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.301.173.320.05 ЛР

Виконав студент гр. _____320_____

_____Перцев Кирило
(підпис, дата) (П.І.Б.)

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

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 і скрипт для роботи з об'єктом цього класу. Зазначена у задачі функція повинна бути окремим методом класу.

Func21. Описати функцію Fib(N) цілого типу, що обчислює N елемент послідовності чисел Фібоначчі FK, яка описується наступними формулами: F1 = 1, F2 = 1, FK = FK−2 + FK−1, K = 3, 4, … . Використовуючи функцію Fib, знайти п'ять чисел Фібоначчі із даними номерами N1, N2, …, N5.

Рисунок 1 – Завдання Func21

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

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. 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

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

Рисунок 2 – Завдання 2

ВИКОНАННЯ РОБОТИ

Завдання 1. Вирішення задачі Func21.

Вхідні дані: номер члену ряду, цілий тип.

Вихідні дані: число Фібоначчі.

Опис класу Task1Window

Об'єкт Task1Window містить такі атрибути:

lb1, lb3, lb4, a_entr, btn1, p_str, lb2: Віджети інтерфейсу.

Методи класу Task1Window:

__init__(parent): Метод ініціалізації, налаштовує інтерфейс користувача.

Fib(N): Метод, що реалізує обчислення послідовності Фібоначчі та відображення результату в інтерфейсі.

Лістинг коду вирішення задачі Func21 наведено в дод. А (стор. 5). Екрани роботи програми показані на рис. Б.1-Б.3.

Алгоритм вирішення показано в дод. В.

Завдання 2. Вирішення задачі task2.

Вхідні дані: кількість точок, цілий тип.

Вихідні дані: текстовий файл з переліком точок, графік функції, найбільше та найменше значення функції.

Лістинг коду вирішення задачі task2 наведено в дод. А (стор. 5). Екрани роботи програми показані на рис. Б.4-Б.8.

Алгоритм вирішення показано в дод. В.

ВИСНОВКИ

Під час виконання лабораторної роботи було вивчено теоретичний матеріал щодо створення програмних додатків.

У першому завданні було реалізовано клас Task1Window, який надає графічний інтерфейс для обчислення чисел Фібоначчі за введеним номером члену ряду.

Друге завдання включало розробку скрипту з графічним інтерфейсом для виконання різних функцій: встановлення початкових параметрів для побудови графіка, створення текстового файлу з даними, зчитування масивів даних з файлу, підрахунок мінімального / максимального значення аргументу / функції та побудова графіка функції у вигляді декартової системи координат.

ДОДАТОК А

Лістинг коду програми до задач 1, 2, 3, 4

Файл main.py

```
# Підключення створенних вікон
import tkinter
from task1 import Task1Window
from task2 import Task2Window

# словник для швидкого доступу до відповідної функції виконання
task_window_dict = {
    "1": (Task1Window, "Lab5_1-320-v05-Pertsev-Kyrylo", "450x200"),
    "2": (Task2Window, "Lab5_2-320-v05-Pertsev-Kyrylo", "1000x300")
}


# 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()
```

