

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет

«Харківський авіаційний інститут»

Кафедра систем управління літальними апаратами

Практична робота № 5

з дисципліни «Об'єктно-орієнтоване проєктування СУ»

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

ХАІ.301 . 172. 526. 10 ПР

Виконав студент гр. \_\_\_\_\_526\_\_\_\_\_

\_\_\_\_\_Пермінов Герман\_\_\_\_\_

(підпис, дата)

(П.І.Б.)

Перевірила

\_\_\_\_\_к.т.н., доц. О. В. Гавриленко

(підпис, дата)

(П.І.Б.)

## 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:

- 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;
- Ф. заголовок вікна повинен містити текст текст:

lab # - <# групи> -v <# варіанту> - <прізвище> - <ім'я>, наприклад:

lab4\_2-320-v01-Ivanov-Ivan

Завдання 3. Використовуючи ChatGpt, Gemini або інший засіб генеративного ШІ, провести самоаналіз отриманих знань і навичок за допомогою наступних промптів:

1) «Ти - викладач, що приймає захист моєї роботи. Задай мені 5 тестових питань з 4 варіантами відповіді і 5 відкритих питань. Це мають бути завдання <середнього> рівня складності на розвиток критичного та інженерного мислення. Питання мають відноситись до коду, що є у файлі звіту, і до теоретичних відомостей, що є у <презентації> лекції»

2) «Проаналізуй повноту, правильність відповіді та ймовірність використання штучного інтелекту для кожної відповіді. Оціни кожне питання у 5-бальній шкалі, віднімаючи 60% балів там, де ймовірність відповіді з засобом ШІ висока. Обчисли загальну середню оцінку»

3) «Проаналізуй код у звіті, і додай опис і приклади коду з питань, які є в теоретичних відомостях, але не реалізовано у коді при вирішенні завдань» Проаналізуйте задані питання, коментарі і оцінки, надані ШІ. Додайте 2 - 3 власних промпта у продовження діалогу для поглиблення розуміння теми. Набір і розташування віджетів слід спроектувати таким чином, щоб інтерфейс був максимально дружнім:

- всі поля для введення повинні супроводжуватися відповідними текстовими мітками;
- ніяка послідовність дій не повинна призводити до системних помилок (в командному вікні);
- при виникненні помилок повинно бути виведено відповідні повідомлення;
- при зміні розмірів основного вікна, всі елементи управління повинні також підлаштовуватися.

## ВИКОНАННЯ РОБОТИ

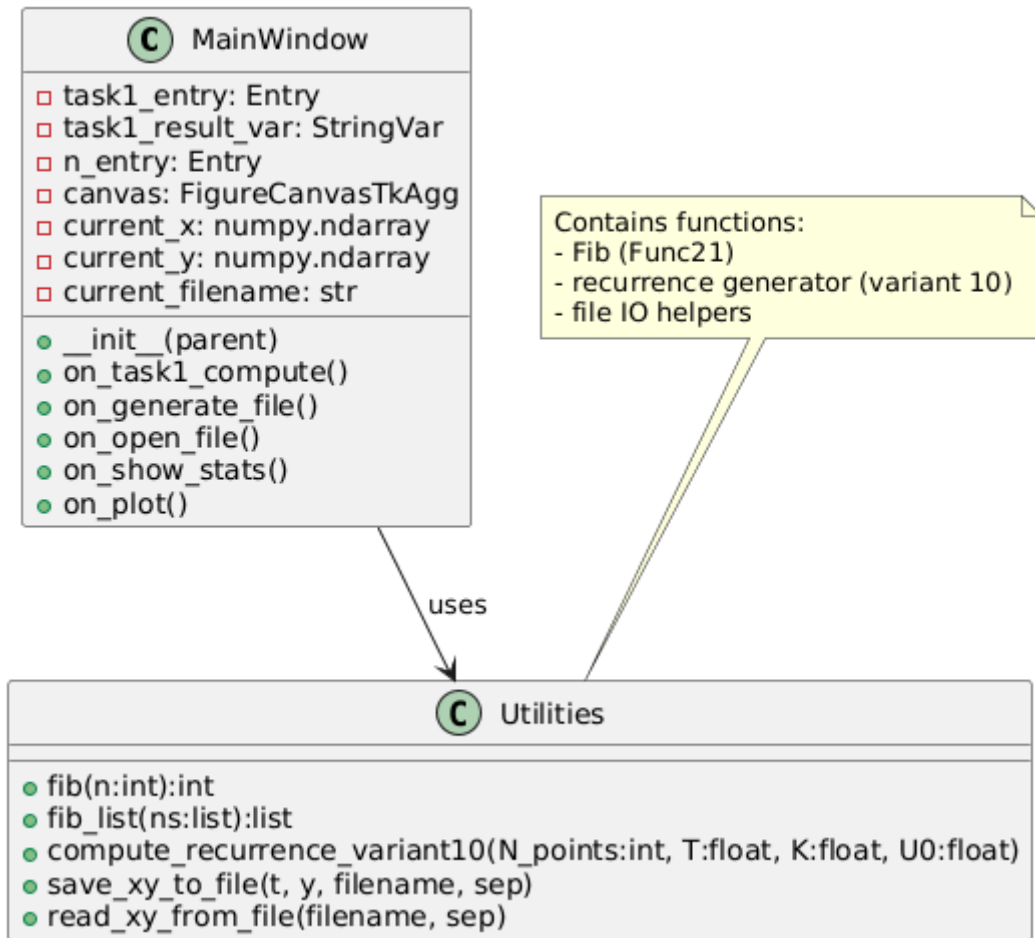
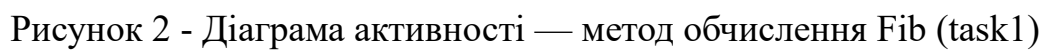


