

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення
Дисципліна: Скриптові мови програмування (Python)

Лабораторна робота №7
Тема: «ПОБУДОВА ГРАФІКІВ МАТЕМАТИЧНИХ ФУНКЦІЙ У
МОВІ PYTHON»

Виконав: ст. гр. КН-24
Куріщенко П. В.
Перевірив: ассистент
Ткаченко О.С.

Кропивницький 2025

Варіант - 16

Мета роботи - набути навичок роботи з бібліотекою Matplotlib для візуалізації даних.

ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ

Завдання 1:

Зобразити 2d графік функції відповідно своєму варіанту (№16 - $Y(x)=\cos(x^2)/x$, $x=[0...5]$) та зберегти у .png файл.

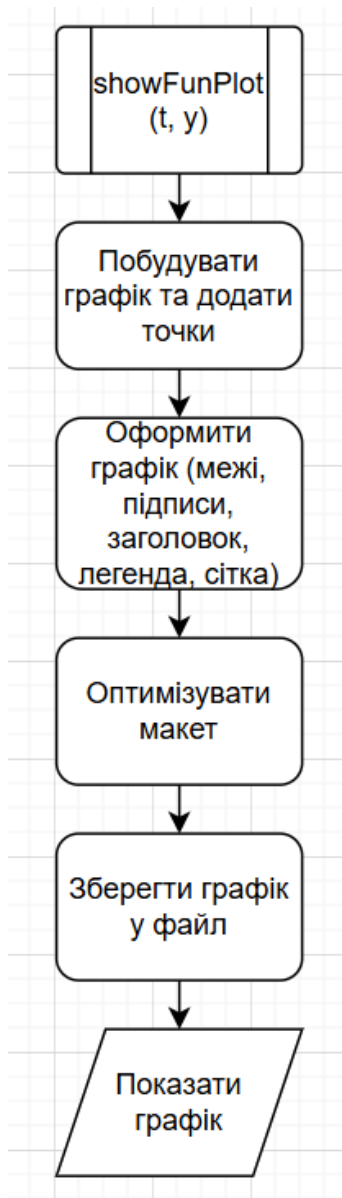
Принцип роботи функції **showFunPlot(t, y):**

Функція showFunPlot(t, y) будує графік залежності у від t, додає точки даних, підписує осі, назву, легенду та сітку. Зберігає графік у файл і відображає його.

Проектні рішення:

- Комбінує лінійний графік і точки для наочності.
- Задає межі осей для контрольованого вигляду.
- Використовує `tight_layout()` для коректного розміщення елементів.
- Зберігає графік у папку `../Plots/` з високою роздільною здатністю (`dpi=200`).
- Додає оформлення (сітка, легенда) для покращення читабельності.