Файл task1.py

```
import tkinter
from tkinter import messagebox


class Task1Window(tkinter.Frame):  # MainWindow наслідуючий клас Frame
    """Graphical user interface and the logic of solving the task Begin1"""

    def __init__(self, parent):
        """Initial settings of the user interface"""
        super().__init__(parent)  # виклик конструктора базового класу
        # Розтягнути фрейм за розмірами вікна
        self.pack(fill=tkinter.BOTH, expand=1)
        # Розтягнути сітку 2x3 за розмірами фрейма
        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)
        # Створити об'єкти віджетів (змінні екземпляра)
        self.lb1 = tkinter.Label(self, text="Enter the sequence number of the
Fibonacci number:")  # статич. текст
        self.lb3 = tkinter.Label(self, text="")  # статичний текст
```

```
        self.lb4 = tkinter.Label(self, text="")  # статичний текст
        self.a_entr = tkinter.Entry(self)  # поле введення для a
        # Командна кнопка (запуск обчислень)
        self.btn1 = tkinter.Button(self, text="Get the number",
command=self.Fib)
        self.p_str = tkinter.StringVar()  # змінна tkinter: P в текстовому
вигляді
        self.lb2 = tkinter.Label(self, textvariable=self.p_str)  # Текстове
поле(P)
        # Розмістити віджети в сітці 2x3
        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=1, column=1, sticky=tkinter.NSEW)
        self.lb3.grid(row=0, column=2, sticky=tkinter.NSEW)
        self.lb4.grid(row=1, column=2, sticky=tkinter.NSEW)

    def Fib(N):
        """Input-calculation-output according to the task Begin1"""
        # Зчитування з поля введення
        try:
            a = int(N.a_entr.get())  # вважати і перетворити в дійсне
        except ValueError:
            # Вивести вікно з помилкою
            messagebox.showerror("Data ERROR", "The entered data must be a
number!")
            N.a_entr.delete(0, tkinter.END)  # очистити поле введення
        else:
            # Перевірити вхідні дані
            if a < 0:  # якщо негативне
                # Змінити на позитивне значення
                a = abs(a)
                N.a_entr.delete(0, tkinter.END)
                N.a_entr.insert(tkinter.END, str(a))
                # Вивести вікно з попередженням
                messagebox.showinfo("Data Warning", "Number has changed to
positive")
            # обчислення
            if a == 0: t3 = 0
            elif a == 1 or a == 2: t3 = 1
            else:
                t1 = 1
                t2 = 1
                for i in range(a - 2):
                    t3 = t2 + t1
                    t1 = t2
                    t2 = t3
            # Виведення результату в текстову мітку (за допомогою змінної
tkinter)
            N.p_str.set(str(t3))
```

Файл task2.py

```
# 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):
    """A MainWindow class that inherits from Frame"""

    def __init__(self, parent):
        """Graphical interface setting"""
        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):
        """Calculation of function values and saving the results to a file"""
        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 = 0.1
            ksi = 1.2
            T0 = 2*T/N
            U = 1
            x = [0]
            y = [0, 0]
            # Розрахунок N значень x, y
            for k in range(2, N):
                x.append(k*T0)
```

```
                tmp_value = (2 - 2 * ksi * T0 / T) * y[k - 1]+(2 * ksi * T0 / T
- 1 - T0 * T0 / T / T) * y[k - 2] + K * T0 * T0 / T / T
                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):
        """Reading the contents of the file and saving it in 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):
        """Display text1 in the messagebox window"""
        messagebox.showinfo("File content", self.text1)

    def show_plot(self):
        """Graphing a function"""
        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)
            # Настройка 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))
```