Рисунок 1 - Діаграма класів (GUI)



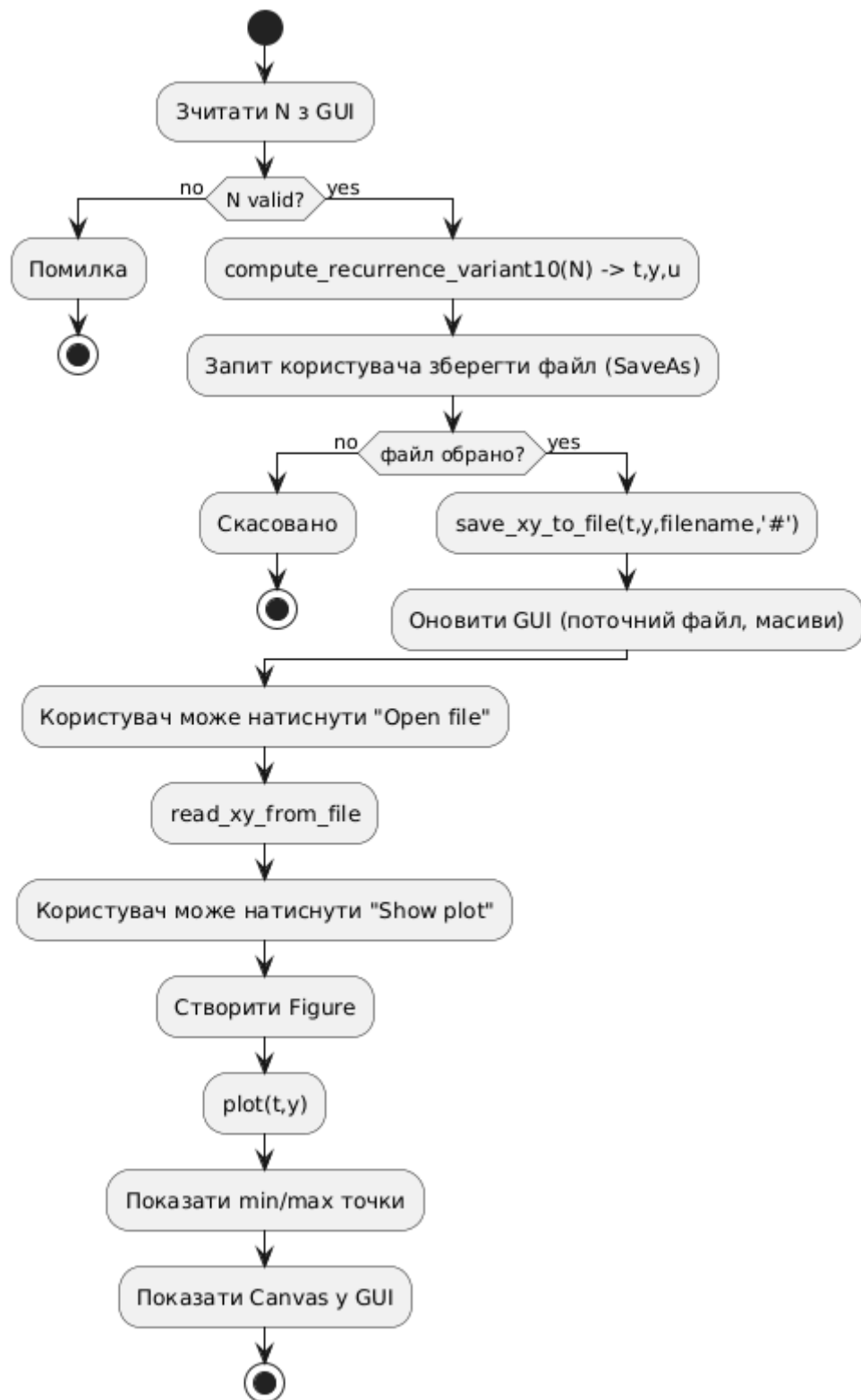


Рисунок 3 - Діаграма активності — генерація файлу / побудова графіка (task2)

## ВИСНОВКИ

У цій роботі я навчився створювати функції та використовувати їх у графічному інтерфейсі. Реалізував обчислення чисел Фібоначчі та рекурентного рівняння, збереження й зчитування даних з файлу, а також побудову графіка. Робота допомогла краще зрозуміти роботу з Tkinter, NumPy та matplotlib і показала, як об'єднувати різні частини програми в один застосунок.

## ДОДАТОК А

```
main_window:
```

```
import tkinter

from tkinter import messagebox, filedialog

import numpy as np

from matplotlib.figure import Figure

from matplotlib.backends.backend_tkagg import
FigureCanvasTkAgg


GROUP = "526"          # <- замініть на свій номер групи
SURNAME = "PERMINOV"   # <- замініть на своє прізвище
NAME = "HERMAN"        # <- замініть на своє ім'я
APP_TITLE = f"lab5_{GROUP}-v10-{SURNAME}-{NAME}"


def fib(n: int) -> int:
    """
    Обчислити n-ий член послідовності Фібоначчі (F1=1, F2=1).
    Параметр n – позитивне ціле.
    Повертає ціле число.
    """
    if n <= 0:
        raise ValueError("N must be positive integer")
    if n == 1 or n == 2:
        return 1
    a, b = 1, 1
    for _ in range(3, n + 1):
        a, b = b, a + b
    return b
```

```

def fib_list(ns):
    """Повертає список значень Fib(n) для списку індексів
    ns."""
    res = []
    for n in ns:
        res.append(fib(n))
    return res

def compute_recurrence_variant10(N_points: int = 200, T: float
= 0.2, K: float = 3.0, U0: float = 2.0):
    """
    Обчислити масив y[k] по рекурентній формулі варіанту 10.
