

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
авіаційно-транспортних систем»

Тема: «"Розробка графічного інтерфейсу для розрахункових
завдань і побудови графіків"»

ХАІ.301.174.322.04 ЛР

Виконав студент гр. _____322_____

_____Зубов Євген Павлович_____
(підпис, дата) (П.І.Б.)

Перевірів

_____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'єктно-орієнтований 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 (див. табл.1) і скрипт для роботи з об'єктом цього класу. Зазначена у задачі функція повинна бути окремим методом класу.

Завдання 2. Розробити скрипт із графічним інтерфейсом, що виконує наступні функції: 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; D. 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; F. заголовок вікна повинен містити текст текст: lab # - <# групи> -v <# варіанту> - <прізвище> - <ім'я>, наприклад: lab4_2-320-v01-Ivanov-Ivan

ВИКОНАННЯ РОБОТИ

Завдання 1. Вирішення задачі Func23:

Вхідні дані (ім'я, опис, тип, обмеження):

A – число, float

Вихідні дані (ім'я, опис, тип):

Вивід результату у інтерфейсі

Алгоритм вирішення показано нижче чи показано на рис. 1.1

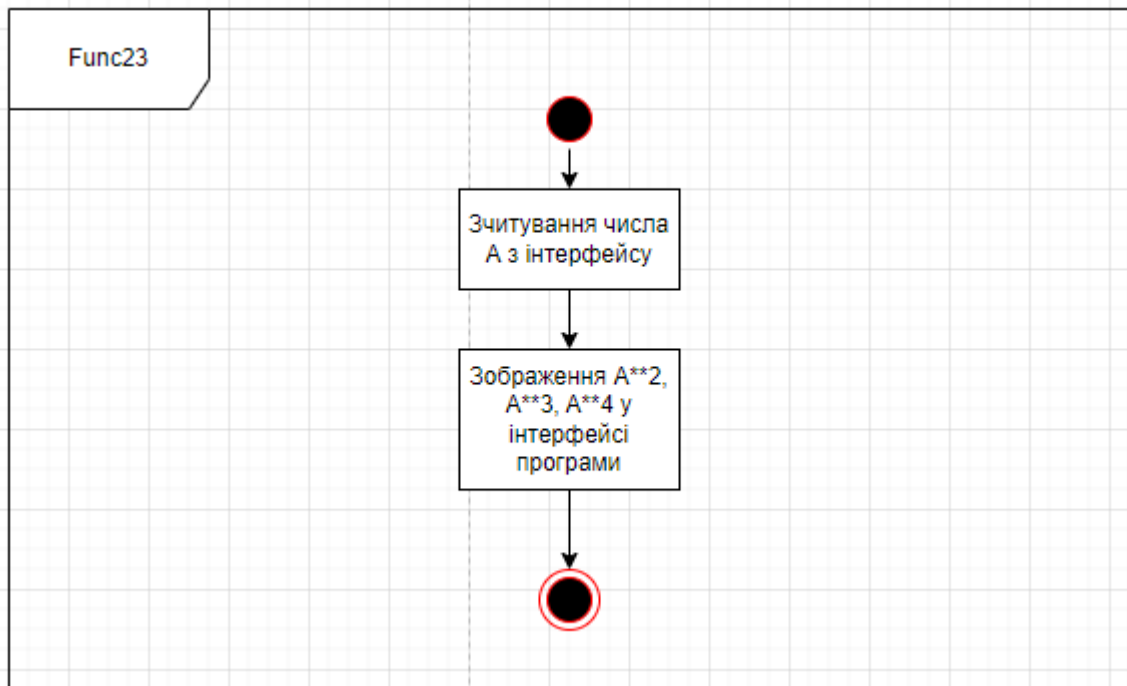


Рисунок 1.1 – Алгоритм вирішення задачі Func23

Лістинг коду вирішення задачі наведено в дод. А.2 (стор. 6).