ДОДАТОК Б

Скрін-шоти вікна виконання програми

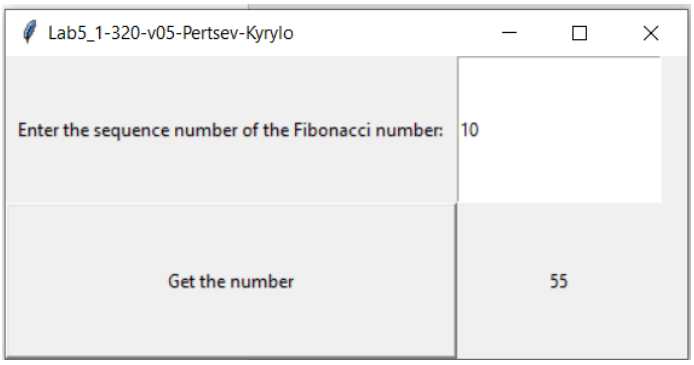

Рисунок Б.1 – Вікно до завдання 1

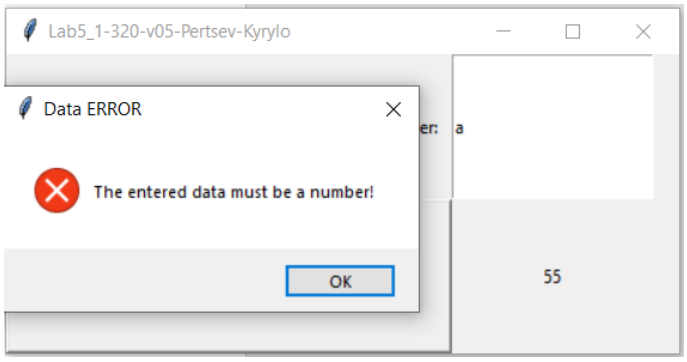

Рисунок Б.2 – Вікно помилки до завдання 1

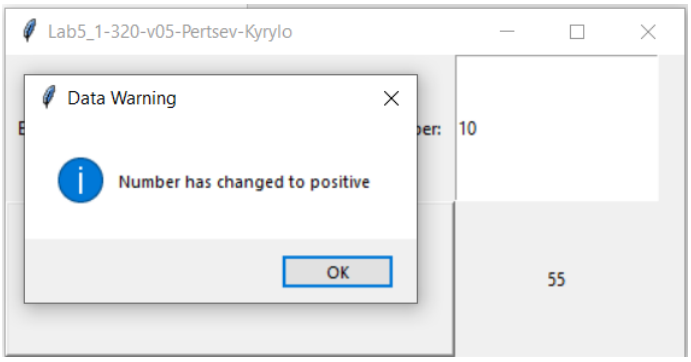

