

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

Кафедра систем управління літальних апаратів

Лабораторна робота №5

з дисципліни «Об'єктно-орієнтоване проектування систем авіоніки»

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

XAI.301.173.320.5ЛР

Виконав студент гр. _____ 320 _____

_____ Черватюк В.О.

Перевірив

_____ к.т.н., доц. О.В.Гавриленко

_____ ас. В. О. Білозерський

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.

Func19. Описати функцію Fact(N) дійсного типу, яка обчислює значення факторіалу $N! = 1 \cdot 2 \cdot \dots \cdot N$ ($N > 0$ – параметр цілого типу; дійсне повертається значення використовується у тому, щоб уникнути цілочисельного переповнення при великих значеннях N). За допомогою цієї функції можна знайти факторіали п'яти даних цілих 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. Роздільник в кожному рядку файлу: для парних варіантів – ';', для непарних – '#';

7	$y[k] = \left(1 - \frac{T_0}{T}\right) \cdot y[k-1] + \frac{T_0}{T} \cdot K \cdot U$	$U[0] = 2 \text{ Вт},$ $y[0] = 0$	$T = 1,5$ $K = 0,8$	$y - T, K$ $U -$ $Q_{\text{н}}, \text{Вт}$
---	--	--------------------------------------	------------------------	--

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

ВИКОНАННЯ РОБОТИ

Завдання 1. Вирішення завдання варіанту 19

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

1) створення класу Tkinter:Frame

Вихідні дані (ім'я, опис, тип):

