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»

Виконала студентка гр. 312ст

Ковальов Олег

(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.

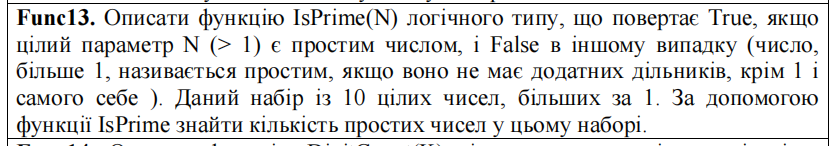
ПОСТАНОВКА ЗАДАЧІ

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.



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 (*див. табл.2)*. Роздільник в

кожному рядку файлу: для парних варіантів – ';', для непарних – '#';

C. зчитування з файлу масивів даних;

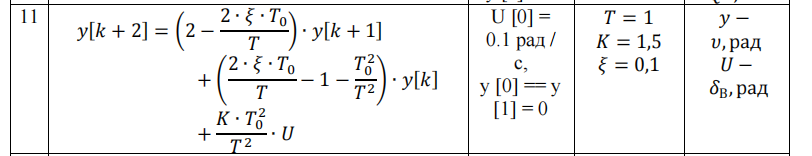
D. підрахунок і відображення мінімального / максимального значення

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;

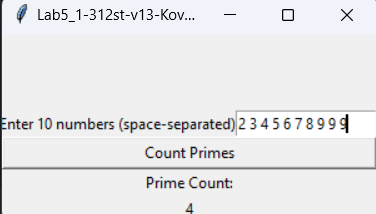


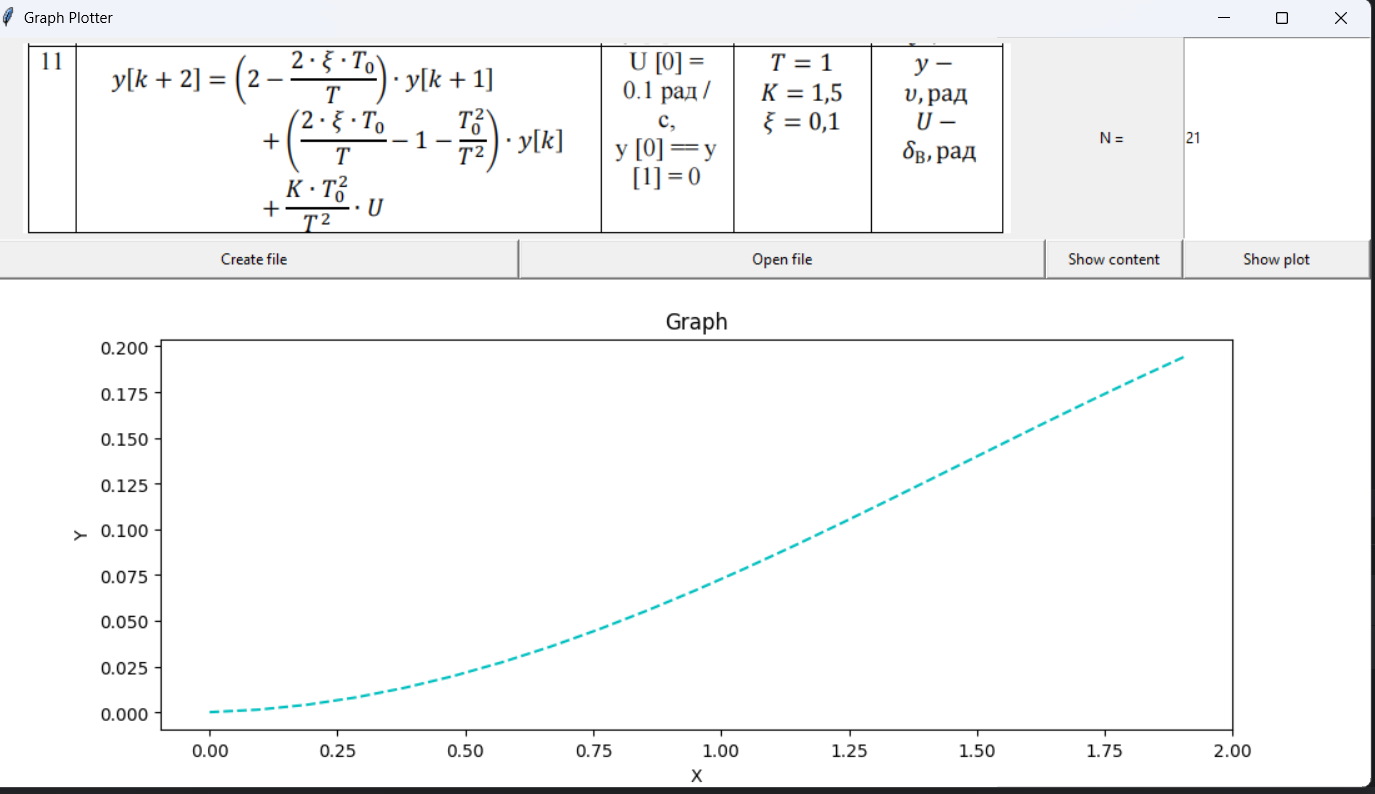
Лістинг коду  
# Підключення створенних вікон  
import tkinter  
from task1 import CalculatorWithPrimeCounter  
from task2 import Task2Window  
  