Рисунок Б.3 – Сповіщення при введенні від’ємного числа

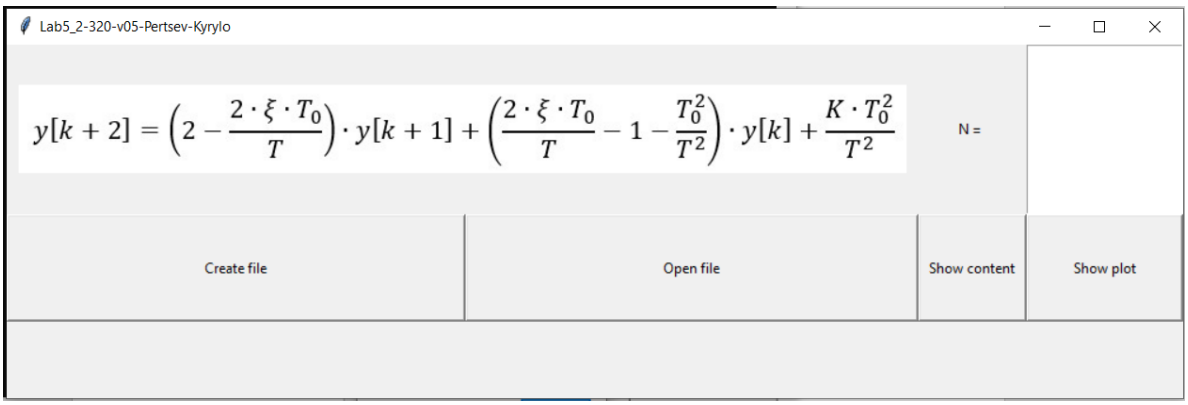

Рисунок Б.4 – Інтерфейс вікна до завдання 2

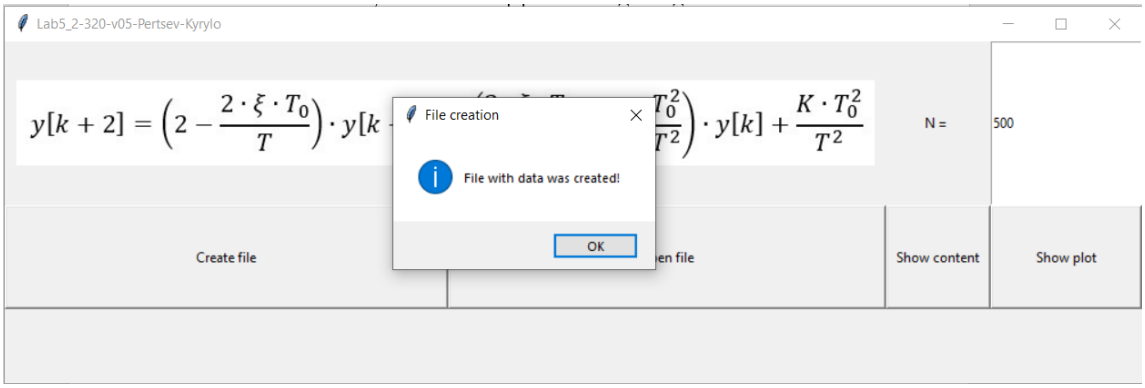

Рисунок Б.5 – Сповіщення про створення файлу