1) обчислення факторіалу 5ти цілих чисел

Алгоритм вирішення показано на рис.1-2

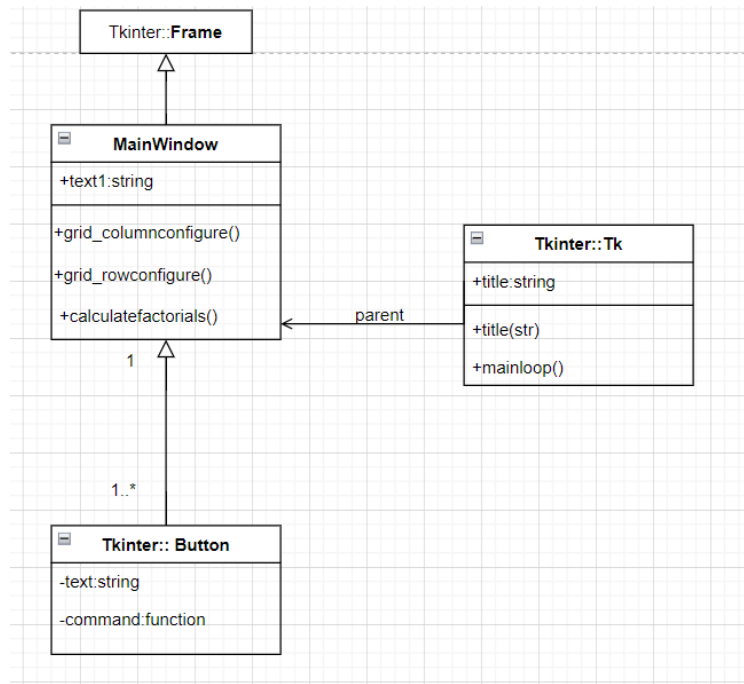


Рисунок 1 – Діаграма класів функції завдання варіанту 19

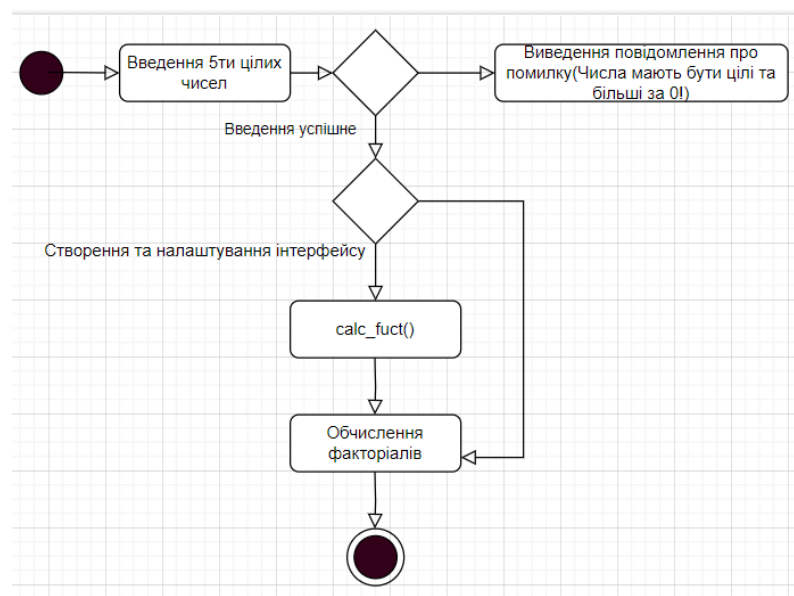


Рисунок 2 – Діаграма активності функції завдання варіанту 19

Лістинг коду вирішення задачі наведено в дод. А (стор. 6-10). Екран роботи програми наведено в дод. Б (стор. 11)

Завдання 2. Вирішення завдання варіанту 7

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

1) створення класу Tkinter.Frame

Вихідні дані (ім'я, опис, тип):