Екран роботи програми показаний на рис. Б.1

Завдання 2. Вирішення задачі Solve14:

Вхідні дані (ім'я, опис, тип, обмеження):

T – час, float, > 0

N – кількість точок, int, > 20

K - const, int, > 0

ξ – коефіцієнт демпфування, float, > 0

U – кутова швидкість, float, > 0

Вихідні дані (ім'я, опис, тип):

Вивід результату у інтерфейсі

Лістинг коду вирішення задачі наведено в дод. А.3 (стор. 7).

Екран роботи програми показаний на рис. Б.2

ВИСНОВКИ

1. Tkinter:

Бібліотека для створення графічного інтерфейсу. Забезпечує прості способи додавання кнопок, полів вводу, меню тощо.

2. Matplotlib:

Бібліотека для побудови графіків і візуалізації даних. Використовується для створення двовимірних графіків та діаграм.

3. Об'єктно-орієнтоване програмування (ООП):

Дозволяє структурувати код у вигляді класів і об'єктів, що робить програму зрозумілою та розширюваною.

4. Розробка інженерних додатків:

Поєднання Tkinter і Matplotlib дозволяє створювати зручні інтерфейси для задач інженерного аналізу, включаючи обробку даних, графіки та автоматизацію розрахунків.

ДОДАТОК А.1

Лістинг головного меню завдання

```
from lab5_1 import Window1
from lab5_2 import Window2
import tkinter

task_list = {
    "1": (Window1, "lab5_1-322-v13-Zubov-Yevgen", "1024x768"),
    "2": (Window2, "lab5_2-322-v13-Zubov-Yevgen", "1024x768")
}

choice = input("Оберіть завдання 1-2 (0-EXIT): ")
while choice != "0":
    # якщо даний ключ є у словнику
    if choice in task_list.keys():
        # Створення відповідного вікна
        application = tkinter.Tk()
        window_class, window_name, window_size = task_list.get(choice)
        window = window_class(application)
        application.geometry(window_size)
        application.title(window_name)
        application.mainloop()
    else:
        print("Неправильно обрано завдання!")
        choice = input("Оберіть завдання знову! (0-EXIT): ")
```

ДОДАТОК А.2

Лістинг завдання №1

```
import tkinter

class Window1(tkinter.Frame):
    def __init__(self, parent):

        super().__init__(parent)
        self.parent = parent
        self.pack(fill=tkinter.BOTH, expand=1)
        self.config(bg="#A43820")
        self.create_widgets()
        self.grid_rowconfigure(0, weight=1)
        self.grid_rowconfigure(1, weight=1)
        self.grid_rowconfigure(2, weight=1)
        self.grid_rowconfigure(3, weight=1)
        self.grid_rowconfigure(4, weight=1)
        self.grid_columnconfigure(0, weight=1)
        self.grid_columnconfigure(1, weight=1)
        self.grid_columnconfigure(2, weight=1)

    def create_widgets(self):

        self.lb1 = tkinter.Label(self, text = "Введіть число A", bg="#A43820", fg
= "white")
        self.a_entr = tkinter.Entry(self)
        self.btn1 = tkinter.Button(self, text = "Возвести у степінь", command =
self.PowerA234)
        self.square_str = tkinter.StringVar()
        self.third_str = tkinter.StringVar()
        self.fourth_str = tkinter.StringVar()
        self.lb2 = tkinter.Label(self, textvariable = self.square_str,
bg="#A43820", fg = "white")
```

```

        self.lb3 = tkinter.Label(self, textvariable = self.third_str,
bg="#A43820", fg = "white")
        self.lb4 = tkinter.Label(self, textvariable = self.fourth_str,
bg="#A43820", fg = "white")

        self.lb1.grid(row=0, column=0, sticky=tkinter.NSEW)
        self.a_entr.grid(row=0, column=1, sticky=tkinter.NSEW)
        self.btn1.grid(row=1, column=0, sticky=tkinter.NSEW)
        self.lb2.grid(row=2, column=0, sticky=tkinter.NSEW)
        self.lb3.grid(row=3, column=0, sticky=tkinter.NSEW)
        self.lb4.grid(row=4, column=0, sticky=tkinter.NSEW)

def PowerA234(self):
    a = float(self.a_entr.get())

    self.square_str.set(a**2)
    self.third_str.set(a**3)
    self.fourth_str.set(a**4)

