

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»

Виконав студент гр. 320
Боднарчук Олексій
(підпис, дата) (П.І.Б.)
Перевірив
(підпис, дата)

2023

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.

Func23. Описати функцію PowerA234(A), яка обчислює другий, третій, четвертий ступінь числа A і повертає ці ступеня у вигляді трьох дійсних чисел (A – дійсний параметр). За допомогою цієї функції знайти другий, третій і четвертий ступінь п'яти даних чисел.

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;

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;

3	$y[k] = 2 \cdot \left(1 - \frac{\xi \cdot T_0}{T}\right) \cdot y[k-1] + \left(\frac{2 \cdot \xi \cdot T_0}{T} - 1 - \frac{T_0^2}{T^2}\right) \cdot y[k-2] + \frac{K \cdot T_0^2}{T^2} \cdot U$	$U[0] = 0.1 \text{ рад / с,}$ $y[0] = y[1] = 0$	$T = 0,05$ $K = 3$ $\xi = 0,25$	$y -$ $v, \text{ рад}$ $U -$ $\delta_v, \text{ рад}$
---	--	--	---------------------------------------	---

Лістинг коду

Підключення створених вікон

import tkinter

from task1 import CalculatorWithPowerA234

from task2 import Task2Window

словник для швидкого доступу до відповідної функції виконання

task_window_dict = {

"1": (CalculatorWithPowerA234, "Lab5_1-320-v23-Kostenko-Valeriya", "200x150"),

"2": (Task2Window, "Lab5_2-320-v3-Kostenko-Valeriya", "600x300")

}

Основна функція

def main():

choice = input("Please, choose the task 1-2 (0-EXIT): ")

while choice != "0":

```

# якщо даний ключ є у словнику
if choice in task_window_dict.keys():
    # Створення відповідного вікна
    application = tkinter.Tk()
    window_class, window_name, window_size =
task_window_dict.get(choice)
    window = window_class(application)
    application.geometry(window_size)
    application.title(window_name)
    application.mainloop()
else:
    print("Wrong task number!")
    choice = input("Please, choose the task again (0-EXIT): ")

if __name__ == '__main__':
    main()
    import tkinter
    from tkinter import messagebox

class CalculatorWithPowerA234(tkinter.Frame):
    """Graphical user interface and logic for calculating powers of a
    number"""

    def __init__(self, parent):
        super().__init__(parent)
        self.pack(fill=tkinter.BOTH, expand=1)
        self.grid_rowconfigure(0, weight=1)
        self.grid_rowconfigure(1, weight=1)
        self.grid_columnconfigure(0, weight=1)
        self.grid_columnconfigure(1, weight=1)
        self.grid_columnconfigure(2, weight=1)

    # Labels and Entry widgets for power calculation

```

```

self.lb5 = tkinter.Label(self, text="Enter number A:")
self.a_power_entr = tkinter.Entry(self)
self.btn2 = tkinter.Button(self, text="Calculate powers",
command=self.calc_powers)
self.result_label2 = tkinter.Label(self, text="Powers:")
self.lb6 = tkinter.Label(self, text="Second Power:")
self.lb7 = tkinter.Label(self, text="Third Power:")
self.lb8 = tkinter.Label(self, text="Fourth Power")
self.result2_str = tkinter.StringVar()
self.result_label_second_power = tkinter.Label(self,
textvariable=self.result2_str)

self.lb5.grid(row=4, column=0, sticky=tkinter.NSEW)
self.a_power_entr.grid(row=4, column=1, sticky=tkinter.NSEW)
self.btn2.grid(row=5, column=0, columnspan=2,
sticky=tkinter.NSEW)
self.result_label2.grid(row=6, column=0, columnspan=2,
sticky=tkinter.NSEW)
self.lb6.grid(row=7, column=0, sticky=tkinter.NSEW)
self.lb7.grid(row=8, column=0, sticky=tkinter.NSEW)
self.lb8.grid(row=9, column=0, sticky=tkinter.NSEW)
self.result_label_second_power.grid(row=7, column=1, rowspan=3,
sticky=tkinter.NSEW)

def calc_powers(self):
    try:
        a_power = float(self.a_power_entr.get())
    except ValueError:
        messagebox.showerror("Data ERROR", "Number must be a valid
real number!")
        self.a_power_entr.delete(0, tkinter.END)
    else:
        second_power, third_power, fourth_power =
PowerA234(a_power)

