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 «Реалізація класу і робота з об'єктами»

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

ХАІ.301 . Інженерія мобільних додатків. 312ст.1 ЛР

Виконав студент гр. \_\_\_\_\_\_312ст\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Васильєв Б.А.\_\_\_

(підпис, дата) (П.І.Б.)

Перевірив

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 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

Func24. Описати функцію Mean(X, Y), що обчислює середнє арифметичне(X+Y)/2 та середнє геометричне (X·Y)1/2 двох додатних чисел X і Y і повертаєрезультат у вигляді двох дійсних чисел (X і Y — дійсні параметри). Задопомогою цієї функції знайти середнє арифметичне та середнє геометричне для пар(A, B),(A, C),(A, D),якщо дані A,B,C,D.

Завдання 2. побудова графіка функції

Зображення, що містить текст, Шрифт, знімок екрана, ряд

Автоматично згенерований опис

ВИКОНАННЯ РОБОТИ

Завдання 1. Func24

Вхідні дані:

n (int): Кількість точок

T0 (float): Крок часу

K (float): Константа

U (float): Початкове значення

Вихідні дані:

t (list[float]): Часові відмітки.

y (list[float]): Значення функції y(t).

file\_output (str): Текстовий файл із координатами t,y.

graph\_plot: Графік функції y(t).

Завдання 2. побудова графіка функції

Вхідні дані:

Числа A, B, C, D (float), введені користувачем.

Вихідні дані:

Середнє арифметичне та геометричне для пар (A, B), (A, C), (A, D).

Зображення, що містить схема, текст, знімок екрана, Креслення

Автоматично згенерований опис

Рисунок 1 діаграма Завдання 1: Вирішення задачі Func24.

Лістинг коду вирішення задачі наведено в дод. А (стор. 5). Екран роботи програми показаний на рис. Б.3.

Рисунок 2 діаграма Завдання 2: Вирішення задачі побудова графіка функції.

Зображення, що містить текст, знімок екрана, схема, Шрифт

Автоматично згенерований опис

Лістинг коду вирішення задачі наведено в дод. Б (стор. 7). Екран роботи програми показаний на рис. Б.4.

ВИСНОВКИ

Розроблено клас з графічним інтерфейсом для обчислення середніх величин. Функція обчислення реалізована як метод класу. Створено скрипт з GUI для збереження даних у файл та побудови графіка за рекурентним виразом.

ДОДАТОК А

Лістинг коду програми до завдання Func24.

import tkinter as tk

from tkinter import messagebox

class MeanCalculatorApp(tk.Frame):

def \_\_init\_\_(self, parent):

super().\_\_init\_\_(parent)

self.parent = parent

self.pack(fill=tk.BOTH, expand=1)

self.create\_widgets()

def create\_widgets(self):

"""Створення віджетів."""

tk.Label(self, text="Enter A:").grid(row=0, column=0)

self.a\_entry = tk.Entry(self)

self.a\_entry.grid(row=0, column=1)

tk.Label(self, text="Enter B:").grid(row=1, column=0)

self.b\_entry = tk.Entry(self)

self.b\_entry.grid(row=1, column=1)

tk.Label(self, text="Enter C:").grid(row=2, column=0)

self.c\_entry = tk.Entry(self)

self.c\_entry.grid(row=2, column=1)

tk.Label(self, text="Enter D:").grid(row=3, column=0)

self.d\_entry = tk.Entry(self)

self.d\_entry.grid(row=3, column=1)

self.calc\_button = tk.Button(self, text="Calculate", command=self.calculate\_means)

self.calc\_button.grid(row=4, column=0, columnspan=2)

self.output\_label = tk.Label(self, text="")

self.output\_label.grid(row=5, column=0, columnspan=2)

def mean(self, x, y):

"""Обчислення середнього арифметичного та геометричного."""