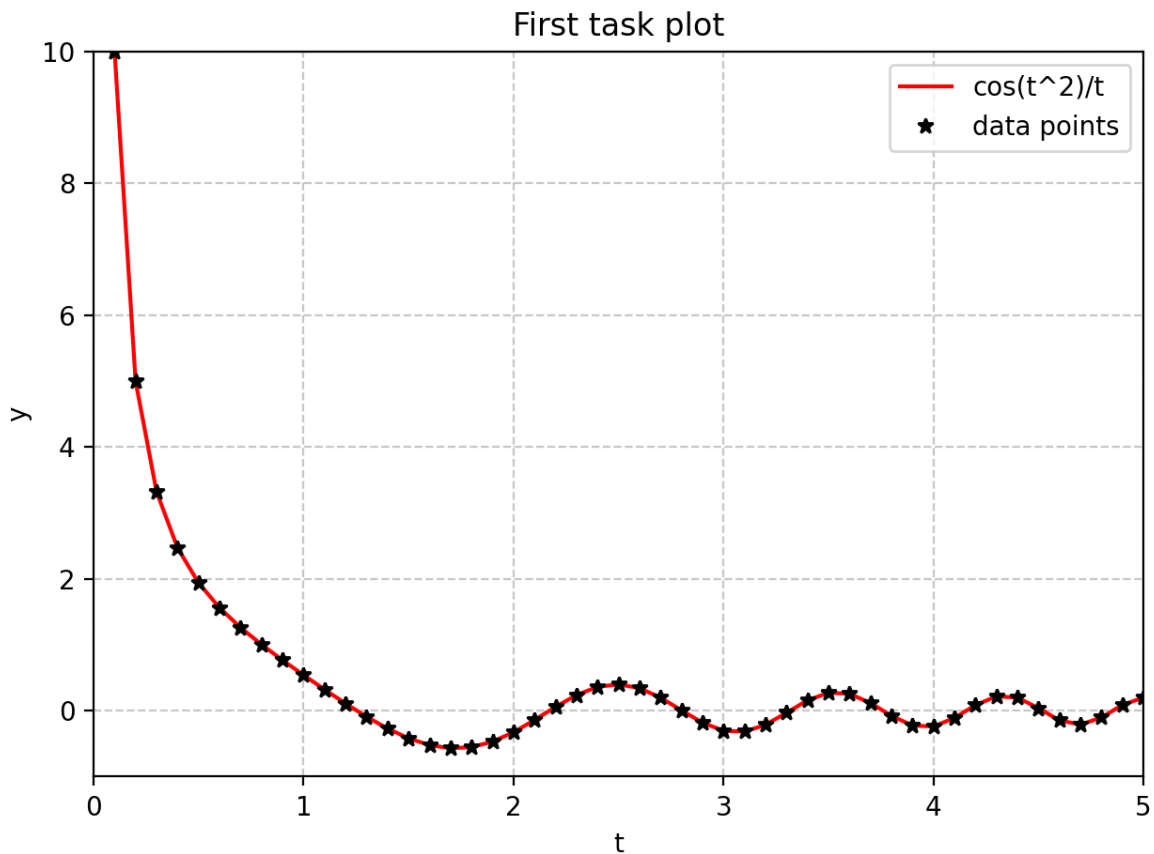
Блок схема:



Лістинг функції:

```
def showFunPlot(t, y):  
    plt.plot(t, y, 'r-', label='cos(t^2)/t')  
    plt.plot(t, y, 'k*', label='data points')  
    plt.axis([0, 5, -1, 10])  
    plt.xlabel('t')  
    plt.ylabel('y')  
    plt.title('First task plot')  
    plt.legend()  
    plt.grid(True, linestyle='--', alpha=0.7)  
    plt.tight_layout()  
    plt.savefig('../Plots/firstTaskPlot.png', dpi=200)  
    plt.show()
```

Результат виконання:



Завдання 2:

Зобразити гістограму частоти появи літер у певному тексті та зберегти у .png файл.

Принцип роботи функції `showLetterFreqPlot(text)`:

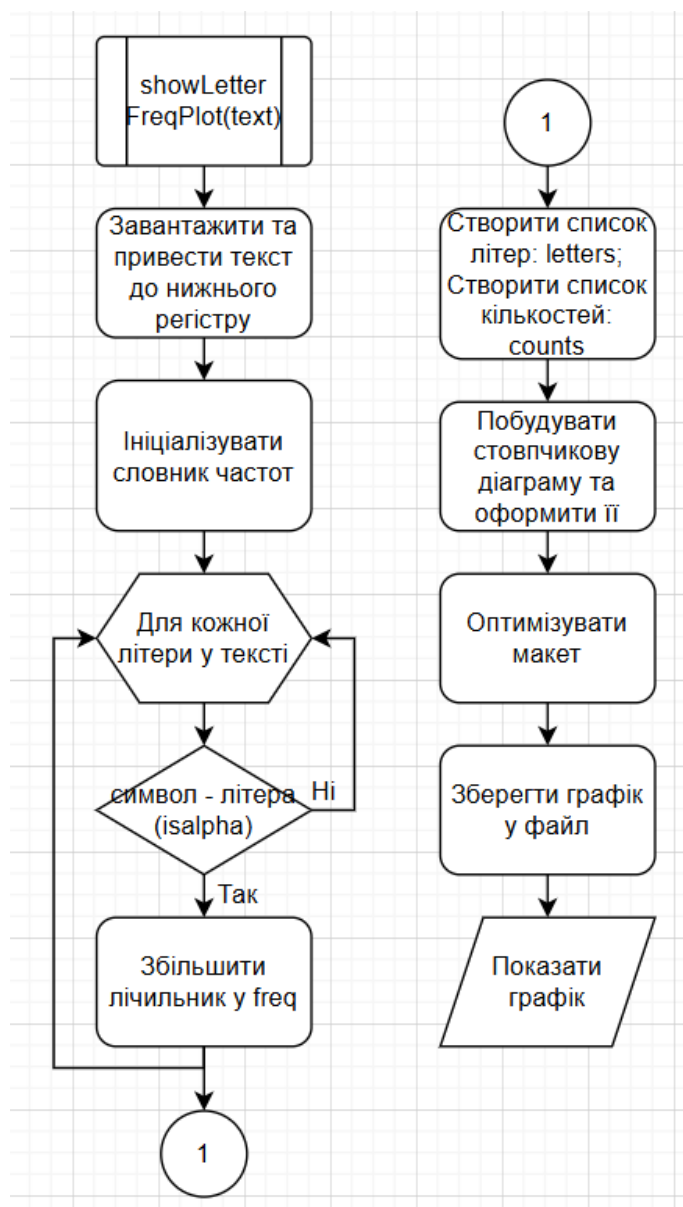
Функція `showLetterFreqPlot(text)` завантажує текст із файлу, підраховує частоти літер (ігноруючи неалфавітні символи), будує стовпчикову діаграму й зберігає її у файл.

Проектні рішення:

- `autoLoad()` дає змогу автоматично підвантажити текст із файлу.
- Використання `isalpha()` гарантує врахування лише літер.
- Графік оформлено з підписами осей і назвою для зручності.

- Зберігання у ../Plots/ з високою якістю (dpi=200).
- `tight_layout()` забезпечує правильне розміщення без обрізки.

Блок схема:



Лістинг функції:

```
def showLetterFreqPlot(text):
    text = autoLoad("test_sentence.txt", text).lower()
    freq = {}
    for letter in text:
        if letter.isalpha():
            freq[letter] = freq.get(letter, 0) + 1
```

