МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Кафедра систем управління літальними апаратами

Лабораторна робота № 5

з дисципліни «Об'єктно-орієнтоване програмування СУ»

Тема: ««Розробка графічного інтерфейсу для розрахункових завдань

і побудови графіків»

ХАІ.301 .174. 312ст.5 ЛР

Виконав студент гр. _____312ст_____

_____Твердохліб Максим Анатолійович
(підпис, дата) (П.І.Б.)

Перевірив
_____ к.т.н., доц. О. В. Гавриленко
_____ ас. В. О. Білозерський
(підпис, дата) (П.І.Б.)

2024

МЕТА РОБОТИ

Застосувати теоретичні знання з основ роботи з бібліотекою tkinter на мові Python, навички використання бібліотеки matplotlib, а також об'єктноорієнтований підхід до проектування програм, і навчитися розробляти скрипти для інженерних додатків з графічним інтерфейсом.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Описати клас, який реалізує графічний інтерфейс користувача для вирішення розрахункової задачі згідно варіанту і скрипт для роботи з об'єктом цього класу. Зазначена у задачі функція повинна бути окремим методом класу.

Func31. Описати функцію Swap(X, I, J), що змінює вміст дійсних елементів XI та XJ списку X (I та J – параметри цілого типу; функція повертає None). З її допомогою для списку чотирьох даних елементів послідовно поміняти вміст двох перших, двох останніх і двох середніх елементів, після чого вивести нові значення елементів списку.

Завдання 2. Розробити скрипт із графічним інтерфейсом, що виконує наступні функції:

8	$y[k+2] = \left(2 - \frac{2 \cdot \xi \cdot T_0}{T}\right) \cdot y[k+1] + \left(\frac{2 \cdot \xi \cdot T_0}{T} - 1 - \frac{T_0^2}{T^2}\right) \cdot y[k] + \frac{K \cdot T_0^2}{T^2} \cdot U$	U [0] = 0.1 рад / с, y [0] == y [1] = 0	$T = 0{,}3$ $K = 2$ $\xi = 0{,}5$	y – υ, рад U – δ_B, рад

A. установка початкових значень параметрів для побудови графіка (змінні Tkinter)

B. створення текстового файлу з двома стовпцями даних: аргумент і значення функції відповідно до варіанту (див. табл.2). Роздільник в кожному рядку файлу: для парних варіантів – ';', для непарних – '#';

C. зчитування з файлу масивів даних;

D. підрахунок і відображення мінімального / максимального значення аргументу / функції у зчитаних масивах;

E. відображення масивів даних за допомогою пакета matplotlib у вигляді графіка функції в декартовій системі координат з назвою функції,

позначенням осей, оцифруванням і сіткою;

F. заголовок вікна повинен містити текст текст:

lab # - <# групи> -v <# варіанту> - <прізвище> - <ім'я>, наприклад:

lab4_2-320-v01-Ivanov-Ivan

Набір і розташування віджетів слід спроектувати таким чином, щоб інтерфейс був максимально дружнім:

• всі поля для введення повинні супроводжуватися відповідними текстовими мітками;

• ніяка послідовність дій не повинна призводити до системних помилок (в командному вікні);

• при виникненні помилок повинно бути виведено відповідні повідомлення;

• при зміні розмірів основного вікна, всі елементи управління повинні також підлаштовуватися.

Код в лістингу програм повинен містити докладні коментарі!

У звіті повинні бути дві діаграми класів зі специфікаціями (відповідальність класу, опис атрибутів, опис методів) і дві діаграми активності для 1) методу, що реалізує обчислення в завданні 1, і 2) методу, що

реалізує відображення графіка функції в завданні 2.

Рекомендації до виконання завдання 2:

У текстовому файлі кожна пара цифр: значення аргументу (по осі X), роздільник, значення функції (по осі У), наприклад:

0 :: 0

0.005 :: 0.71618

0.01 :: 1.3852

0.015 :: 1.6665

0.02 :: 1.479

0.025 :: 1.0432

0.03 :: 0.67931

0.035 :: 0.59063

0.04 :: 0.76774

0.045 :: 1.0428

Аргументом є час: t [k] = kT0, T0 = 2T / N, N = [20..1000] – кількість точок треба підібрати, щоб графік був гладким. Функція являє собою характеристику одного з об'єктів управління:

• кут тангажа літака – υ, рад

• кутова швидкість обертання електродвигуна – ω, рад/с

• температура термостата – T,K

Виконання роботи

Завдання 1. Виконання задачі Func 31

Вхідні дані:

Ім’я змінної	Опис	Тип	Обмеження
X	Список з чотирьох дійсних чисел	list[float]	Повинен містити рівно чотири дійсні числа
I	Індекс першого елемента, який потрібно поміняти місцями	int	Повинен бути в межах `[0, len(X)-1]`
J	Індекс другого елемента, який потрібно поміняти місцями	int	Повинен бути в межах `[0, len(X)-1]`

Вихідні дані:

Ім’я змінної	Опис	Тип
X	Оновлений список після виконання функції `Swap`	list[float]
message	Підтвердження виконання або повідомлення про помилку	str

Алгоритм вирішення задачі показано на рис. 1

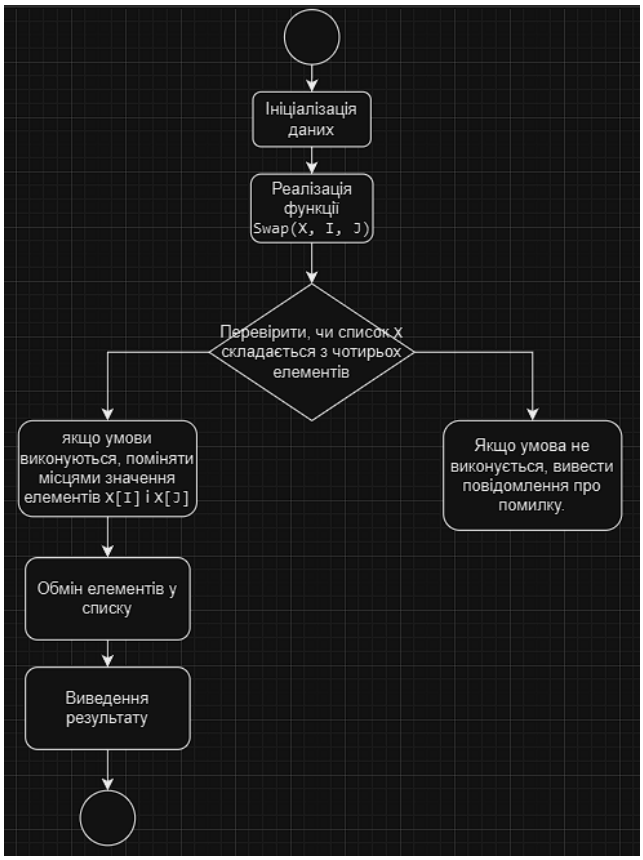

Рисунок 1 – Алгоритм вирішення задачі 1

Лістинг коду вирішення задачі наведено в дод. А. стор. 14 Екран роботи наведено в дод. Б.1 стор. 15

Завдання 2. Вирішення задачі 8

Вхідні дані:

Ім’я	Опис	Тип	Обмеження
entry_T	Період (T)	Число (float)	T>0
entry_K	Коефіцієнт (K)	Число (float)	Будь-яке дійсне число
entry_xi	Демпфування (ξ)	Число (float)	Будь-яке дійсне число
entry_U0	Початкове значення (U[0])	Число (float)	Будь-яке дійсне число
entry_y0	Початкове значення (y[0])	Число (float)	Будь-яке дійсне число
entry_y1	Початкове значення (y[1])	Число (float)	Будь-яке дійсне число

Вихідні дані:

Ім’я	Опис	Тип
data_x	Список індексів k	Список (int)
data_y	Список значень y[k]	Список (float)
error_message	Повідомлення про помилки	Текст (str)
graph	Графік функції k та y[k]	Графічний об'єкт

Алгоритм вирішення задачі показано на рис. 2

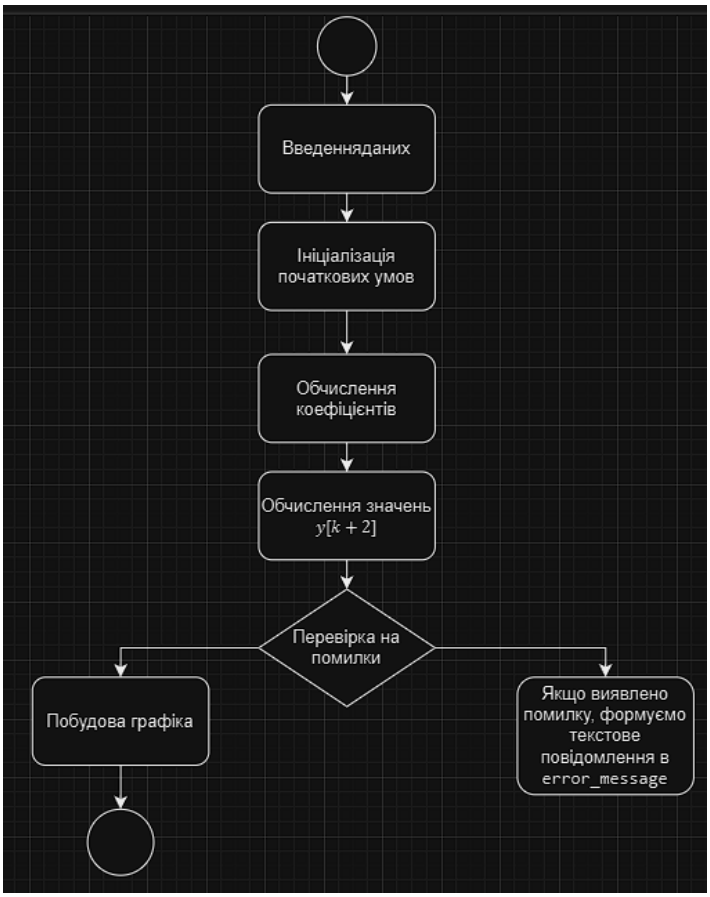

Рисунок 2 – Алгоритм вирішення задачі 8

Лістинг коду вирішення задачі наведено в дод. А стор. 14.Екран роботи програми показаний в дод. Б.2 стор. 15

Висновки:

Під час виконання лабораторної роботи було вивчено основи створення графічного інтерфейсу користувача з використанням бібліотеки Tkinter, а також використання бібліотеки matplotlib для побудови графіків.

Додаток А

Лістинг коду програму до завдання Func 31

```
import tkinter as tk
from tkinter import messagebox


def Swap(X, I, J):
    """Змінює місцями елементи XI та XJ у списку X."""
    if I < 0 or J < 0 or I >= len(X) or J >= len(X):
        raise ValueError("Індекси мають бути в межах довжини списку!")
    X[I], X[J] = X[J], X[I]


class SwapApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Cwap елементів списку")

        # Елементи інтерфейсу
        self.label = tk.Label(root, text="Введіть список із 4 елементів через
кому:")
        self.label.pack(pady=5)

        self.entry = tk.Entry(root)
        self.entry.pack(pady=5)

        self.calculate_button = tk.Button(root, text="Поміняти місцями",
command=self.swap_elements)
        self.calculate_button.pack(pady=5)

        self.result_label = tk.Label(root, text="Результат відобразиться
тут.")
        self.result_label.pack(pady=5)

    def swap_elements(self):
        try:
            # Зчитування введеного списку
            input_data = self.entry.get()
            elements = list(map(float, input_data.split(",")))

            if len(elements) != 4:
```

```
                raise ValueError("Список повинен містити рівно 4 елементи!")

            # Використання функції Swap
            Swap(elements, 0, 1)  # Заміна перших двох елементів
            Swap(elements, 2, 3)  # Заміна останніх двох елементів
            Swap(elements, 1, 2)  # Заміна двох середніх елементів

            # Відображення результату
            result_text = f"Новий список: {elements}"
            self.result_label.config(text=result_text)

        except ValueError as e:
            messagebox.showerror("Помилка", str(e))


if __name__ == "__main__":
    root = tk.Tk()
    app = SwapApp(root)
    root.mainloop()
```

Лістинг коду програми до завдання 8

```
import tkinter as tk
from tkinter import filedialog, messagebox
import matplotlib.pyplot as plt

# Глобальні змінні для даних
data_x = []
data_y = []

# Функція для обчислення y[k]
def calculate_values():
    try:
        # Отримуємо вхідні дані з полів
        T = float(entry_T.get())
        K = float(entry_K.get())
        xi = float(entry_xi.get())
        U0 = float(entry_U0.get())
        y0 = 0  # Початкове значення y[0]
        y1 = 0  # Початкове значення y[1]
        N = 100  # Кількість ітерацій
```

```
        if T <= 0:
            raise ValueError("T має бути більше нуля.")

        global data_x, data_y
        data_x = list(range(N))
        data_y = [y0, y1]

        # Коефіцієнти рівняння
        a1 = 2 - 2 * xi * T
        a2 = 2 * xi * T - 1 - T**2
        b = K * T**2

        # Обчислюємо y[k+2] за формулою
        for k in range(2, N):
            y_next = a1 * data_y[k - 1] + a2 * data_y[k - 2] + b * U0
            data_y.append(y_next)

        messagebox.showinfo("Успіх", "Обчислення завершено.")
    except ValueError as e:
        messagebox.showerror("Помилка", f"Помилка в обчисленнях: {e}")
    except Exception as e:
        messagebox.showerror("Помилка", f"Неочікувана помилка: {e}")

# Функція для обчислення мінімуму та максимуму
def calculate_min_max():
    try:
        if not data_x or not data_y:  # Перевірка на порожні масиви
            raise ValueError("Масиви даних порожні.")
        min_x, max_x = min(data_x), max(data_x)
        min_y, max_y = min(data_y), max(data_y)
        messagebox.showinfo("Результати", f"Мінімум X: {min_x}\nМаксимум X:
{max_x}\nМінімум Y: {min_y}\nМаксимум Y: {max_y}")
    except Exception as e:
        messagebox.showerror("Помилка", f"Помилка: {e}")

# Функція для запису даних у файл
def save_to_file():
    try:
        file_path = filedialog.asksaveasfilename(defaultextension=".txt",
filetypes=[("Text files", "*.txt")])
        if not file_path:
            return
```

```
            delimiter = ';'

            with open(file_path, "w") as f:
                for x, y in zip(data_x, data_y):
                    f.write(f"{x}{delimiter} {y}\n")

            messagebox.showinfo("Успіх", "Дані успішно записано у файл.")
        except Exception as e:
            messagebox.showerror("Помилка", f"Помилка: {e}")

# Функція для зчитування даних з файлу
def load_from_file():
    global data_x, data_y
    try:
        file_path = filedialog.askopenfilename(filetypes=[("Text files",
"*.txt")])
        if not file_path:
            return

        with open(file_path, "r") as f:
            lines = f.readlines()

        delimiter = ";" if ";" in lines[0] else "#"
        data_x, data_y = [], []
        for line in lines:
            x, y = map(float, line.strip().split(delimiter))
            data_x.append(x)
            data_y.append(y)

        messagebox.showinfo("Успіх", "Дані успішно зчитано.")
    except Exception as e:
        messagebox.showerror("Помилка", f"Помилка: {e}")

# Функція для побудови графіка
def plot_graph():
    try:
        if not data_x or not data_y:
            raise ValueError("Дані для графіка порожні.")

        plt.figure()
        plt.plot(data_x, data_y, label="y[k]")
```

```
        plt.title("Графік функції y[k]")
        plt.xlabel("k (індекс)")
        plt.ylabel("y[k]")
        plt.grid(True)
        plt.legend()
        plt.show()
    except Exception as e:
        messagebox.showerror("Помилка", f"Помилка: {e}")

# Основна функція для побудови GUI
def main():
    root = tk.Tk()
    root.title("Розв’язання рівняння різницевого оператора")

    tk.Label(root, text="T (період):").grid(row=0, column=0, padx=5, pady=5)
    global entry_T
    entry_T = tk.Entry(root)
    entry_T.grid(row=0, column=1, padx=5, pady=5)

    tk.Label(root, text="K (коефіцієнт):").grid(row=1, column=0, padx=5,
pady=5)
    global entry_K
    entry_K = tk.Entry(root)
    entry_K.grid(row=1, column=1, padx=5, pady=5)

    tk.Label(root, text="ξ (демпфування):").grid(row=2, column=0, padx=5,
pady=5)
    global entry_xi
    entry_xi = tk.Entry(root)
    entry_xi.grid(row=2, column=1, padx=5, pady=5)

    tk.Label(root, text="U[0] (початкове значення):").grid(row=3, column=0,
padx=5, pady=5)
    global entry_U0
    entry_U0 = tk.Entry(root)
    entry_U0.grid(row=3, column=1, padx=5, pady=5)

    tk.Button(root, text="Обчислити y[k]",
command=calculate_values).grid(row=4, column=0, columnspan=2, pady=5)
    tk.Button(root, text="Зберегти у файл", command=save_to_file).grid(row=5,
column=0, columnspan=2, pady=5)
```

```
    tk.Button(root, text="Зчитати з файлу",
command=load_from_file).grid(row=6, column=0, columnspan=2, pady=5)
    tk.Button(root, text="Обчислити min/max",
command=calculate_min_max).grid(row=7, column=0, columnspan=2, pady=5)
    tk.Button(root, text="Побудувати графік", command=plot_graph).grid(row=8,
column=0, columnspan=2, pady=5)

    root.mainloop()

if __name__ == "__main__":
    main()
```

Додаток Б

Скріншоти вікна виконання програми

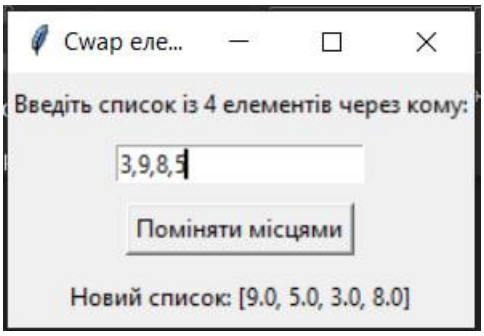

Рисунок Б.1 – Екран виконання програми до завдання Func 31

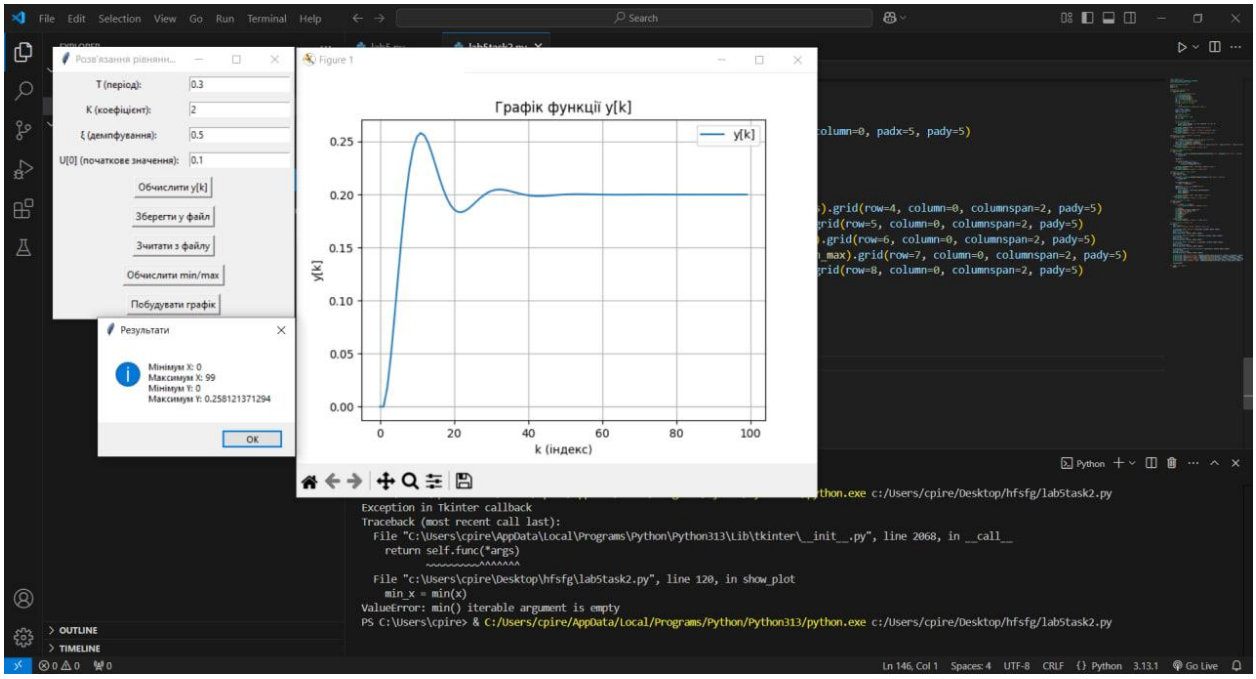

Рисунок Б.2 – Екран виконання програми до завдання 8