```

ДОДАТОК А.3

Лістинг завдання №2

```

import tkinter
from tkinter import messagebox

from pylab import *
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg

class Window2(tkinter.Frame):
    def __init__(self, parent):

        super().__init__(parent)
        self.parent = parent
        self.pack(fill=tkinter.BOTH, expand=1)
        self.config(bg="#A43820")
        self.create_widgets()
        self.grid_rowconfigure(0, weight=1)
        self.grid_rowconfigure(1, weight=1)
        self.grid_rowconfigure(2, weight=1)
        self.grid_rowconfigure(3, weight=1)
        self.grid_rowconfigure(4, weight=1)
        self.grid_rowconfigure(5, weight=1)
        self.grid_columnconfigure(0, weight=1)
        self.grid_columnconfigure(1, weight=1)
        self.grid_columnconfigure(2, weight=1)
        self.grid_columnconfigure(3, weight=1)
        self.grid_columnconfigure(4, weight=1)

    def create_widgets(self):

        self.lb1 = tkinter.Label(self, text = "Введіть число N", bg="#A43820", fg
= "white")
        self.lb2 = tkinter.Label(self, text="Введіть число K", bg="#A43820",
fg="white")
        self.lb3 = tkinter.Label(self, text="Введіть число ξ", bg="#A43820",
fg="white")
        self.lb4 = tkinter.Label(self, text="Введіть число U", bg="#A43820",
fg="white")
        self.lb5 = tkinter.Label(self, text="Введіть число T", bg="#A43820",
fg="white")

```

```

self.N_entr = tkinter.Entry(self)
self.K_entr = tkinter.Entry(self)
self.Xi_entr = tkinter.Entry(self)
self.U_entr = tkinter.Entry(self)
self.T_entr = tkinter.Entry(self)

self.create_btn = tkinter.Button(self, text="Создать файл",
command=self.create_file,background="#A43820", fg = "white")

self.lb1.grid(row=0, column=0)
self.lb2.grid(row=0, column=1)
self.lb3.grid(row=0, column=2)
self.lb4.grid(row=0, column=3)
self.lb5.grid(row=0, column=4)
self.N_entr.grid(row=1, column=0)
self.K_entr.grid(row=1, column=1)
self.Xi_entr.grid(row=1, column=2)
self.U_entr.grid(row=1, column=3)
self.T_entr.grid(row=1, column=4)
self.create_btn.grid(row=2, column=0)

def create_file(self):
    try:
        self.N = int(self.N_entr.get())
        if self.N < 20:
            raise ValueError
    except ValueError:
        messagebox.showerror("Ошибка", "N >= 20!")
    else:
        try:
            self.K = float(self.K_entr.get())
        except ValueError:
            messagebox.showerror("Ошибка", "K(!=0)")
        else:
            try:
                self.Xi = float(self.Xi_entr.get())
            except ValueError:
                messagebox.showerror("Ошибка", "ξ(!=0)")
            else:
                try:
                    self.U = float(self.U_entr.get())
                except ValueError:
                    messagebox.showerror("Ошибка", "U(!=0)")
                else:
                    try:
                        self.T = float(self.T_entr.get())
                    except ValueError:
                        messagebox.showerror("Ошибка", "T(!=0)")
                    else:
                        x, y = self.solve(self.N, self.K, self.Xi, self.U,
self.T)

                        self.data_save(x, y, "graph_data.txt")
                        self.plot_data(x, y)

def solve(self, N, K, Xi, U, T):
    T0 = 2 * T / N
    x = np.linspace(0, N * T0, N)
    y = np.zeros(N)