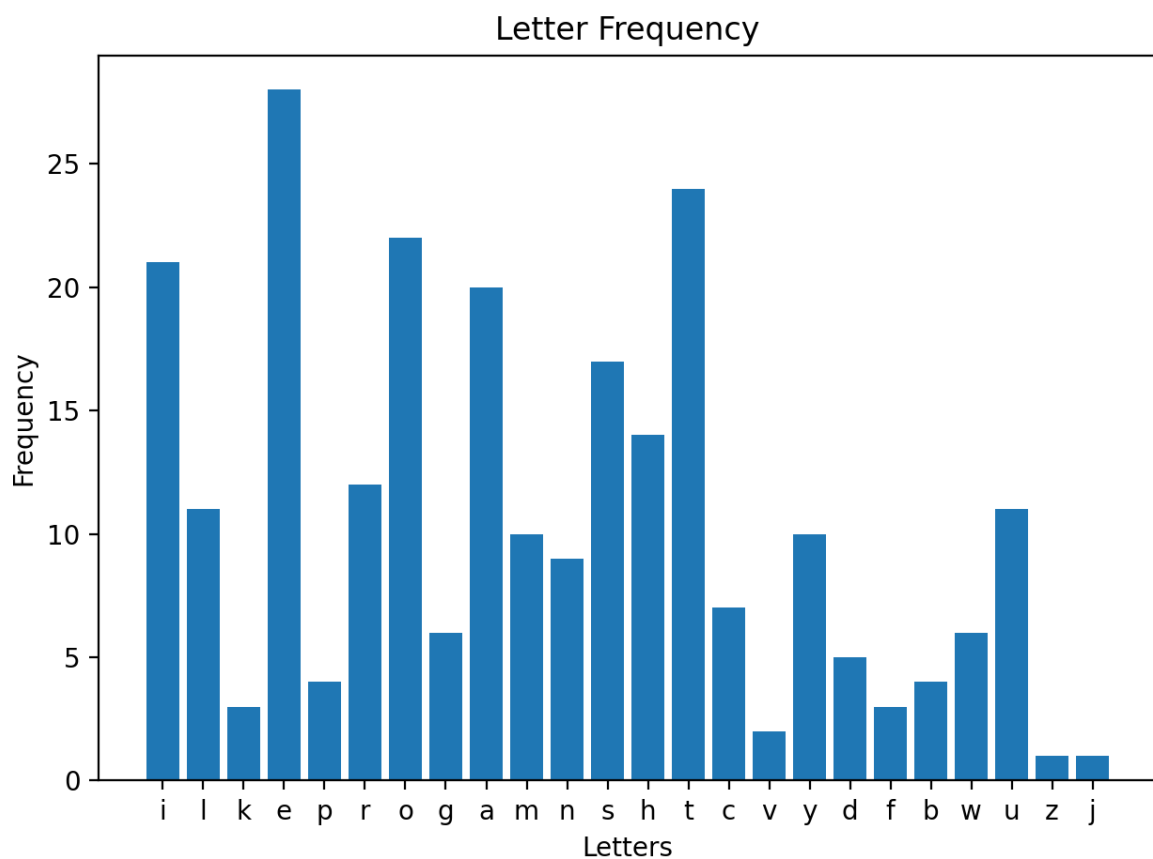
```

letters = list(freq.keys())
counts = list(freq.values())

plt.bar(letters, counts)
plt.xlabel('Letters')
plt.ylabel('Frequency')
plt.title('Letter Frequency')
plt.tight_layout()
plt.savefig('../Plots/secondTaskPlot.png', dpi=200)
plt.show()

```

Результат виконання:



Завдання 3:

Зобразити гістограму частоти появи у певному тексті звичайних, питальних та окличних речень, а також речень, що завершуються трикрапкою та зберегти у .png файл.

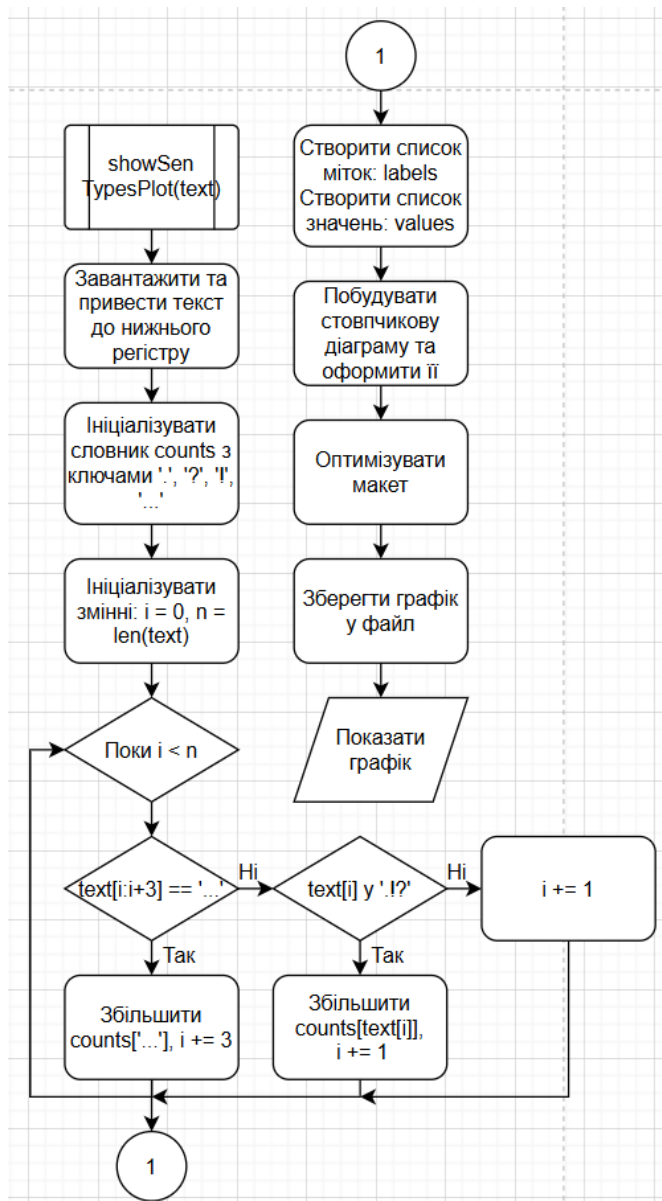
Принцип роботи функції **showSenTypesPlot(text)::**

