['New York', 'Paris', 'London']}  
df = pd.DataFrame(data)
```

Цей код створить DataFrame з трьома стовпцями: "Name", "Age" і "City".  
Кожен стовпець буде мати відповідні значення зі списків.

Також ви можете завантажити дані з CSV-файлу за допомогою функції `read_csv()`:

```
df = pd.read_csv('data.csv')
```

Тут 'data.csv' - це назва вашого файлу з даними.

Після цього можна виконувати різні операції з DataFrame, наприклад, відбирати певні рядки або стовпці, змінювати значення, додавати нові стовпці, сортувати дані тощо. Також можна використовувати функції pandas для агрегації, групування і обчислення статистики.

Це лише початок роботи з DataFrame в Python за допомогою бібліотеки pandas. Подальші деталі та операції можна знайти у документації pandas: <https://pandas.pydata.org/>.

## **Тема 4: Введення в машинне навчання з використанням scikit-learn**

### 4.1. Огляд бібліотеки scikit-learn та її можливостей.

Бібліотека scikit-learn є однією з найпопулярніших бібліотек для машинного навчання та аналізу даних у мові програмування Python. Вона надає широкий спектр інструментів для класифікації, регресії, кластеризації, зниження розмірності, роботи з текстом та багатьох інших завдань машинного навчання.

Основні переваги scikit-learn:

1. Простота використання: scikit-learn надає чистий та легкий для розуміння інтерфейс API, який спрощує розробку та використання моделей машинного навчання.



2. Велика кількість реалізованих алгоритмів: бібліотека містить реалізації багатьох популярних алгоритмів машинного навчання, таких як будь-які види дерев рішень, методи опорних векторів, наївний Баєс, k-середніх і багато інших.

3. Підтримка препроцесингу та валідації даних: scikit-learn має вбудовані інструменти для попереднього опрацювання даних, які включають масштабування, відбір ознак, кодування категоріальних змінних та перевірку якості моделей.

4. Інтеграція з іншими бібліотеками Python: scikit-learn поєднується з іншими популярними бібліотеками Python, такими як NumPy і pandas, для спрощення роботи з даними та векторно-матричними операціями.

5. Інструкції та приклади: scikit-learn надає добре документовані API та широкий набір прикладів використання, які допомагають користувачам вивчити бібліотеку та зрозуміти її можливості.

Загалом, scikit-learn є потужним інструментом для розв'язання задач машинного навчання та аналізу даних у мові програмування Python. Вона допомагає в розробці та випробуванні моделей, валідації результатів і підтримці повного процесу розробки моделі.

#### 4.2. Базовий аналіз даних та класифікація.

Базовий аналіз даних та класифікація є важливими концепціями в аналітиці даних за допомогою мови програмування Python. Ці концепції дозволяють отримати цінну інформацію з наборів даних та класифікувати їх для подальшого використання.

У Python існує кілька потужних бібліотек, які допомагають здійснювати базовий аналіз даних та класифікацію. Найпопулярніші з них:

1. NumPy: Це основна бібліотека для наукових обчислень в Python. Вона надає підтримку для обробки багатовимірних масивів, математичних функцій і операцій лінійної алгебри.

2. Pandas: Ця бібліотека надає зручні інструменти для роботи з даними. Вона дозволяє читати, записувати та маніпулювати даними у вигляді датафреймів – структур, які схожі на таблиці з бази даних.

3. Scikit-learn: Ця бібліотека містить алгоритми машинного навчання, включаючи класифікацію. Вона дозволяє виконувати різноманітні класифікаційні завдання, такі як логістична регресія, дерева рішень, випадковий ліс і багато інших.

4. Matplotlib: Ця бібліотека дозволяє створювати візуалізації даних, такі як графіки і діаграми. Вона допомагає аналізувати та представляти результати класифікації в зрозумілій формі.

Ці бібліотеки разом з Python надають потужні засоби для базового аналізу даних та класифікації. Ви можете почати з вивчення їх документації, пробувати приклади та поглиблюватися в аналітику даних за допомогою Python.

## **Тема 5: Застосування Python в наукових дослідженнях**

### **5.1. Приклади конкретних досліджень, де використовується Python.**

Ось кілька прикладів конкретних досліджень, де використовується мова програмування Python:

1. Обробка та аналіз даних: Python широко використовується для обробки та аналізу даних у наукових дослідженнях. Наприклад, дослідники можуть використовувати пакети, такі як NumPy та pandas, для роботи з числовими даними, або бібліотеку matplotlib для візуалізації результатів.

2. Машинне навчання та штучний інтелект: Python є однією з найпопулярніших мов програмування для розробки моделей машинного навчання та штучного інтелекту. Бібліотеки, такі як TensorFlow, PyTorch та scikit-learn, надають потужні інструменти для тренування моделей, розпізнавання образів, обробки природної мови та багато іншого.

3. Обробка зображень та комп'ютерне зорове обчислення: Python дозволяє дослідникам працювати з більшими обсягами зображень та застосовувати алгоритми обробки зображень. Бібліотеки, такі як OpenCV та scikit-image, надають функції для розпізнавання об'єктів, виявлення особливостей та обробки зображень.