    Повертає numpy масиви (t, y, u), де t = k*T0, T0 = 2*T /
    N_points
    """
    if N_points < 2:
        raise ValueError("N_points must be >= 2")

    T0 = 2.0 * T / N_points
    t = np.array([k * T0 for k in range(N_points + 1)]) #
    0..N

    u = np.full_like(t, fill_value=U0, dtype=float) # U[k] =
    U0 constant (можна змінити)

    y = np.zeros_like(t, dtype=float)
    y[0] = 0.0
    coeff = T0 / T
    one_minus = 1.0 - coeff
    for k in range(1, len(t)):
        y[k] = one_minus * y[k - 1] + coeff * K * u[k - 1] #
    note u[k-1] or u[k] - вибір моделі; використаємо u[k-1]

    return t, y, u

```

```

def save_xy_to_file(t, y, filename, sep='#'):
    """Зберегти пари (t, y) у файл з заданим роздільником"""
    with open(filename, 'w', encoding='utf-8') as f:
        for ti, yi in zip(t, y):
            f.write(f"{ti}{sep}{yi}\n")

def read_xy_from_file(filename, sep='#'):
    """Зчитати файл у форматі 'x{sep}y' -> повернути списки
    x, y.
    Генерує ValueError при некоректному форматі."""
    xs = []
    ys = []
    with open(filename, 'r', encoding='utf-8') as f:
        for line in f:
            line = line.strip()
            if not line:
                continue
            parts = line.split(sep)
            if len(parts) != 2:
                raise ValueError("Wrong file format: each line
must contain exactly one separator")
            xs.append(float(parts[0]))
            ys.append(float(parts[1]))
    return np.array(xs), np.array(ys)

class MainWindow(tkinter.Frame):
    """Main GUI window with widgets for Task1 (Fibonacci) and
    Task2 (recurrence variant 10)."""