```

```
        result_text = f"Second Power: {second_power:.2f}\nThird Power: {third_power:.2f}\nFourth Power: {fourth_power:.2f}"
        self.result2_str.set(result_text)
```

```
def PowerA234(A):
    """Calculate the second, third, and fourth power of a given number."""
    square = A**2
    cube = A**3
    fourth_power = A**4
    return square, cube, fourth_power
```

```
# Для графічного інтерфейсу
import tkinter
from tkinter import messagebox
from tkinter.filedialog import askopenfile
# Для малювання графіка
from pylab import *
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
from matplotlib.figure import Figure
from PIL import Image, ImageTk
```

```
class Task2Window(tkinter.Frame):
    """Клас MainWindow, що наслідує Frame"""

    def __init__(self, parent):
        """Настройка графічного інтерфейсу"""
        super().__init__(parent)
        # Розтягнути фрейм
        self.pack(fill=tkinter.BOTH, expand=1)
        # Розтягнути сітку
        self.grid_rowconfigure(0, weight=1)
        self.grid_rowconfigure(1, weight=1)
```

```

self.grid_rowconfigure(2, 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)
# Створення віджетів (зображення виразу та поле для введення
N)
self.img = ImageTk.PhotoImage(file='image.png')
self.lb_image = tkinter.Label(self, image=self.img)
self.lb1 = tkinter.Label(self, text="N = ")
self.N_entr = tkinter.Entry(self)
# Створення віджетів (4 командні кнопки)
self.but1 = tkinter.Button(self, text="Create file",
command=self.create_file)
self.but2 = tkinter.Button(self, text="Open file",
command=self.open_file)
self.but3 = tkinter.Button(self, text="Show content",
command=self.show_msg)
self.but4 = tkinter.Button(self, text="Show plot",
command=self.show_plot)
# Розміщення віджетів в сітці основного вікна
self.lb_image.grid(row=0, column=0, columnspan=2,
sticky=tkinter.NSEW)
self.lb1.grid(row=0, column=2, sticky=tkinter.NSEW)
self.N_entr.grid(row=0, column=3, sticky=tkinter.NSEW)
self.but1.grid(row=1, column=0, sticky=tkinter.NSEW)
self.but2.grid(row=1, column=1, sticky=tkinter.NSEW)
self.but3.grid(row=1, column=2, sticky=tkinter.NSEW)
self.but4.grid(row=1, column=3, sticky=tkinter.NSEW)
self.text1 = "" # вміст файлу

def create_file(self):
    """Розрахунок значень функції і збереження результатів у
файл"""

```

```

try:
    N = int(self.N_entr.get())
    if N < 20:
        raise ValueError
except ValueError:
    messagebox.showerror("Data ERROR", "N must be integer that
    >= 20!")
else:
    # Параметри виразу
    K = 3
    T = 0.05
    T0 = 2*T/N
    U = 0.1
    eps=0.25
    x = [0]
    y = [0]
    # Розрахунок N значень x, y
    for k in range(1, N):
        x.append(k*T0)
        tmp_value = 2*(1-((eps*T0)/T))* y[k-1]+(((2*eps*T0)/T)-1-
        ((T0*T0)/(T*T)))*y[k-2]+((K*T0**2)/T**2)*U
        y.append(tmp_value)
    # збереження результатів у файл
    with open("graph_data.txt", 'w') as f:
        for i, x in enumerate(x):
            f.write("{}#{ }\n".format(x, y[i]))
    # повідомлення про успішний запис результатів у файл
    messagebox.showinfo("File creation", "File with data was
    created!")

def open_file(self):
    """Зчитування вмісту файлу і збереження в text1"""
    # Виклик вікна діалогу для відкриття файлу
    fopen = askopenfile(mode='r', defaultextension=". txt",