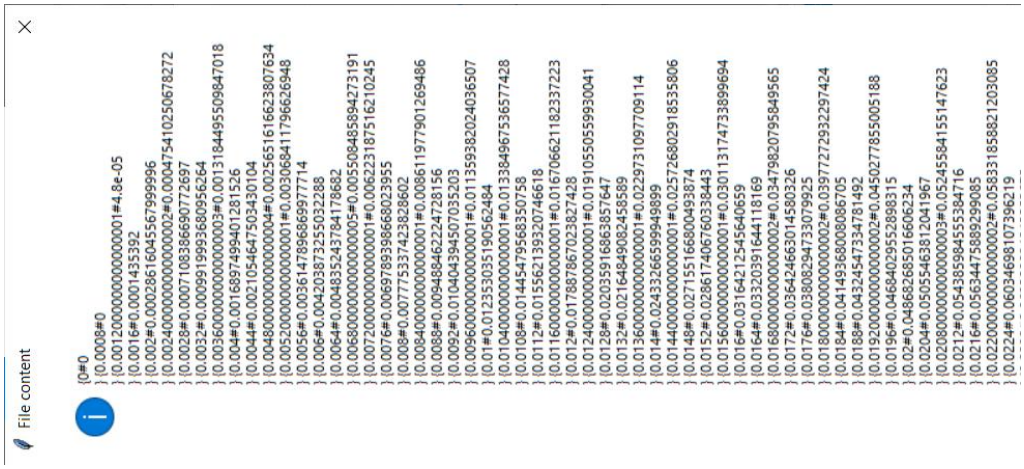

Рисунок Б.6 – Вміст файлу

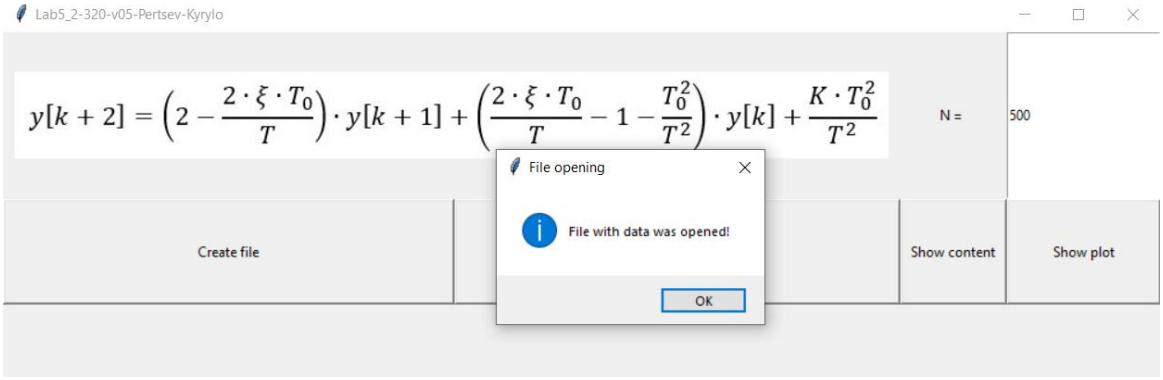

Рисунок Б.7 – Сповіщення про відкриття файлу

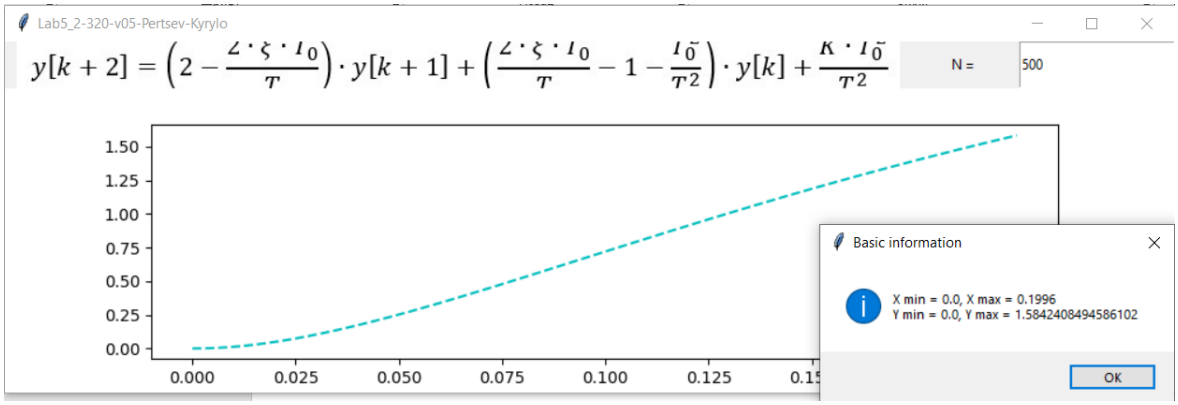

Рисунок Б.8 – Побудова графіку