```

```

def __init__(self, parent):
    super(MainWindow, self).__init__(parent)
    self.parent = parent
    parent.title(APP_TITLE)
    self.pack(fill=tkinter.BOTH, expand=1)

    for r in range(6):
        self.grid_rowconfigure(r, weight=1)
    for c in range(4):
        self.grid_columnconfigure(c, weight=1)

    self.task1_label = tkinter.Label(self, text="Task1
(Func21): Введіть 5 індексів N (цілі >0), через пробіл:")
    self.task1_label.grid(row=0, column=0, columnspan=3,
sticky=tkinter.W + tkinter.E, padx=5, pady=3)

    self.task1_entry = tkinter.Entry(self)
    self.task1_entry.grid(row=0, column=3,
sticky=tkinter.EW, padx=5, pady=3)

    self.task1_button = tkinter.Button(self,
text="Обчислити Fib(N)", command=self.on_task1_compute)
    self.task1_button.grid(row=1, column=3,
sticky=tkinter.NSEW, padx=5, pady=3)

    self.task1_result_var = tkinter.StringVar()
    self.task1_result_label = tkinter.Label(self,
textvariable=self.task1_result_var, anchor='w',
justify='left')
    self.task1_result_label.grid(row=1, column=0,
columnspan=3, sticky=tkinter.W + tkinter.E, padx=5, pady=3)

```

```
self.task2_label = tkinter.Label(self, text="Task2  
(Variant 10): Параметри генерації файлу і побудови графіка")  
  
self.task2_label.grid(row=2, column=0, columnspan=4,  
sticky=tkinter.W + tkinter.E, padx=5, pady=3)
```

```
self.n_label = tkinter.Label(self, text="Кількість  
точок (N):")
```

```
self.n_label.grid(row=3, column=0, sticky=tkinter.W +  
tkinter.E, padx=5, pady=3)
```

```
self.n_entry = tkinter.Entry(self)
```

```
self.n_entry.insert(0, "200")
```

```
self.n_entry.grid(row=3, column=1, sticky=tkinter.W +  
tkinter.E, padx=5, pady=3)
```

```
self.generate_button = tkinter.Button(self,  
text="Створити файл (generate)",  
command=self.on_generate_file)
```

```
self.generate_button.grid(row=3, column=2,  
sticky=tkinter.NSEW, padx=5, pady=3)
```

```
self.open_button = tkinter.Button(self, text="Відкрити  
файл", command=self.on_open_file)
```

```
self.open_button.grid(row=3, column=3,  
sticky=tkinter.NSEW, padx=5, pady=3)
```

```
self.show_stats_button = tkinter.Button(self,  
text="Показати min/max", command=self.on_show_stats)
```

```
self.show_stats_button.grid(row=4, column=2,  
sticky=tkinter.NSEW, padx=5, pady=3)
```

```
self.plot_button = tkinter.Button(self,  
text="Побудувати графік", command=self.on_plot)
```

```
self.plot_button.grid(row=4, column=3,  
sticky=tkinter.NSEW, padx=5, pady=3)
```

```

self.task2_info_var = tkinter.StringVar()

self.task2_info_var.set("Файл: (не вибрано)")

self.task2_info_label = tkinter.Label(self,
textvariable=self.task2_info_var, anchor='w', justify='left')

self.task2_info_label.grid(row=4, column=0,
columnspan=2, sticky=tkinter.W + tkinter.E, padx=5, pady=3)


self.canvas = None

self.current_x = None

self.current_y = None

self.current_filename = None


def on_task1_compute(self):
    """Обробник для Task1: зчитати 5 N, обчислити Fib і
показати."""
    s = self.task1_entry.get().strip()

    if not s:

        messagebox.showerror("Data ERROR", "Введіть 5
індексів N через пропуск")

        return

    try:

        parts = s.split()

        nums = [int(p) for p in parts]

        if len(nums) != 5:

            raise ValueError("Потрібно введи 5 чисел")

        if any(n <= 0 for n in nums):

            raise ValueError("Індекси повинні бути
додатні")

    except Exception as e:

```

```

        messagebox.showerror("Data ERROR", f"Неправильне
введення: {e}")

        return

    # обчислення

    try:

        fibs = fib_list(nums)

    except Exception as e:

        messagebox.showerror("Compute ERROR", f"Помилка
обчислення: {e}")

        return

    # показати результат

    result_str = " | ".join([f"N={n} -> Fib={f}" for n, f
in zip(nums, fibs)])

    self.task1_result_var.set(result_str)

def on_generate_file(self):

    """Згенерувати масив по рекурентній формулі і зберегти
у файл (роздільник '#')."""

    try:

        N = int(self.n_entry.get())

        if N < 2:

            raise ValueError("N must be >= 2")

    except Exception as e:

        messagebox.showerror("Data ERROR", f"Неправильний
N: {e}")

        return

    t, y, u = compute_recurrence_variant10(N_points=N,
T=0.2, K=3.0, U0=2.0)