# словник для швидкого доступу до відповідної функції виконання  
task\_window\_dict = {  
 "1": (CalculatorWithPrimeCounter, "Lab5\_1-312st-v13-Kovalyov-Oleg", "300x150"),  
 "2": (Task2Window, "Lab5\_2-312st-v11-Kovalyov-Oleg", "900x300")  
}  
  
  
# 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 CalculatorWithPrimeCounter(tkinter.Frame):  
 """Graphical user interface and logic for counting prime numbers in a set"""  
  
 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 prime counting  
 self.lb5 = tkinter.Label(self, text="Enter 10 numbers (space-separated):")  
 self.numbers\_entry = tkinter.Entry(self)  
 self.btn\_count\_primes = tkinter.Button(self, text="Count Primes", command=self.count\_primes)  
 self.result\_label = tkinter.Label(self, text="Prime Count:")  
 self.result\_str = tkinter.StringVar()  
 self.result\_label\_prime\_count = tkinter.Label(self, textvariable=self.result\_str)  
  
 self.lb5.grid(row=4, column=0, sticky=tkinter.NSEW)  
 self.numbers\_entry.grid(row=4, column=1, sticky=tkinter.NSEW)  
 self.btn\_count\_primes.grid(row=5, column=0, columnspan=2, sticky=tkinter.NSEW)  
 self.result\_label.grid(row=6, column=0, columnspan=2, sticky=tkinter.NSEW)  
 self.result\_label\_prime\_count.grid(row=7, column=0, columnspan=2, sticky=tkinter.NSEW)  
  
 def count\_primes(self):  
 try:  
 numbers = [int(x.strip()) for x in self.numbers\_entry.get().split()]  
 if len(numbers) == 10:  
 prime\_count = sum(self.is\_prime(num) for num in numbers)  
 self.result\_str.set(prime\_count)  
 else:  
 messagebox.showerror("Input Error", "Please enter exactly 10 integers separated by spaces")  
 except ValueError:  
 messagebox.showerror("Input Error", "Please enter 10 valid integers separated by spaces")  
  
 def is\_prime(self, n):  
 """Check if a number is prime."""  
 if n < 2:  
 return False  
 for i in range(2, int(n\*\*0.5) + 1):  
 if n % i == 0:  
 return False  
 return True

import tkinter  
from tkinter import messagebox  
from tkinter.filedialog import askopenfile  
from matplotlib.backends.backend\_tkagg import FigureCanvasTkAgg  
from matplotlib.figure import Figure  
from PIL import Image, ImageTk  
  
  
class Task2Window(tkinter.Frame):  
 """Класс MainWindow, наследующий Frame"""  
  
 def \_\_init\_\_(self, parent):  
 """Настройка графического интерфейса"""  
 super().\_\_init\_\_(parent)  
 self.pack(fill=tkinter.BOTH, expand=1)  
 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)  
  
 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)  
  
 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)  
  
 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 = "" # содержимое файла  
  
 def create\_file(self):  
 """Расчет значений функции и сохранение результатов в файл"""  
 try:  
 N = int(self.N\_entr.get())  
 if N < 20:  
 raise ValueError  
 except ValueError:  
 messagebox.showerror("Data ERROR", "N must be an integer that is >= 20!")  
 else:  
 K = 1.5  
 T = 1  
 T0 = 2 \* T / N  
 U = 0.1  
 eps = 0.1  
 x = [0]  
 y = [0]  
  
 for k in range(1, N):  
 x.append(k \* T0)  
 tmp\_value = (2 - ((2 \* 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)  
  
 with open("graph\_data.txt", 'w') as f:  
 for i, x\_val in enumerate(x):  
 f.write("{}#{}\n".format(x\_val, y[i]))  
  
 messagebox.showinfo("File creation", "File with data was created!")  
  
 def open\_file(self):  
 """Чтение содержимого файла и сохранение в text1"""  
 fopen = askopenfile(mode='r', defaultextension=".txt",  
 filetypes=(("Text files", "\*.txt"), ("All files", "\*.\*")))  
 if fopen is None:  
 return  
 self.text1 = fopen.readlines()  
 messagebox.showinfo("File opening", "File with data was opened!")  
  
 def show\_msg(self):  
 """Отобразить text1 в окне messagebox"""  
 messagebox.showinfo("File content", self.text1)  
  
 def show\_plot(self):  
 """Построение графика функции"""  
 x = []  
 y = []  
 try:  
 for line in self.text1:  
 words = line.split('#')  
 x.append(float(words[0]))  
 y.append(float(words[1]))  
 except ValueError:  
 messagebox.showerror("Data ERROR", "Wrong file format!")  
 else:  
 fig = Figure(figsize=(5, 4))  
 a = fig.add\_subplot(111)  
 a.plot(x, y, 'c--')  
 a.set\_xlabel('X')  
 a.set\_ylabel('Y')  
 a.set\_title('Graph')  
 drawing = FigureCanvasTkAgg(fig, master=self)  
 drawing.get\_tk\_widget().grid(row=2, column=0, columnspan=4, sticky=tkinter.NSEW)  
 drawing.draw()  
  
 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 процессі виконання роботи я отримаd теоретичні знання з основ програмування на мові Python звикористанням об'єктів і класів, навички використання бібліотеки для візуалізації масивів даних, і навчитися розробляти скрипти для роботи з об'єктами призначених для користувача класів.