```



```

        filetypes=(("Text files", "*.txt"), ("All files", "*.*"))
if fopen is None: # якщо помилка відкриття файлу
    return
self.text1 = fopen.readlines() # файл -> список рядків
messagebox.showinfo("File opening", "File with data was opened!")

def show_msg(self):
    """Відобразити text1 у вікні messagebox"""
    messagebox.showinfo("File content", self.text1)

def show_plot(self):
    """Рисування графіку функції"""
    x = []
    y = []
    try: # розібрати список рядків text1
        for line in self.text1: # для кожного рядка
            words = line.split('#') # зберегти як список
            x.append(float(words[0])) # 1 ел.списка -> число -> x
            y.append(float(words[1])) # 2 ел.списка -> число -> y
    except ValueError:
        messagebox.showerror("Data ERROR", "Wrong file format!")
    else:
        # Область малювання графіка на полотні (Canvas)
        fig = Figure(figsize=(3, 3)) # створення об'єкта Figure
        a = fig.add_subplot(111) # створення об'єкта області
малювання (subplot)
        # Налаштування області побудови графіка
        a.plot(x, y, 'c--')
        # ...
        # Створення об'єкта Canvas і розміщення в основному вікні
        drawing = FigureCanvasTkAgg(fig, master=self)
        drawing.get_tk_widget().grid(row=2, column=0, columnspan=4,
sticky=tkinter.NSEW)
        drawing.draw()

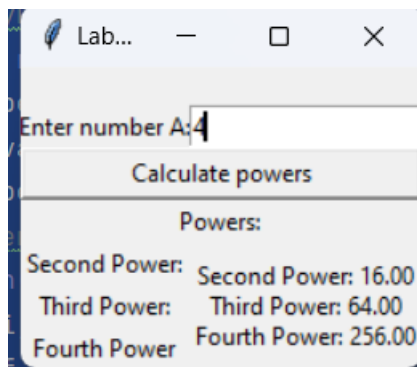
```

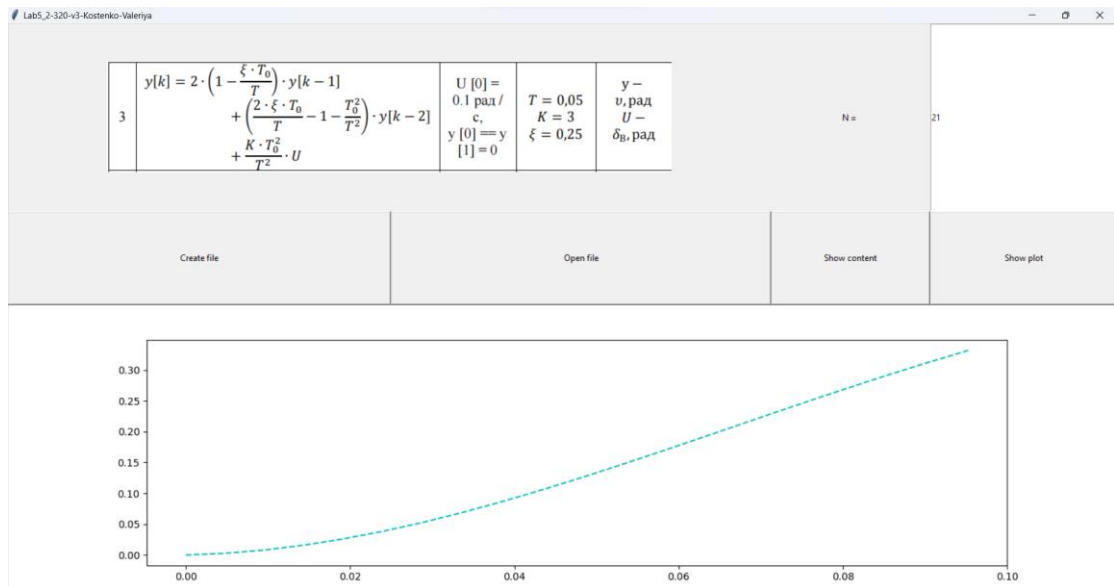
```

# Інформація про максимальне/мінімальне значення
аргументу/функції
min_x = min(x)
min_y = min(y)
max_x = max(x)
max_y = max(y)
messagebox.showinfo("Basic information", "X min = { }, X max
= { }\n"
                        "Y min = { }, Y max =
{ }".format(min_x, max_x, min_y, max_y))

```

Виконання





Висновок

У процесі виконання роботи я отримав теоретичні знання з основ програмування на мові Python звикористанням об'єктів і класів, навички використання бібліотеки для візуалізації масивів даних, і навчився розробляти скрипти для роботи з об'єктами призначених для користувача класів.