- Функція `showSenTypesPlot(text)` рахує кількість речень за розділовими знаками (., ?, !, ...) у тексті, будує стовпчикову діаграму та зберігає її у файл.

Проектні рішення:

- `autoLoad()` дозволяє підвантажити текст із файлу.
- Ручний парсинг символів дає змогу точно відрізнити ... від трьох окремих крапок.
- Діаграма має кольорове оформлення, підписи осей і заголовки.
- Зберігання у `../Plots/` з високою якістю (`dpi=200`).
- `tight_layout()` гарантує коректне відображення без обрізання.

Блок схема:



Лістинг функції:

```

def showSenTypesPlot(text):
    text = autoLoad("test_sentence.txt", text).lower()
    counts = {'.': 0, '?': 0, '!': 0, '...': 0}

    i = 0
    n = len(text)
    while i < n:
        if text[i:i+3] == '...':
            counts['...'] += 1
            i += 3
        elif text[i] in '!.?':
            counts[text[i]] += 1
            i += 1
  
```



```

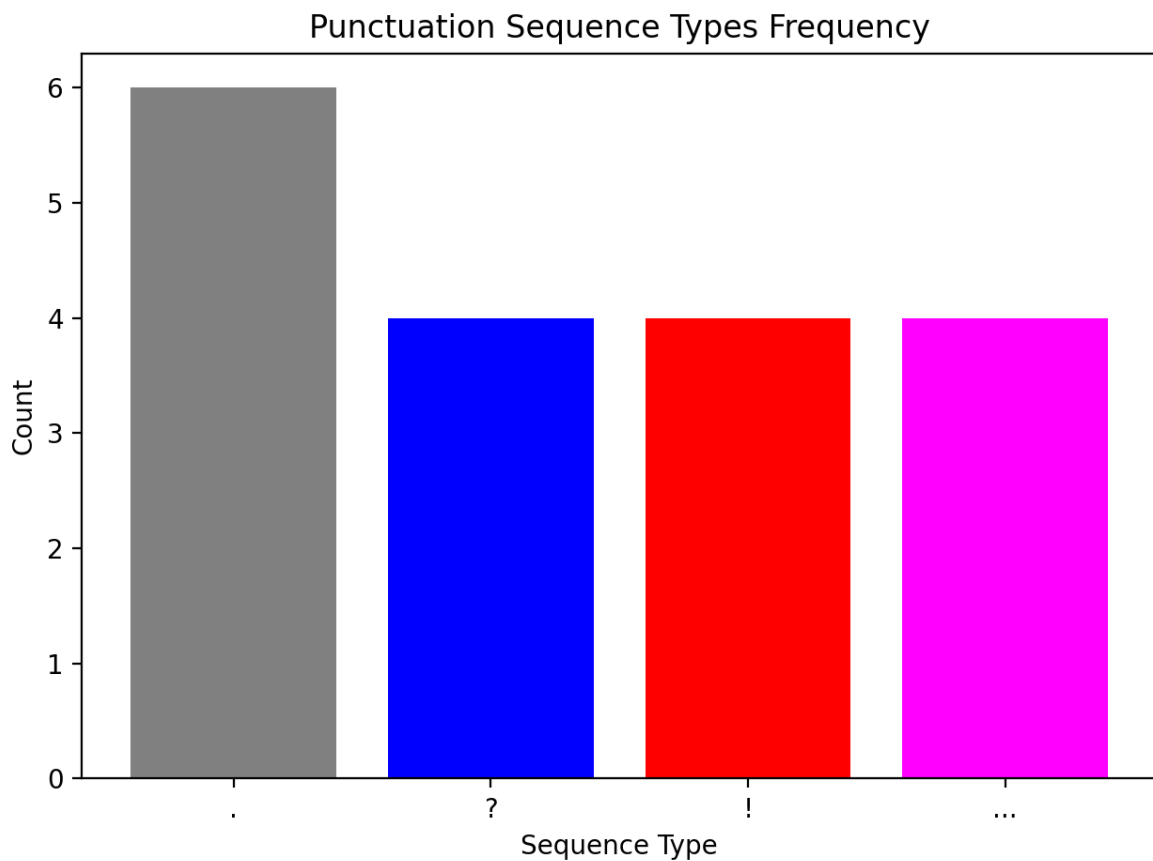
        else: i += 1

labels = [k for k in counts if counts[k] > 0]
values = [counts[k] for k in labels]

plt.bar(labels, values, color=['gray', 'blue', 'red',
'magenta'][:len(labels)])
plt.xlabel('Sequence Type')
plt.ylabel('Count')
plt.title('Punctuation Sequence Types Frequency')
plt.tight_layout()
plt.savefig('../Plots/thirdTaskPlot.png', dpi=200)
plt.show()

```

Результат виконання:



КОНТРОЛЬНІ ЗАПИТАННЯ

1. Які засоби мова Python надає для роботи з 2D графікою? Які бібліотеки призначені для роботи з графікою?

Python надає зручні інструменти для побудови 2D графіки за допомогою бібліотек:

- **Matplotlib** — найпопулярніша бібліотека для побудови графіків, діаграм та гістограм.
- **NumPy** — використовується разом із Matplotlib для математичних обчислень, зокрема для створення масивів значень, наприклад: `t = linspace(0.1, 5, 1000)`.

У функції `showFunPlot(t, y)` використовуються ці бібліотеки:

- `matplotlib.pyplot` для побудови графіка та збереження зображення,
- `numpy` може бути використаний для генерації значень `t` у вигляді рівномірного масиву.

2. Яким чином можна відобразити графік математичної функції?

Щоб побудувати графік, потрібно:

- мати масив значень `x` (наприклад `t`)
- обчислити значення $y = f(x)$
- використати `plt.plot(t, y)`

У функції `showFunPlot(t, y)` графік будується за значеннями $\cos(t^2)/t$:

```
plt.plot(t, y, 'r-', label='cos(t^2)/t')
```

3. Як можна налаштувати колір та тип лінії на графіку математичної функції?

Колір і стиль лінії задаються як строковий аргумент у функції `plot()`.

Наприклад:

- `'r-'` — червона лінія
- `'k*'` — чорні зірочки

У `showFunPlot` це виглядає так:

```
plt.plot(t, y, 'r-', label='cos(t^2)/t') # червона лінія
plt.plot(t, y, 'k*', label='data points') # чорні зірочки
```

4. Яким чином можна відобразити гістограму?

Для побудови стовпчикової діаграми (гістограми) використовують `plt.bar(x, y)`, де `x` — підписи (наприклад, літери або символи), а `y` — кількість.

У функціях `showLetterFreqPlot` та `showSenTypesPlot` гістограма створюється так:

```
plt.bar(letters, counts)          # частоти літер
plt.bar(labels, values, color=...)# типи речень (. ? !
...)
```

5. Яким чином можна зберегти зображення у файл?

Використовують функцію `plt.savefig("шлях/до/файлу.png", dpi=якість)`.

У всіх трьох функціях використовується:

```
plt.savefig('../Plots/secondTaskPlot.png', dpi=200)
```

Це дозволяє зберегти графік у форматі `.png` з високою якістю.