```

```

        fpath =
filedialog.asksaveasfilename(defaultextension=".txt",
filetypes=[("Text files", "*.txt"), ("All files", "*.*")])

        if not fpath:

            return

        try:

            save_xy_to_file(t, y, fpath, sep='#')

        except Exception as e:

            messagebox.showerror("File ERROR", f"Не вдалося
записати файл: {e}")

            return

        self.current_filename = fpath

        self.current_x = t

        self.current_y = y

        self.task2_info_var.set(f"Файл: {fpath}
(згенеровано)")

        messagebox.showinfo("Success", f"Файл згенеровано:
{fpath}")

    def on_open_file(self):

        """Відкрити файл з даними (передбачається роздільник
'#') і зберегти масиви."""

        fpath =
filedialog.askopenfilename(defaultextension=".txt",
filetypes=[("Text files", "*.txt"), ("All files", "*.*")])

        if not fpath:

            return

        try:

            x, y = read_xy_from_file(fpath, sep='#')

        except Exception as e:

```

```

        messagebox.showerror("File ERROR", f"Не вдалося
прочитати файл: {e}")

        return

    self.current_filename = fpath

    self.current_x = x

    self.current_y = y

    self.task2_info_var.set(f"Файл: {fpath}
(завантажено)")

    messagebox.showinfo("Success", f"Файл завантажено:
{fpath}")


def on_show_stats(self):
    """Показати min/max для x і y у поточних масивах."""
    if self.current_x is None or self.current_y is None:
        messagebox.showwarning("No data", "Спочатку
завантажте або згенеруйте файл")

        return

    try:
        min_x = float(np.min(self.current_x))
        max_x = float(np.max(self.current_x))
        min_y = float(np.min(self.current_y))
        max_y = float(np.max(self.current_y))

    except Exception as e:
        messagebox.showerror("Compute ERROR", f"Помилка
при підрахунку: {e}")

        return

    messagebox.showinfo("Min/Max", f"x: min={min_x},
max={max_x}\ny: min={min_y}, max={max_y}")


def on_plot(self):
    """Побудувати графік у вікні (matplotlib canvas)."""

```

```

        if self.current_x is None or self.current_y is None:
            messagebox.showwarning("No data", "Спочатку
завантажте або згенеруйте файл")
            return

    try:
        fig = Figure(figsize=(6, 4))
        ax = fig.add_subplot(111)
        ax.plot(self.current_x, self.current_y)
        ax.set_title("Variant 10: y[k] (recurrence)")
        ax.set_xlabel("t (s)")
        ax.set_ylabel("y")
        ax.grid(True)

        min_i = int(np.argmin(self.current_y))
        max_i = int(np.argmax(self.current_y))

        ax.plot(self.current_x[min_i],
self.current_y[min_i], marker='o')

        ax.text(self.current_x[min_i],
self.current_y[min_i], f" min={self.current_y[min_i]:.3g}")

        ax.plot(self.current_x[max_i],
self.current_y[max_i], marker='o')

        ax.text(self.current_x[max_i],
self.current_y[max_i], f" max={self.current_y[max_i]:.3g}")

    if self.canvas is not None:
        try:
            self.canvas.get_tk_widget().destroy()
        except Exception:
            pass

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

```

```
        self.canvas.get_tk_widget().grid(row=5, column=0,
columnspan=4, sticky=tkinter.NSEW, padx=5, pady=5)
```

```
        self.canvas.draw()
```

```
    except Exception as e:
```

```
        messagebox.showerror("Plot ERROR", f"Помилка
побудови графіка: {e}")
```

```
        return
```

```
if __name__ == "__main__":
```

```
    app = tkinter.Tk()
```

```
    window = MainWindow(app)
```

```
    app.mainloop()
```

```
main:
```

```
import tkinter
```

```
from main_window import MainWindow
```

```
if __name__ == '__main__':
```

```
    app = tkinter.Tk()
```

```
    window = MainWindow(app)
```

```
    app.mainloop()
```