ДОДАТОК В

Діаграми активності зо завдань 1 та всієї програми

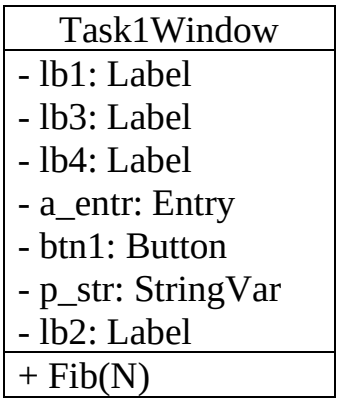

Рисунок В.1 – Діаграма класу Task1Window

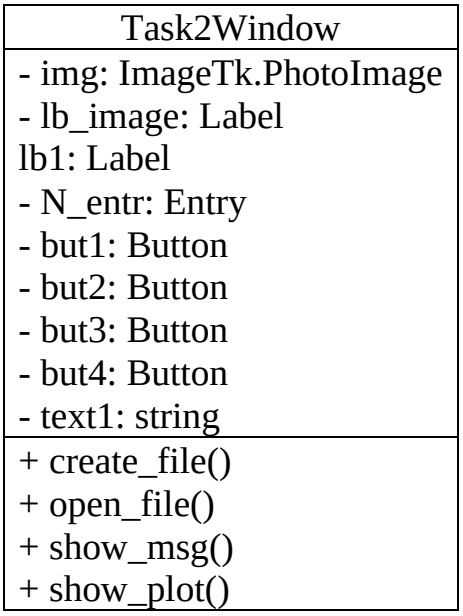

Рисунок В.2 – Діаграма класу Task2Window