```



```

        for k in range(2, N-2):
            y[k+2] = (2 - ((2*Xi * T0) / T)) * y[k - 1] + (((2 * Xi * T0) / T) -
1 - (T0 ** 2 / T ** 2)) * y[k] + (
                (K * (T0 ** 2)) / T ** 2) * U
        return x, y

def plot_data(self, x, y):
    self.figure = plt.Figure(figsize=(5, 4), dpi=100)
    self.ax = self.figure.add_subplot(111)
    self.canvas = FigureCanvasTkAgg(self.figure, master=self.parent)
    self.canvas_widget = self.canvas.get_tk_widget()
    self.canvas_widget.pack(side=tkinter.RIGHT, fill=tkinter.BOTH,
expand=True)
    self.canvas_widget.place(x=500, y=300)

    self.ax.clear()
    self.ax.plot(x, y)
    self.ax.set_title("Завдання №14")
    self.ax.set_xlabel('Час')
    self.ax.set_ylabel('Значення функції')
    self.ax.grid(True)
    self.ax.legend()
    self.canvas.draw()

def data_save(self, x, y, filename):
    with open(filename, "w") as f:
        for x_vals, y_vals in zip(x, y):
            f.write(f"Крок: {x_vals} | Значення Y: {y_vals}\n")

```

ДОДАТОК Б

Скрін-шоти вікна виконання програми

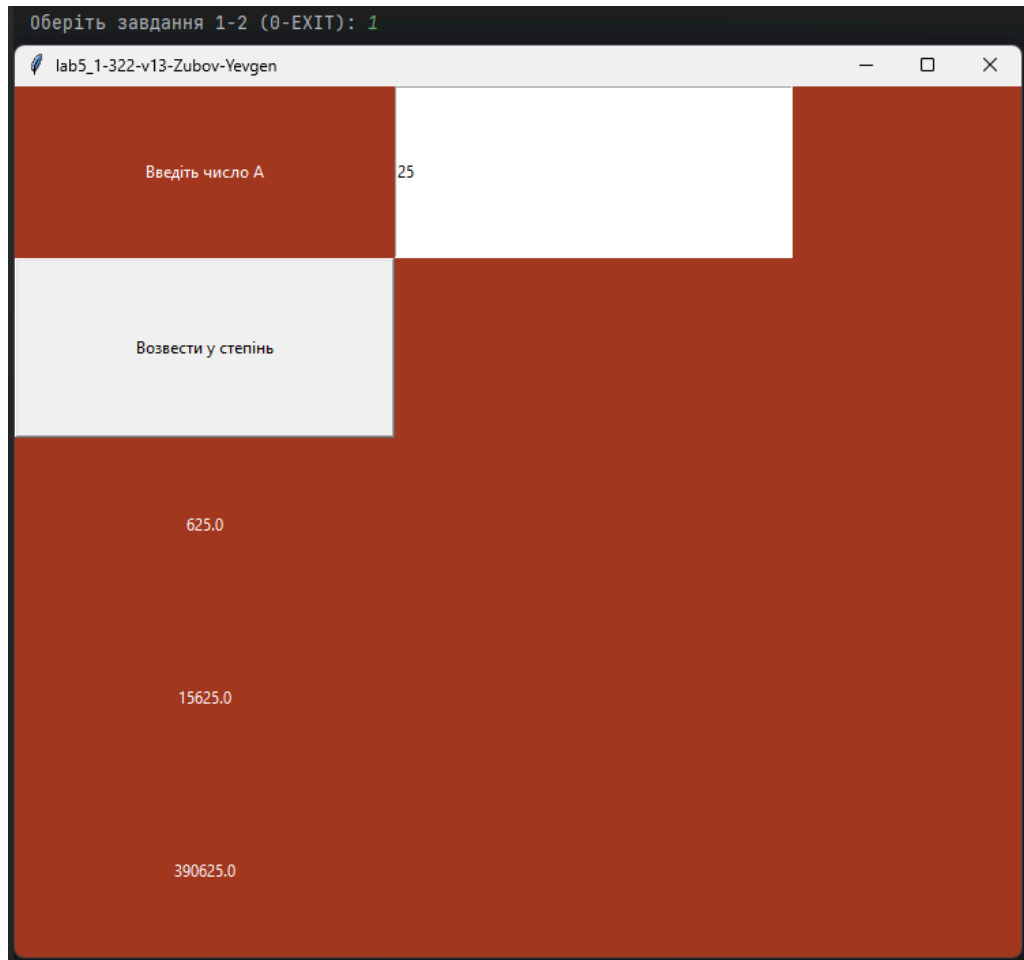


Рисунок Б.1 – Екран виконання програми для вирішення завдання
Func23

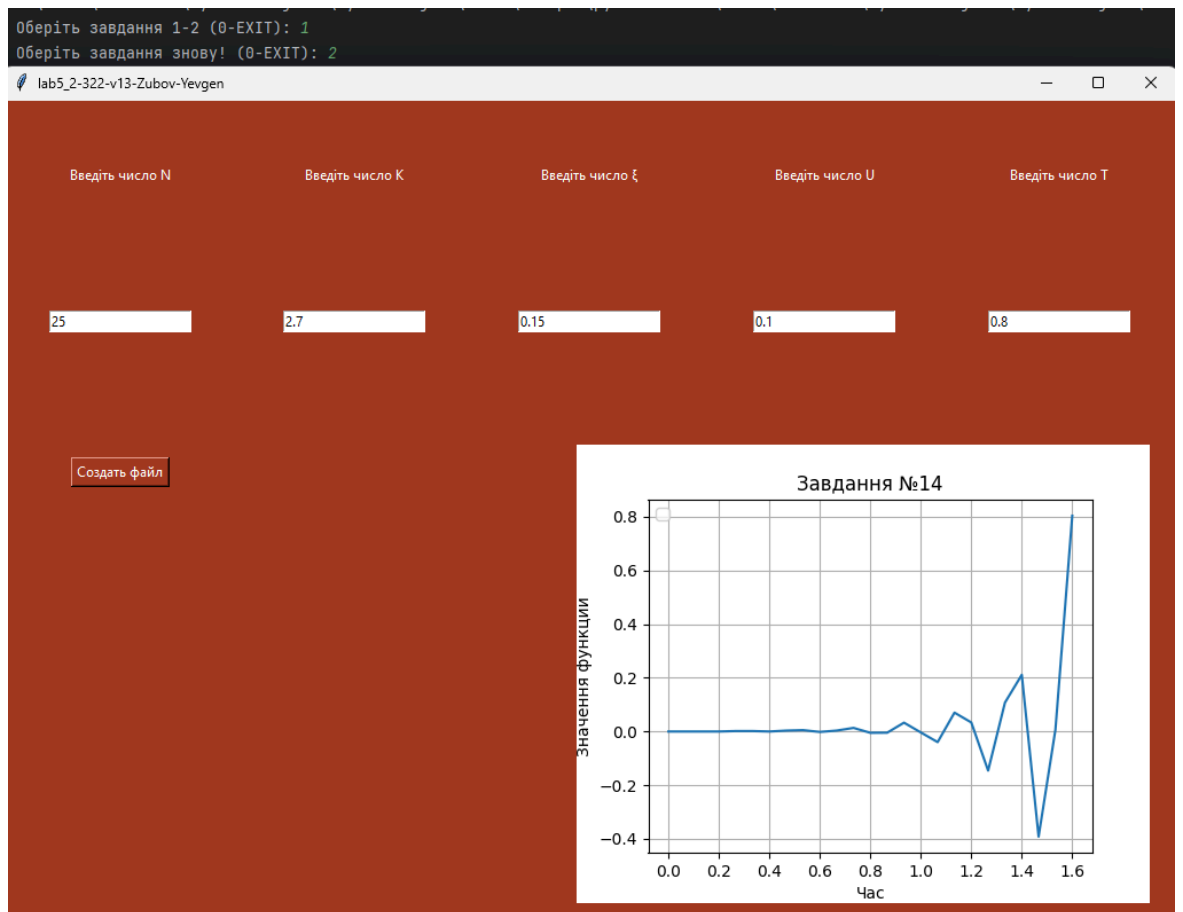


Рисунок Б.2 – Екран виконання програми для вирішення завдання
Solve14