arithmetic\_mean = (x + y) / 2

geometric\_mean = (x \* y) \*\* 0.5

return arithmetic\_mean, geometric\_mean

def calculate\_means(self):

"""Зчитування даних"""

try:

A = float(self.a\_entry.get())

B = float(self.b\_entry.get())

C = float(self.c\_entry.get())

D = float(self.d\_entry.get())

results = [

self.mean(A, B),

self.mean(A, C),

self.mean(A, D)

]

output\_text = (

f"(A, B): Arithmetic = {results[0][0]:.2f}, Geometric = {results[0][1]:.2f}\n"

f"(A, C): Arithmetic = {results[1][0]:.2f}, Geometric = {results[1][1]:.2f}\n"

f"(A, D): Arithmetic = {results[2][0]:.2f}, Geometric = {results[2][1]:.2f}"

)

self.output\_label.config(text=output\_text)

except ValueError:

messagebox.showerror("Помилка", "Введіть дійсні числа!")

if \_\_name\_\_ == "\_\_main\_\_":

root = tk.Tk()

root.title("Mean Calculator")

app = MeanCalculatorApp(root)

root.mainloop()

ДОДАТОК Б

Лістинг коду програми до завдання 2 побудова графіка функції.

import tkinter as tk

from tkinter import messagebox, filedialog

from matplotlib.figure import Figure

from matplotlib.backends.backend\_tkagg import FigureCanvasTkAgg

class RecurrenceApp(tk.Frame):

"""побудова графіка рекурентного виразу."""

def \_\_init\_\_(self, parent):

super().\_\_init\_\_(parent)

self.parent = parent

self.pack(fill=tk.BOTH, expand=1)

self.create\_widgets()

def create\_widgets(self):

"""Створення віджетів."""

# Buttons

self.save\_button = tk.Button(self, text="Save Data to File", command=self.save\_data)

self.save\_button.pack()

self.plot\_button = tk.Button(self, text="Show Plot", command=self.show\_plot)

self.plot\_button.pack()

self.canvas\_frame = tk.Frame(self)

self.canvas\_frame.pack(fill=tk.BOTH, expand=1)

def calculate\_recurrence(self, n=100, T0=0.3, K=2.5, U=2):

"""Обчислення значень y за рекурентним виразом."""

y = [0]

for k in range(n):

y\_next = (1 - T0 / 0.3) \* y[k] + (T0 / 0.3) \* K \* U

y.append(y\_next)

return [k \* T0 for k in range(n + 1)], y

def save\_data(self):

"""Збереження даних у файл."""

t, y = self.calculate\_recurrence()

file = filedialog.asksaveasfile(mode='w', defaultextension=".txt")

if file:

for i in range(len(t)):

file.write(f"{t[i]:.5f};{y[i]:.5f}\n")

file.close()

messagebox.showinfo("успіх", "Дані збережено!")

def show\_plot(self):

"""Побудова графіка."""

t, y = self.calculate\_recurrence()

fig = Figure(figsize=(5, 4), dpi=100)

ax = fig.add\_subplot(111)

ax.plot(t, y, label="y(t)")

ax.set\_title("Рекурентний графік")

ax.set\_xlabel("Time (t)")

ax.set\_ylabel("y(t)")

ax.grid(True)

ax.legend()

canvas = FigureCanvasTkAgg(fig, master=self.canvas\_frame)

canvas.get\_tk\_widget().pack(fill=tk.BOTH, expand=1)

canvas.draw()

if \_\_name\_\_ == "\_\_main\_\_":

root = tk.Tk()

root.title("Recurrence Plotter")

app = RecurrenceApp(root)

root.mainloop()

ДОДАТОК В

Скрін-шоти вікна виконання програми

Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис

Рисунок Б.3 – Екран виконання програми для вирішення завдання Func24.

Зображення, що містить текст, знімок екрана, Графік, ряд

Автоматично згенерований опис

Рисунок Б.4 – Збереження даних в текстовому файлі завдання 2. побудова графіка функції