### **Часть 1: Получение данных из API и отображение их в интерфейсе**

Для начала создадим базовый функционал получения данных от Google Books API, их обработки и отображения в графическом интерфейсе.

#### **1.1 Обработка ответа с API**

Предположим, что мы получаем данные о книгах с API, фильтруем их и отображаем название книги, авторов и описание.

import requests

import random

import time

def get\_book\_recommendations(api\_key, query="fiction", max\_results=5):

"""

Получает рекомендации книг из Google Books API с использованием случайных параметров.

"""

start\_index = random.randint(0, 50) # Случайный индекс, чтобы получать разные результаты

timestamp = int(time.time())

url = f"https://www.googleapis.com/books/v1/volumes?q={query}&langRestrict=ru&maxResults={max\_results}&key={api\_key}&startIndex={start\_index}&t={timestamp}"

try:

response = requests.get(url)

response.raise\_for\_status() # Проверяем статус ответа

data = response.json()

recommendations = []

for item in data.get("items", []):

volume\_info = item.get("volumeInfo", {})

title = volume\_info.get("title", "Без названия")

authors = volume\_info.get("authors", ["Неизвестный автор"])

description = volume\_info.get("description", "Описание отсутствует.")

thumbnail = volume\_info.get("imageLinks", {}).get("thumbnail", "") # Ссылка на обложку

recommendations.append({

"title": title,

"authors": ", ".join(authors),

"description": description,

"thumbnail": thumbnail

})

return recommendations

except requests.RequestException as e:

print(f"Ошибка при запросе к API: {e}")

return []

#### **1.2 Добавление виджетов в Tkinter**

Теперь добавим элементы интерфейса в Tkinter для ввода данных, выполнения запросов и отображения результатов.

def show\_recommendations\_window(api\_key):

"""

Открывает окно с рекомендациями книг, отображает обложки, названия и описания.

"""

recommendations\_window = tk.Toplevel()

recommendations\_window.title("Рекомендации книг")

recommendations\_window.configure(bg="#2E2E2E")

recommendations\_window.attributes('-fullscreen', True)

# Заголовок окна

label\_style = {"font": ("Segoe UI", 24, "bold"), "bg": "#2E2E2E", "fg": "white"}

tk.Label(recommendations\_window, text="Рекомендации книг", \*\*label\_style).pack(pady=20)

# Создание области с прокруткой

canvas = tk.Canvas(recommendations\_window, bg="#2E2E2E", highlightthickness=0)

scrollbar = tk.Scrollbar(recommendations\_window, orient="vertical", command=canvas.yview)

scrollable\_frame = tk.Frame(canvas, bg="#2E2E2E")

scrollable\_frame.bind(

"<Configure>",

lambda e: canvas.configure(scrollregion=canvas.bbox("all"))

)

canvas.create\_window((0, 0), window=scrollable\_frame, anchor="n") # Центровка по верхнему краю

canvas.configure(yscrollcommand=scrollbar.set)

canvas.pack(side="left", fill="both", expand=True)

scrollbar.pack(side="right", fill="y")

# Получение новых рекомендаций

recommendations = get\_book\_recommendations(api\_key)

# Очистка старых рекомендаций перед выводом новых

for widget in scrollable\_frame.winfo\_children():

widget.destroy()

# Стили для текста

book\_title\_style = {"font": ("Segoe UI", 16, "bold"), "bg": "#2E2E2E", "fg": "white", "anchor": "center"}

book\_desc\_style = {"font": ("Segoe UI", 14), "bg": "#2E2E2E", "fg": "white", "anchor": "center"}

for rec in recommendations:

# Фрейм для одной книги

frame = tk.Frame(scrollable\_frame, bg="#2E2E2E", pady=15)

frame.pack(fill="x", padx=20)

# Обложка книги

image\_label = tk.Label(frame, bg="#2E2E2E")

image\_label.pack(pady=5, anchor="center")

if rec["thumbnail"]:

response = requests.get(rec["thumbnail"])

image\_data = BytesIO(response.content)

img = Image.open(image\_data).resize((200, 200)) # Размер обложки уменьшен

photo = ImageTk.PhotoImage(img)

image\_label.configure(image=photo)

image\_label.image = photo

# Название книги

title\_label = tk.Label(frame, text=rec["title"], \*\*book\_title\_style, wraplength=600, justify="center")

title\_label.pack(pady=5)

# Описание книги

desc\_label = tk.Label(frame, text=rec["description"], \*\*book\_desc\_style, wraplength=600, justify="center")

desc\_label.pack(pady=5)

# Кнопка закрытия

tk.Button(recommendations\_window, text="Закрыть", command=recommendations\_window.destroy,

font=("Segoe UI", 16), bg="#444444", fg="white").pack(pady=20)

### **Заключение**

Теперь мой проект включает интеграцию данных из Google Books API и отображение этих данных в графическом интерфейсе Tkinter