1) графік

Алгоритм вирішення показано на рис.1-2

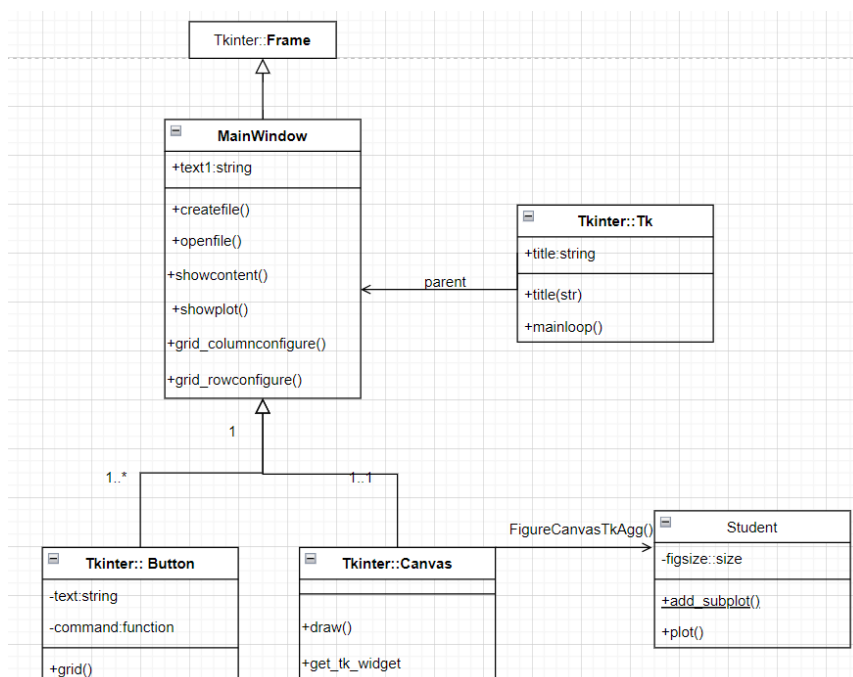


Рисунок 1 – Діаграма класів функції завдання варіанту 7

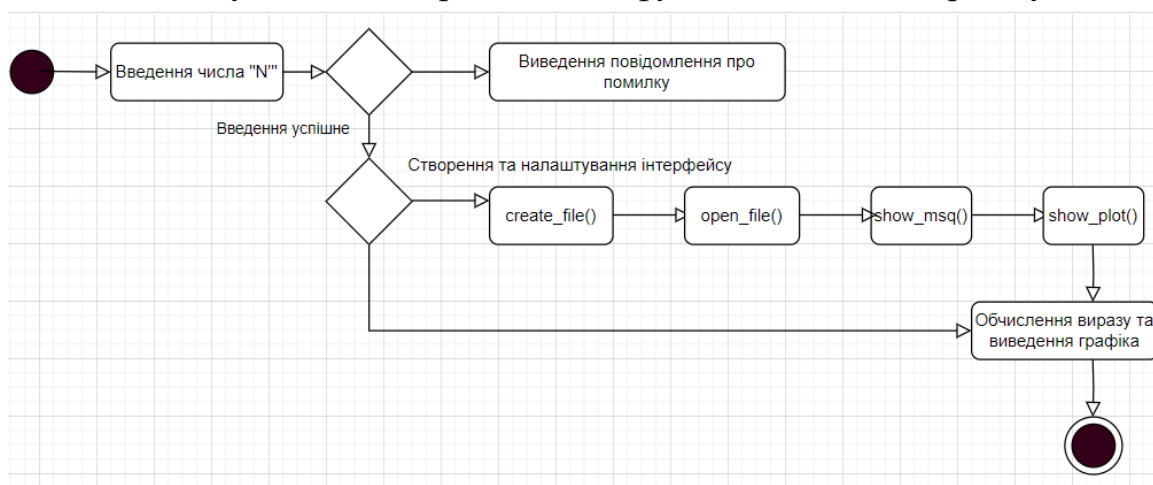


Рисунок 2 – Діаграма активності функції завдання варіанту 7

Лістинг коду вирішення задачі наведено в дод. А (стор. 7-10). Екран роботи програми наведено в дод. Б (стор. 11)

ВИСНОВКИ

Були закріплені теоретичні знання з основ роботи з бібліотекою `tkinter` на мові Python, навички використання бібліотеки `matplotlib`, а також об'єктно-орієнтований підхід до проектування програм. Навчилися розробляти скрипти для інженерних додатків з графічним інтерфейсом.

ДОДАТОК А

Лістинг коду програми до завдання варіанту 19

task1.py

```
import tkinter
from tkinter import messagebox

class Task1Window(tkinter.Frame):
    """Графічний інтерфейс користувача і логіка рішення задачі Func19"""

    def __init__(self, parent):
        """Початкові налаштування інтерфейсу користувача"""
        super().__init__(parent)
        # Розтягнути фрейм за розмірами вікна
        self.pack(fill=tkinter.BOTH, expand=1)
        # Розтягнути сітку 2x2 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)
        # Створити об'єкти віджетів (змінні екземпляра)
        self.lb1 = tkinter.Label(self, text="Enter 5 integer numbers:") # статич.
текст
        self.n_entr = tkinter.Entry(self) # поле введення для n!
        # Командна кнопка (запуск обчислень)
        self.btn1 = tkinter.Button(self, text="Calculate factorials",
command=self.calc_fact)
        self.n_str = tkinter.StringVar() # змінна tkinter: факторіали в текстовому
вигляді
        self.lb2 = tkinter.Label(self, textvariable=self.n_str) # Текстове поле(n!)
        # Розмістити віджети в сітці 2x2
        self.lb1.grid(row=0, column=0, sticky=tkinter.NSEW)
        self.n_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)

    def calc_fact(self):
        """Введення-підрахунок-виведення згідно завдання Fuct19"""
        # Зчитування з поля введення
        try:
            input_numbers = self.n_entr.get().split() # вважати і розбити по
пробілах
            numbers = [int(num) for num in input_numbers] # перетворити в цілі числа
        except ValueError:
            # Вивести вікно з помилкою
            messagebox.showerror("Data ERROR", "Numbers must be integers!")
            self.n_entr.delete(0, tkinter.END) # очистити поле введення
        else:
            # Перевірити вхідні дані
            if len(numbers) != 5:
                # Очистити поле
```

```

        self.n_entr.delete(0, tkinter.END)
        self.n_entr.insert(tkinter.END, ' '.join(str(num) for num in
numbers))

        # Вивести вікно з попередженням
        messagebox.showinfo("Data Warning", "Enter exactly 5 integer
numbers")

        elif any(num <= 0 for num in numbers):
            messagebox.showerror("Data ERROR", "Numbers must be greater than 0!")
            self.n_entr.delete(0, tkinter.END) # очистити поле введення
        else:
            # обчислення факторіалів
            factorials = [self.Fact(num) for num in numbers]
            # Виведення результату в текстову мітку (за допомогою змінної
tkinter)

            self.n_str.set(' '.join(str(fact) for fact in factorials))

def Fact(self, N):
    """Функція для обчислення факторіалу"""
    if N == 0:
        return 1
    else:
        return N * self.Fact(N - 1)

