**Лабораторная работа №9**

**РАЗРАБОТКА GUI ПРИЛОЖЕНИЯ С ПОМОЩЬЮ ГРАФИЧЕСКИХ БИБЛИОТЕК**

**Цель работы**: рассмотреть возможности библиотеки Tkinter, её базовые виджеты и изучить основные принципы создания приложений с графическим интерфейсом с помощью данной библиотеки.

**Задание:** написать GUI приложение, которое представляет собой библиотеку с поиском книг по автору и наименованию, добавлению новых книг и чтению книг в форматах epub, fb2 и pdf.

**Выполнение задания:** Пользователь может искать книги по автору и/или названию, добавлять новые книги, выбирая файлы PDF, EPUB и отображать содержимое выбранной книги в новом окне. Обработка разных форматов реализована с помощью библиотеки fitz для PDF и ebooklib для EPUB. Неподдерживаемые форматы приводят к сообщению об ошибке.

**Листинг программы:**

*import os  
import tkinter as tk  
from tkinter import filedialog, messagebox, Listbox  
import fitz  
from ebooklib import epub  
from tkinter import simpledialog  
  
class LibraryApp:  
 def \_\_init\_\_(self, root):  
 self.root = root  
 self.root.title("Библиотека")  
 self.root.geometry("600x400")  
  
 self.library = []  
  
 self.setup\_ui()  
  
 def setup\_ui(self):  
 self.search\_frame = tk.Frame(self.root)  
 self.search\_frame.pack(pady=10)  
  
 tk.Label(self.search\_frame, text="Автор:").grid(row=0, column=0, padx=5)  
 self.author\_entry = tk.Entry(self.search\_frame)  
 self.author\_entry.grid(row=0, column=1, padx=5)  
  
 tk.Label(self.search\_frame, text="Наименование:").grid(row=0, column=2, padx=5)  
 self.title\_entry = tk.Entry(self.search\_frame)  
 self.title\_entry.grid(row=0, column=3, padx=5)  
  
 tk.Button(self.search\_frame, text="Поиск", command=self.search\_books).grid(row=0, column=4, padx=5)  
  
 self.books\_listbox = Listbox(self.root, width=80, height=15)  
 self.books\_listbox.pack(pady=10)  
 self.books\_listbox.bind("<Double-1>", self.read\_book)  
  
 self.add\_button = tk.Button(self.root, text="Добавить книгу", command=self.add\_book)  
 self.add\_button.pack(pady=5)  
  
 def search\_books(self):  
 author = self.author\_entry.get().lower()  
 title = self.title\_entry.get().lower()  
  
 self.books\_listbox.delete(0, tk.END)  
  
 for book in self.library:  
 if (author in book['author'].lower() or not author) and (title in book['title'].lower() or not title):  
 self.books\_listbox.insert(tk.END, f"{book['title']} — {book['author']} ({book['path']})")  
  
 def add\_book(self):  
 file\_path = filedialog.askopenfilename(  
 title="Выберите книгу",  
 filetypes=(("Все поддерживаемые", "\*.pdf;\*.epub;\*.fb2"), ("PDF файлы", "\*.pdf"), ("EPUB файлы", "\*.epub"),  
 ("FB2 файлы", "\*.fb2"))  
 )  
 if not file\_path:  
 return  
  
 title = os.path.basename(file\_path)  
 author = simpledialog.askstring("Автор", "Введите автора книги:")  
  
 if not author:  
 messagebox.showerror("Ошибка", "Автор обязателен!")  
 return  
  
 self.library.append({"title": title, "author": author, "path": file\_path})  
 messagebox.showinfo("Успех", f"Книга '{title}' добавлена!")  
 self.search\_books()  
  
 def read\_book(self, event):  
 selection = self.books\_listbox.curselection()  
 if not selection:  
 return  
  
 book\_index = selection[0]  
 selected\_book = self.library[book\_index]  
 file\_path = selected\_book['path']  
  
 if file\_path.endswith('.pdf'):  
 self.read\_pdf(file\_path)  
 elif file\_path.endswith('.epub'):  
 self.read\_epub(file\_path)  
 else:  
 messagebox.showerror("Ошибка", "Формат не поддерживается.")  
  
 def read\_pdf(self, file\_path):  
 doc = fitz.open(file\_path)  
 text = ""  
 for page in doc:  
 text += page.get\_text()  
 doc.close()  
  
 self.show\_text\_window(text)  
  
 def read\_epub(self, file\_path):  
 book = epub.read\_epub(file\_path)  
 text = ""  
 for item in book.get\_items():  
 if item.get\_type() == epub.EpubHtml:  
 text += item.get\_content().decode('utf-8')  
  
 self.show\_text\_window(text)  
  
 def show\_text\_window(self, text):  
 text\_window = tk.Toplevel(self.root)  
 text\_window.title("Чтение книги")  
 text\_window.geometry("800x600")  
  
 text\_widget = tk.Text(text\_window, wrap=tk.WORD)  
 text\_widget.pack(expand=True, fill=tk.BOTH)  
 text\_widget.insert(tk.END, text)  
 text\_widget.config(state=tk.DISABLED)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 root = tk.Tk()  
 app = LibraryApp(root)  
 root.mainloop()*

Результат работы программы:

