

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 .173. 310ст.5 ЛР

Виконав студент гр. _____310ст_____

_____Возвишаєв Олексій Андрійович
(підпис, дата) (П.І.Б.)

Перевірів

_____к.т.н., доц. О. В. Гавриленко
_____ас. В. О. Білозерський
(підпис, дата) (П.І.Б.)

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.

Func25. Описати функцію TrianglePS(a), що обчислює по стороні a рівностороннього трикутника його периметр $P = 3 \cdot a$ та площу $S = a^2 \cdot (3)^{1/2} / 4$ і повертає їх у вигляді двох дійсних чисел (a – дійсний параметр). За допомогою цієї функції знайти периметри та площі трьох рівносторонніх трикутників з цими сторонами.

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

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

- А. установка початкових значень параметрів для побудови графіка (змінні Tkinter)
 - В. 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 – '#';
 - С. зчитування з файлу масивів даних;
 - Д. підрахунок і відображення мінімального / максимального значення аргументу / функції у зчитаних масивах;
 - Е. відображення масивів даних за допомогою пакета matplotlib у вигляді графіка функції в декартовій системі координат з назвою функції, позначенням осей, оцифруванням і сіткою;
 - Ф. 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 Y), наприклад:

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: $t[k] = kT_0$, $T_0 = 2T/N$, $N = [20..1000]$ – кількість точок треба підібрати, щоб графік був гладким. Функція являє собою характеристику одного з об'єктів управління:

- *кут тангажа літака – ν , рад*
- *кутова швидкість обертання електродвигуна – ω , рад/с*
- *температура термостата – T , К*

ВИКОНАННЯ РОБОТИ

Завдання 1. Вирішення задачі Funs25

Вхідні дані

Ім'я змінної	Опис	Тип	Обмеження
a	Сторона рівностороннього трикутника	float	Повинна бути додатною (більше 0)
self.entry	Поле введення сторони трикутника	str	Має бути конвертоване у float (допускаються тільки дійсні числа)

Вихідні дані

Ім'я змінної	Опис	Тип
perimeter	Периметр рівностороннього трикутника	float
area	Площа рівностороннього трикутника	float
self.result_label	Текстовий вивід з периметром і площею	str
Повідомлення про помилку	Сповіщення про некоректне введення	Текстове повідомлення через messagebox

Алгоритм вирішення чи показано на рис. 1

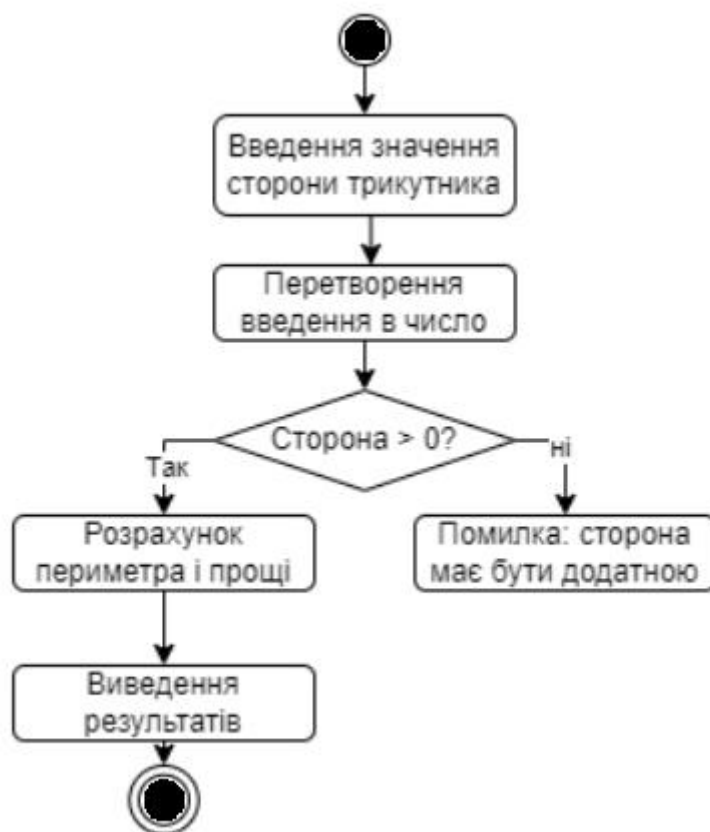


Рисунок 1 – Алгоритм вирішення завдання Funs25

Лістинг коду вирішення задачі наведено в дод. А стор. 8 Екран роботи програми показаний на рис. Б.11.

Завдання 2. Вирішення задачі 2

Вхідні дані

Ім'я змінної	Опис	Тип	Обмеження
entry_T	Період (T)	Число (float)	Будь-яке дійсне число
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

Вихідні дані

Ім'я змінної	Опис	Тип
data_x	Список значень X	Список (float)
data_y	Список значень Y	Список (float)
Повідомлення про помилки	При збереженні, зчитуванні, обчисленнях	Текст (str)
Графік	Графік функції (x, y) вікно matplotlib	Графічний об'єкт

Алгоритм вирішення показано на рис.2



Рисунок 1 – Алгоритм вирішення до завдання 2

Лістинг коду вирішення задачі наведено в дод. А стор. 8. Екран роботи програми показаний на рис. Б.11.

ВИСНОВКИ

У ході виконання лабораторної роботи було вивчено основи розробки графічного інтерфейсу користувача на базі бібліотеки Tkinter, а також застосування бібліотеки matplotlib для побудови графіків.

ДОДАТОК А

Лістинг коду програми до задачі Funs25

```
import tkinter as tk
from tkinter import messagebox

def TrianglePS(a):

    if a <= 0:
        raise ValueError("Сторона трикутника має бути додатною !!!!!!!")
    perimeter = 3 * a
    area = (a ** 2 * (3 ** 0.5)) / 4
    return perimeter, area

class TriangleApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Калькулятор 5лб")

        # Елементи інтерфейсу
        self.label = tk.Label(root, text="Введіть сторону трикутника")
        self.label.pack(pady=5)

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

        self.calculate_button = tk.Button(root, text="Обчислити",
command=self.calculate)
        self.calculate_button.pack(pady=5)

        self.result_label = tk.Label(root, text="Результати виходят тута :)")
        self.result_label.pack(pady=5)

    def calculate(self):
        try:
            a = float(self.entry.get())
            perimeter, area = TrianglePS(a)
            result_text = f"Периметр: {perimeter:.2f}\nПлоща: {area:.2f}"
            self.result_label.config(text=result_text)
        except ValueError as e:
            messagebox.showerror("Помилка", str(e))

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


Лістинг коду програми до задачі 2

```

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 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

        # Визначення роздільника залежно від варіанту
        delimiter = ';' if variant % 2 == 0 else '#'

        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 # Замінили nonlocal на global
    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)

        # Логування для перевірки зчитаних даних
        print("Зчитано дані:")
        print(f"data_x: {data_x}")
    
```

```

    print(f"data_y: {data_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="Функція y(t)")
        plt.title("Графік функції")
        plt.xlabel("Час (t)")
        plt.ylabel("Значення (y)")
        plt.grid(True)
        plt.legend()
        plt.show()
    except Exception as e:
        messagebox.showerror("Помилка", f"Помилка: {e}")

# Основна функція для побудови GUI
def main():
    global variant # Оголошуємо global для варіанту
    variant = 2 # Призначте значення варіанту

    root = tk.Tk()
    root.title("lab5_2-310СТ-v02-Vozvyshaev-Alex")

    tk.Label(root, text="T (період):").grid(row=0, column=0, padx=5, pady=5)
    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)
    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)
    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)
    entry_U0 = tk.Entry(root)
    entry_U0.grid(row=3, column=1, padx=5, pady=5)

    tk.Button(root, text="Зберегти у файл", command=save_to_file).grid(row=4,
column=0, columnspan=2, pady=5)
    tk.Button(root, text="Зчитати з файлу", command=load_from_file).grid(row=5,
column=0, columnspan=2, pady=5)
    tk.Button(root, text="Обчислити min/max",
command=calculate_min_max).grid(row=6, column=0, columnspan=2, pady=5)
    tk.Button(root, text="Побудувати графік", command=plot_graph).grid(row=7,
column=0, columnspan=2, pady=5)

    root.mainloop()

if __name__ == "__main__":
    main()

```

ДОДАТОК Б

Скрін-шоти вікна виконання програми Funs25

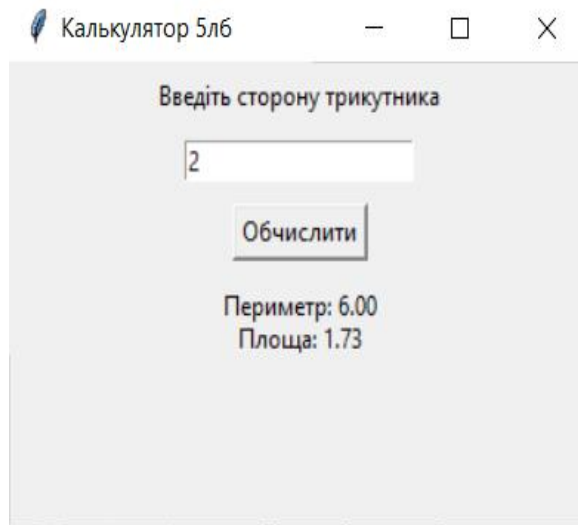


Рисунок Б.1 – Екран виконання програми для вирішення завдання Funs25

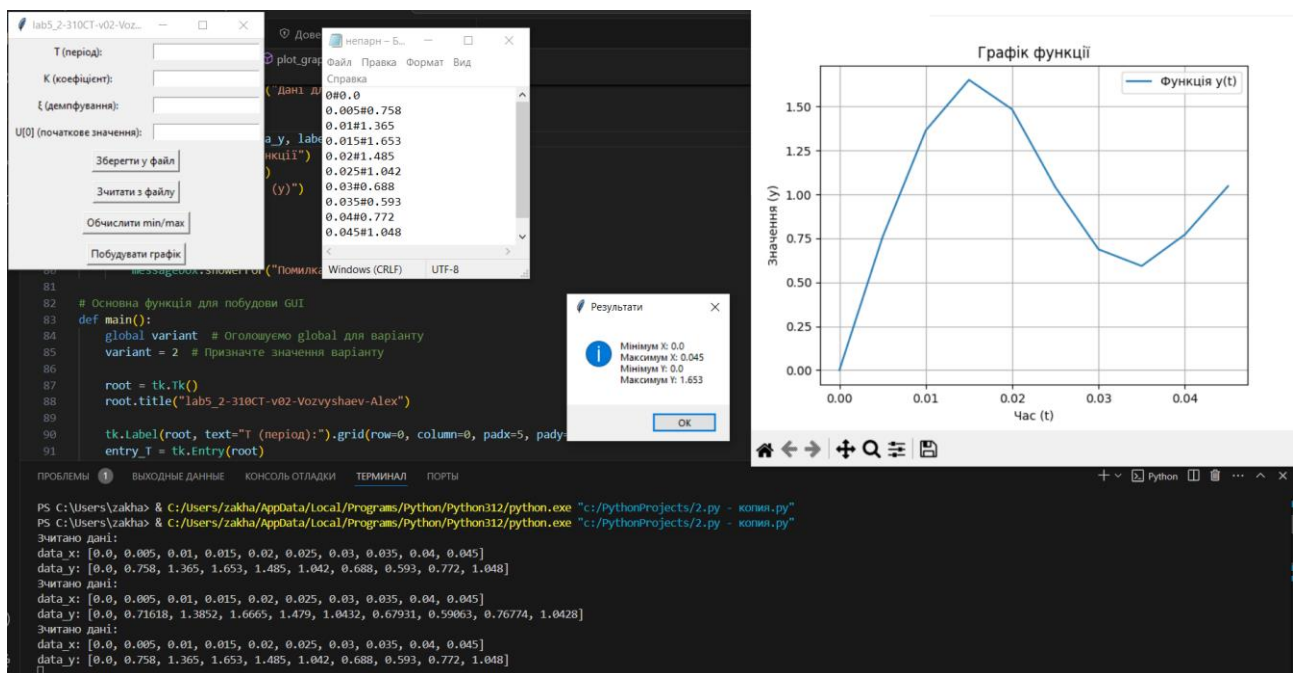


Рисунок Б.2 – Екран виконання програми для вирішення завдання Завдання 2