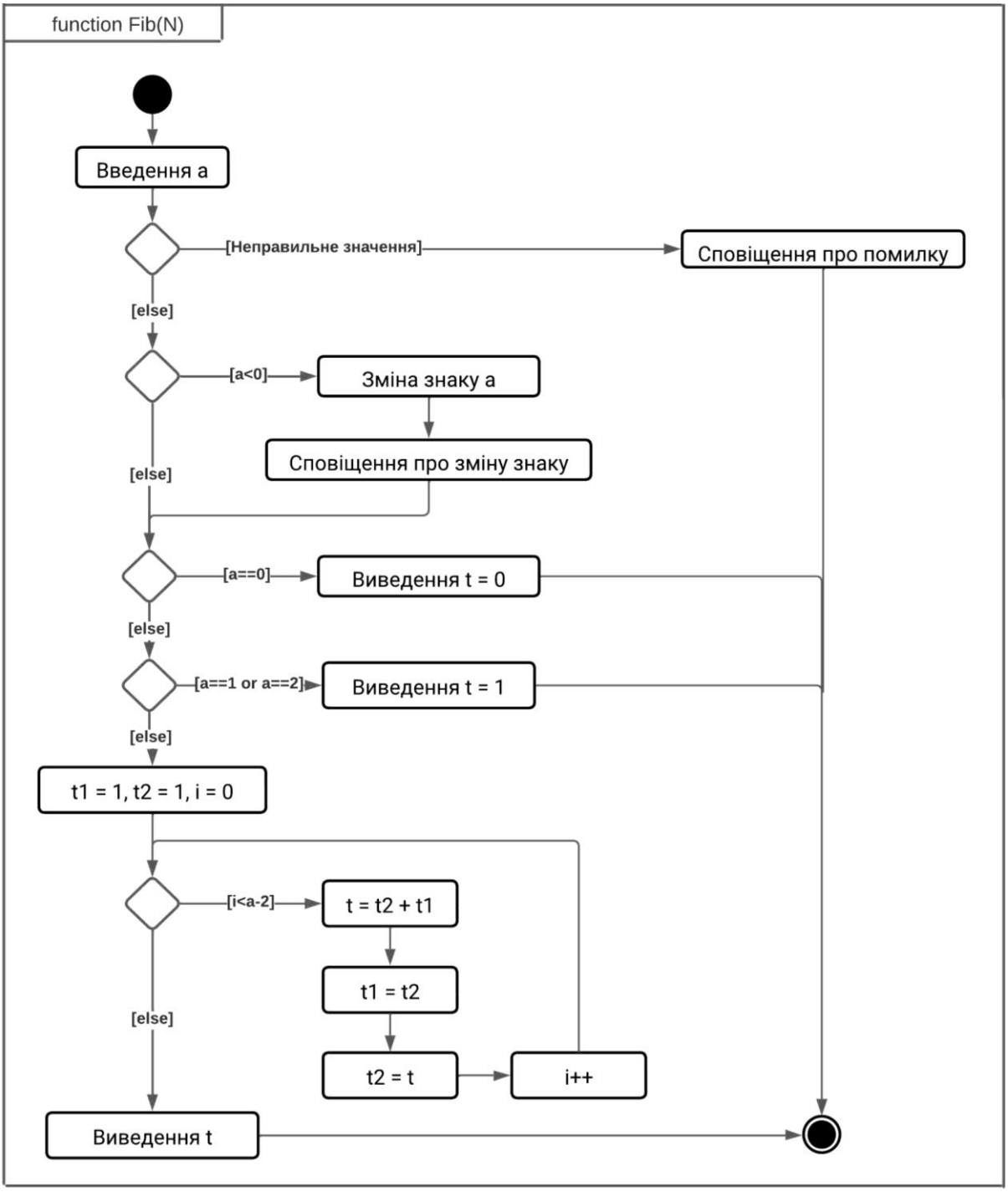

Рисунок Б.3 – Діаграма активності до функції Fib(N) завдання 1

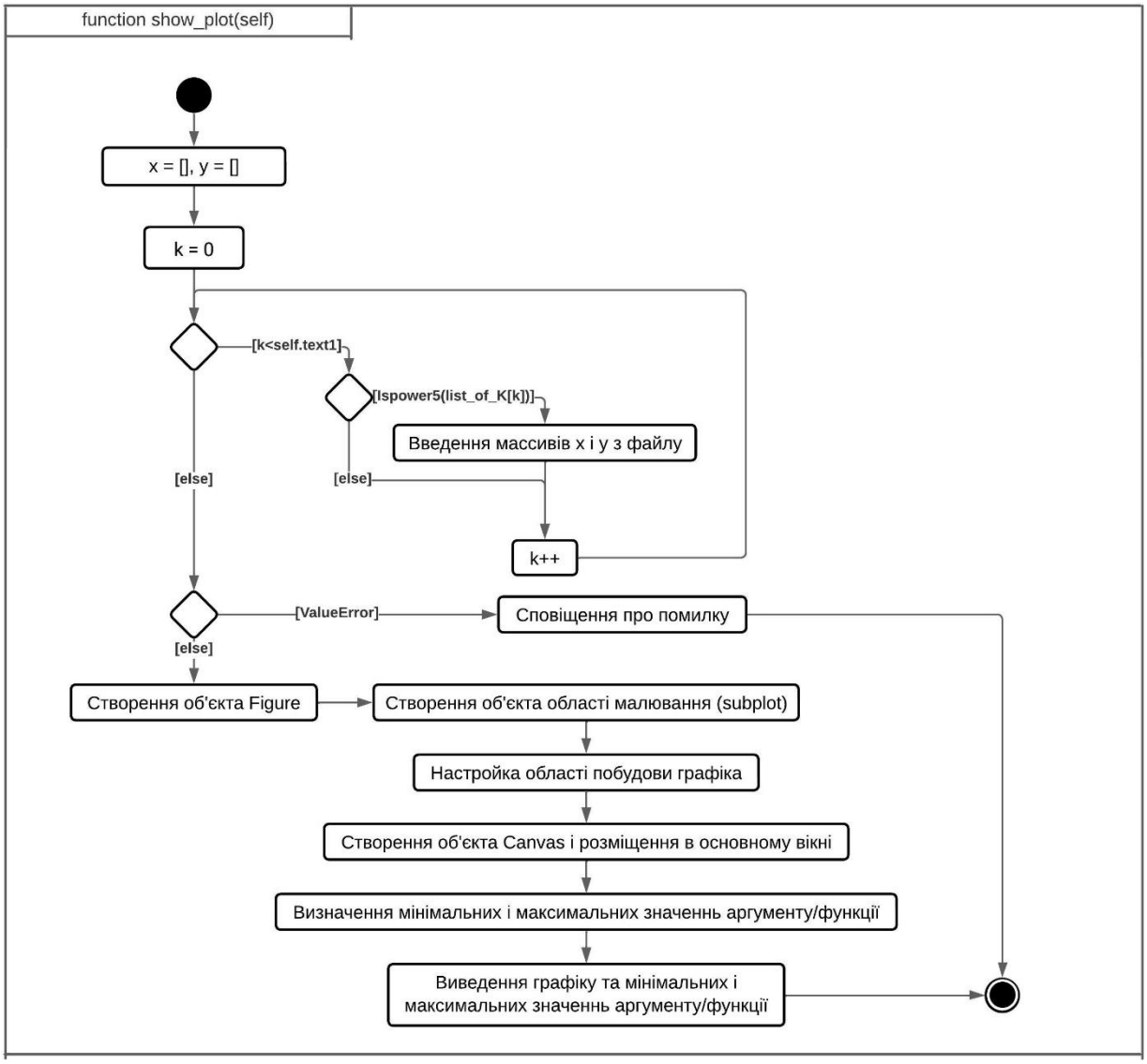

Рисунок Б.4 – Діаграма активності до функції show_plot(self) завдання 2