```

Лістинг коду програми до завдання варіанту 7

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):
    """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 = 0.8
        T = 1.5
        T0 = 2*T/N
        U = 2
        x = [0]
        y = [0]
        # Розрахунок N значень x, y
        for k in range(1, N):
            x.append(k*T0)
            tmp_value = (1 - T0/T) * y[k-1] + T0/T * K * 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"
            {}").format(min_x, max_x, min_y, max_y))

```

main.py

```

# Підключення створених 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-v19-Chervatiuk-Valeria", "300x200"),
    "2": (Task2Window, "Lab5_2-320-v07-Chervatiuk-Valeria", "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()
```

ДОДАТОК Б

Скрін-шоти вікна виконання програми

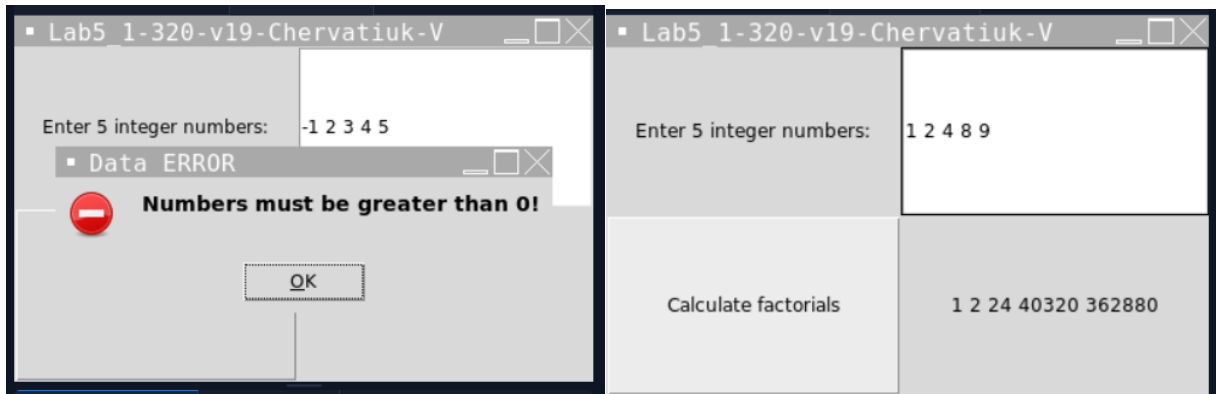


Рисунок Б.1 – Екран виконання програми для вирішення завдання варіанту 19

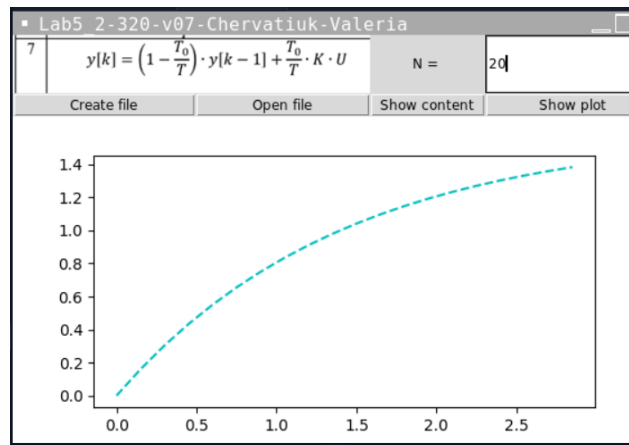


Рисунок Б.2 – Екран виконання програми для вирішення завдання варіанту 7