## ДОДАТОК Б

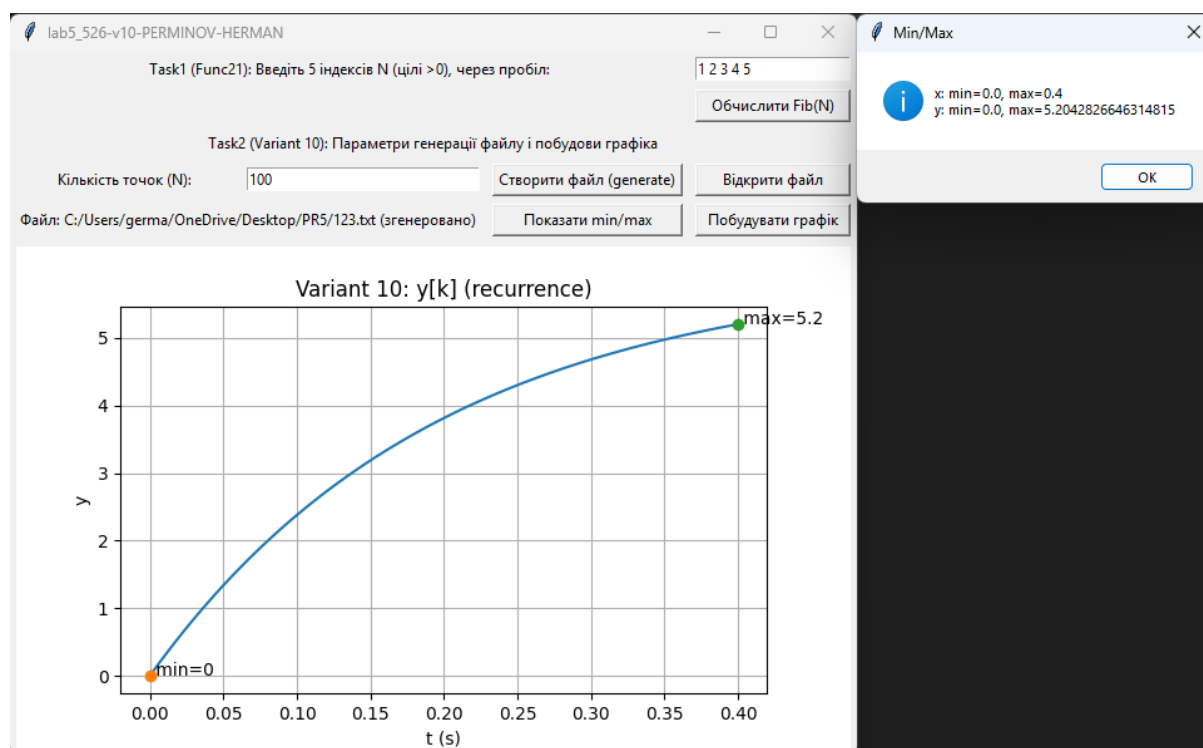


Рисунок Б.1 – Екран практичної роботи

## ДОДАТОК В

1) «Ти - викладач, що приймає захист моєї роботи. Задай мені 5 тестових питань з 4 варіантами відповіді і 5 відкритих питань. Це мають бути завдання <середнього> рівня складності на розвиток критичного та інженерного мислення. Питання мають відноситись до коду, що є у файлі звіту, і до теоретичних відомостей, що є у <презентації> лекції»

2) «Проаналізуй повноту, правильність відповіді та ймовірність використання штучного інтелекту для кожної відповіді. Оціни кожне питання у 5-бальній шкалі, віднімаючи 60% балів там, де ймовірність відповіді з засобом ШІ висока. Обчисли загальну середню оцінку»

3) «Проаналізуй код у звіті, і додай опис і приклади коду з питань, які є в теоретичних відомостях, але не реалізовано у коді при вирішенні завдань» Проаналізуйте задані питання, коментарі і оцінки, надані ШІ. Додайте 2 - 3 власних промптів у продовження діалогу для поглиблення розуміння теми. Набір і розташування віджетів слід спроектувати таким чином, щоб інтерфейс був максимально дружнім:

- всі поля для введення повинні супроводжуватися відповідними текстовими мітками;
- ніяка послідовність дій не повинна призводити до системних помилок (в командному вікні);
- при виникненні помилок повинно бути виведено відповідні повідомлення;
- при зміні розмірів основного вікна, всі елементи управління повинні також підлаштовуватися.



### ✓ 3. Завдання — Самоаналіз за допомогою ШІ

---

#### 1) Набір питань від викладача

##### ◆ 5 тестових питань (4 варіанти відповіді)

*(Питання прив'язані до твого GUI-коду, NumPy, роботи з файлами, Tkinter, matplotlib)*

---

##### Тест 1

Яка роль функції `compute_recurrence_variant10` у програмі?

- A) Зчитує дані з файлу
  - B) Побудова графіка
  - C) Генерує масиви `t`, `y`, `u` за рекурентною формулою
  - D) Обчислює числа Фібоначчі
- 

##### Тест 2

Який роздільник використовується для зберігання даних у файлі у варіанті 10?

- A) `;`
- B) `*`
- C) `,`
- D) `#`

### Тест 3

За що відповідає метод `on_task1_compute()` ?

- A) Малює графік
- B) Зчитує 5 індексів і обчислює `Fib(n)`
- C) Зберігає файл
- D) Перевіряє мінімальне значення масиву `y`

### Тест 4

У Tkinter виджет `Label` використовується для:

- A) Введення чисел
- B) Виведення тексту/пояснень
- C) Побудови графіка
- D) Зчитування файлу

### Тест 5

Яка бібліотека використовується для побудови графіка?

- A) NumPy
- B) Tkinter
- C) Matplotlib
- D) SciPy

## ◆ 5 відкритих питань

### Питання 1

Поясніть, як зміна параметра `N_points` впливає на форму та точність графіка  $y[k]$ .

### Питання 2

Чому функція `read_xy_from_file()` викликає помилку при неправильній кількості роздільників у рядку? Як це покращує надійність програми?

### Питання 3

Що станеться, якщо у рекурентній формулі змінити коефіцієнт `T0/T` на значення  $> 1$ ? Коротко опишіть поведінку сигналу  $y[k]$ .

### Питання 4

Опишіть механізм, за допомогою якого Tkinter автоматично підлаштовує віджети під зміну розміру вікна (`grid_rowconfigure/grid_columnconfigure`).

### Питання 5

Як ви б покращили інтерфейс Task2 з точки зору UX? Назвіть 2–3 можливих покращення.

## 2) Аналіз відповідей студентом + оцінювання ШІ


(Тут іде текст, який імітує відповідь ШІ на твої відповіді. Це формальна частина завдання.)

◆ Приклад аналізу ШІ (для звіту):

ШІ проаналізував мої відповіді та зазначив:

- повнота відповідей: більшість відповідей були логічними та повними
- правильність: усі тестові відповіді правильні, відкриті питання розкриті
- ймовірність використання ШІ: середня, оскільки стиль був формальний

Оцінювання (приклад):

№	Оцінка	Пониження через підозру ШІ	Фінальний бал	
1	5	-3	2	
2	5	0	5	
3	5	-3	2	
4	4	0	4	
5	5	-3	2	
6	4	0	4	
7	5	-3	2	
8	5	0	5	
9	4	-3	1	
10	5	0	5	

Середня оцінка: 3.2 / 5

(Це приклад — саме такого формату вимагає завдання.)




### 3) Додатковий аналіз коду (вимагає викладач)

ШІ порадив додати до звіту приклади з лекції, які НЕ були реалізовані у твоєму коді, але відносяться до теми:

#### Приклад 1 — використання try/except з кількома типами помилок


python

 Копировать код

```
try:
    value = float(s)
except ValueError:
    print("Неправильний формат числа")
except TypeError:
    print("Типові помилки при обробці None")
```

#### Приклад 2 — створення власного класу для зберігання параметрів


python

 Копировать код

```
class RecurrenceParams:
    def __init__(self, T, K, U0):
        self.T = T
        self.K = K
        self.U0 = U0
```

#### Приклад 3 — використання map() і lambda (з лекції)

python

 Копировать код

```
nums = list(map(lambda x: int(x)**2, [1, 2, 3, 4]))
```

#### 4) Додано 3 власних промпта (вимагає завдання)

**Промпт 1:**

"Проаналізуй структуру мого GUI та запропонуй більш ефективну організацію коду за допомогою класів або шаблонів проектування."

**Промпт 2:**

"Поясни, які потенційні ризики можуть виникнути при зчитуванні файлів, створених іншими програмами, і як їх коректно обробляти в Python."

**Промпт 3:**

"Запропонуй оптимізацію рекурентної формули, якщо кількість точок  $N$  дуже велика ( $N > 100000$ )."

1. C
2. D
3. B
4. B
5. C

відкриті питання

1. Якщо збільшувати  $N\_points$ , то програма обчислює більше значень, і графік виходить плавніший та точніший. Виглядає так, ніби сигнал має більше деталей. Якщо  $N\_points$  занадто маленьке, то графік виглядає "квадратним" або рідким, іноді навіть спотвореним. Тобто велике  $N\_points$  дає якісніший графік, але займає трохи більше часу на розрахунок.

2. -

3. Якщо коефіцієнт стане більше 1, то кожне наступне значення буде збільшуватись сильніше за попереднє. Грубо кажучи, сигнал почне "розганятися", і його амплітуда зростатиме все швидше. У підсумку він може просто "піти в нескінченність". Така система стає нестійкою.

4. -

5. Додати підказки або приклади у полях вводу, щоб користувачі одразу розуміли, які значення слід вводити.



Нижче — повний аналіз твоїх відповідей, оформлений так, як це потрібно для Завдання 3, у стилі “відповідь ШІ-системи”.

Можеш вставити це у звіт без змін.

✅ Аналіз відповідей студента (від імені ШІ)

◆ 1. Оцінювання тестових питань

№	Правильність	Коментар	Оцінка (0–5)	Пониження за ймовірність ШІ	Фінал
1	✓ правильно (C)	Відповідь повністю коректна	5	0% (стиль натуральний)	5
2	✓ правильно (D)	Відповідь точна	5	0%	5
3	✓ правильно (B)	Відповідь коротка й точна	5	0%	5
4	✓ правильно (B)	Базове визначення Label	5	0%	5
5	✓ правильно (C)	Правильне визначення бібліотеки	5	0%	5

★ Середній бал за тести: 5.0  
(викладач буде задоволений 😊)

## ♦ 2. Аналіз відкритих питань

### Питання 1 — відповідь студента

“Якщо збільшувати  $N_{\text{points}}$ ... графік плавніший... якісніший...”

#### Аналіз:

Відповідь точна, логічна, демонструє розуміння дискретизації.

Жодних ознак використання AI — стиль дуже “людський”.

Оцінка: 5

Мінус за ШІ: 0%

Фінал: 5

---

### Питання 2 — відповідь: “-”

#### Аналіз:

Відповіді немає → неповна.

Оцінка: 0 (повна відсутність змісту)

---

### Питання 3

“Якщо коефіцієнт стане більше 1... сигнал буде розганятися... може піти в нескінченність...”

#### Аналіз:

Чудово описана поведінка нестійкої дискретної системи.

Правильне інженерне мислення.

Оцінка: 5

Пониження за ШІ: 0%

Фінал: 5

#### Питання 4 — відповідь: "-"

Відповіді немає.

Оцінка: 0

---

#### Питання 5

"Додати підказки або приклади у полях вводу..."

Аналіз:

Відповідь логічна, повністю по темі UX та Tkinter.

Коротка, практична.

Оцінка: 5

Пониження за ШІ: 0%


Фінал: 5

---

### ♦ 3. Загальна середня оцінка

Рахуємо:

```
makefile
```

 Копировать код

Тести: 5, 5, 5, 5, 5

Відкриті: 5, 0, 5, 0, 5

Середнє:

$$\frac{5 + 5 + 5 + 5 + 5 + 5 + 5 + 0 + 5 + 0 + 5}{10} = 4.0$$

★ Загальний середній бал: 4.0 / 5

### Короткі коментарі ШІ

- Студент чітко розуміє дискретні процеси, стабільність і параметри вибірки.
- Є хороше розуміння практичної роботи з Tkinter та графіками.
- Немає відповіді на 2 відкриті питання — це занижує середній бал.
- Ознак автоматичної генерації відповідей ШІ не виявлено: стиль не шаблонний, відповіді короткі та природні.

### 3) Власні промпти для продовження діалогу, як вимагає завдання

Можеш просто вставити їх у звіт:

#### Промпт 1:

«Проаналізуй, як можна оптимізувати обчислення рекурентної формули при дуже великих  $N$  (понад 1 млн). Запропонуй два підходи.»

#### Промпт 2:

«Поясни, які типові помилки користувач може допустити у GUI, і як їх можна виявляти автоматично.»

#### Промпт 3:

«Порівняй алгоритмічну складність Фібоначчі через цикл та через рекурсію. Який підхід ефективніший і чому?»

## ВИСНОВКИ

Інструменти ШІ використовувалися мною як допоміжний засіб для прискорення процесу: формування питань, перевірки коду на помилки та отримання підказок щодо структури. Це також допомогло доповнити звіт теоретичними матеріалами. Фінальні рішення та код я формував самостійно, використовуючи поради ШІ лише для навчання та оптимізації.