4. Біоінформатика: Python є популярною мовою програмування у біоінформатиці. Використовується для аналізу геномних даних, розробки біоінформатичних алгоритмів та моделей машинного навчання для виявлення генетичних залежностей.

5. Астрофізика: Python використовується в астрофізиці для обробки та аналізу космічних даних. Бібліотека Astropy надає інструменти для роботи зі створенням та аналізом астрономічних даних, розрахунків космічних моделей та візуалізації результатів.

Це лише кілька прикладів досліджень, де Python використовується в якості мови програмування. Все більше наукових галузей використовують Python для своїх досліджень завдяки його простоті використання та широкому спектру бібліотек.

## 5.2. Важливість автоматизації та швидкодії у наукових розрахунках.

Автоматизація та швидкодія є надзвичайно важливими аспектами у наукових розрахунках, особливо використовуючи мову програмування Python.

Автоматизація дозволяє виконувати повторювані завдання, що займають багато часу, швидше та ефективніше. Завдяки Python та його багатим набором бібліотек для наукових обчислень, таких як NumPy, SciPy, Pandas та інші, ви можете легко автоматизувати обчислення та обробку даних. Це дозволяє вам зосередитися на самому дослідженні, а не на рутинних операціях.

Швидкодія також має велике значення, особливо коли маєте справу з великими обсягами даних або складними математичними розрахунками. Python, завдяки своїй ефективній реалізації та можливості використання прискорювачів,

таких як бібліотека NumPy, може запускати обчислення швидше, ніж багато інших мов програмування.

Також варто зазначити, що Python є легко зрозумілою та доступною мовою програмування. Це важливо, оскільки наукові розрахунки часто виконуються командою учених з різним рівнем програмування. Простота Python дозволяє широкому колу науковців використовувати його для своїх потреб без значних зусиль.

Загалом, автоматизація та швидкодія у наукових розрахунках на прикладі Python допомагають скоротити час, затрачений на операції обробки даних та обчислень, що дозволяє більше часу приділити самому науковому дослідженню та аналізу результатів.

### 3.4. Питання для самоперевірки

1. Як Python використовується в наукових дослідженнях?
2. Які основні бібліотеки Python використовуються для наукових цілей?
3. Як Python сприяє обробці та аналізу наукових даних?
4. Як можна використовувати Python для візуалізації наукових результатів?
5. Які наукові галузі найбільше користуються Python?
6. Чим Python відрізняється від інших мов програмування в контексті наукових досліджень?
7. За допомогою яких інструментів Python можна моделювати наукові експерименти?
8. Як Python допомагає в розробці машинного навчання для наукових цілей?
9. Як Python сприяє автоматизації наукових процесів?
10. Які ресурси або документація рекомендуються для початківців, які хочуть вивчити Python для наукових цілей?

Created by <https://GPTGO.ai>

#gptgo #chatgpt #freechatgpt #chatgptfree

Тестові завдання для перевірки знань студентів:

1. Які основні бібліотеки Python використовуються для наукових обчислень?
  - a) NumPy
  - b) SciPy
  - c) Pandas
  - d) Matplotlib

2. Які можливості забезпечує бібліотека NumPy?

- a) Обробка масивів чисел
- b) Робота з графіками
- c) Виконання статистичних аналізів
- d) Робота з базами даних

3. Яку функцію використовують для імпорту бібліотеки NumPy у Python?

- a) `import numpy as np`
- b) `import num_py as npy`
- c) `import numpy as numpy`
- d) `import numericpy as np`

4. Яка функція використовується для створення графіків засобами бібліотеки Matplotlib?

- a) `plot()`
- b) `scatter()`
- c) `hist()`
- d) all of the above (всі вищезазначені)

5. Яким чином можна виконувати векторні обчислення у Python за допомогою бібліотеки NumPy?

- a) Використовуючи цикли `for`
- b) Використовуючи готові функції NumPy
- c) Використовуючи розгалуження `if-else`
- d) Використовуючи рекурсію

Домашні завдання:

1. Завдання: Напишіть програму на Python, яка використовує модуль `numpy` для обчислення середнього значення списку чисел.

2. Завдання: Реалізуйте програму на Python, яка обчислює факторіал заданого числа, використовуючи рекурсію.
3. Завдання: Напишіть програму на Python для обчислення суми всіх парних чисел у заданому діапазоні (включаючи крайні значення).

**Наукове видання**

**ПРАКТИЧНІ КЕЙСИ  
ШТУЧНИЙ ІНТЕЛЕКТ У ДІЯЛЬНОСТІ ПЕДАГОГА**

**ЗБІРНИК МАТЕРІАЛІВ**

***Випуск 1 (2)***

Матеріали надруковані в авторській редакції.  
За достовірність фактів, посилань, стилістичне та орфографічне оформлення відповідальність несуть автори публікацій.  
Факти та посилання, надані сервісами штучного інтелекту, не перевіряються.

Відповідальний за збірник: Богачков Ю. М.

Комп'ютерна верстка: Ухань П. С.

Інститут цифровізації освіти  
Національної академії педагогічних наук України  
м. Київ, вул. Максима Берлінського, 9  
Свідоцтво про державну реєстрацію:  
серія ДК №7609 від 23.02.2022 р.  
електронна пошта (E-mail): [iitlt@iitlt.gov.ua](mailto:iitlt@iitlt.gov.ua)