

Инструкция для написания программы "Просмотрщик файлов"

1. Импорт библиотек Tkinter

```
python Copy code  
  
import tkinter as tk  
from tkinter import filedialog
```

Этот код импортирует библиотеку Tkinter, которая позволяет создавать графические интерфейсы, и модуль filedialog для работы с диалогами выбора файлов.

2. Создание класса приложения

```
python Copy code  
  
class FileViewerApp:  
    def __init__(self, root):  
        # Основные настройки окна  
        self.root = root  
        self.root.title("File Viewer App")  
        self.root.geometry("800x600")
```

Этот код создает класс FileViewerApp, который будет содержать все элементы нашего приложения.

3. Создание элементов интерфейса

```
python Copy code  
  
# Создание кнопки "Сохранить файл"  
self.save_button = tk.Button(root, text="Сохранить файл", command=  
self.save_button.pack(side=tk.TOP, padx=10, pady=10)  
  
# Создание кнопки "Открыть файл"  
self.open_button = tk.Button(root, text="Открыть файл", command=  
self.open_button.pack(side=tk.TOP, padx=10, pady=10)  
  
# Создание текстового поля  
self.text_area = tk.Text(root, wrap=tk.WORD, width=50, height=2  
self.text_area.pack(side=tk.RIGHT, padx=10, pady=10)  
  
# Создание кнопки "Выход"  
self.exit_button = tk.Button(root, text="Выход", command=root.de  
self.exit_button.pack(side=tk.BOTTOM, pady=10)
```

Этот код создает кнопки "Сохранить файл", "Открыть файл", текстовое поле и кнопку "Выход". Каждая кнопка связана с соответствующей функцией.

4. Функция сохранения файла

```
python Copy code

def save_file(self):
    file_path = filedialog.asksaveasfilename(defaultextension=".txt")
    if file_path:
        with open(file_path, 'w', encoding='utf-8') as file:
            file_content = self.text_area.get(1.0, tk.END)
            file.write(file_content)
```

Эта функция открывает диалоговое окно сохранения файла, получает путь к файлу, и если путь выбран, сохраняет содержимое текстового поля в выбранный файл.

5. Функция открытия файла

```
python Copy code

def open_file(self):
    file_path = filedialog.askopenfilename()
    if file_path:
        with open(file_path, 'r', encoding='utf-8') as file:
            file_content = file.read()
            self.text_area.delete(1.0, tk.END)
            self.text_area.insert(tk.END, file_content)
```

Эта функция открывает диалоговое окно выбора файла, получает путь к выбранному файлу и отображает его содержимое в текстовом поле.

6. Запуск приложения

```
python Copy code

if __name__ == "__main__":
    root = tk.Tk()
    app = FileViewerApp(root)
    root.mainloop()
```

Этот код создает корневое окно Tkinter, создает объект FileViewerApp и запускает цикл событий для отображения приложения.

Дополнительные задания для самостоятельной работы

1. **Добавьте кнопку "Очистить текст":** Создайте новую функцию, которая будет очищать текстовое поле, и добавьте кнопку "Очистить текст" в интерфейс.
2. **Добавьте цветовую схему:** Попробуйте изменить цвета кнопок и фона приложения. Используйте атрибуты ``bg`` (background) и ``fg`` (foreground) для установки цвета фона и текста соответственно.
3. **Добавьте иконки на кнопки:** Попробуйте добавить маленькие изображения (иконки) на кнопки. Вы можете использовать изображения формата GIF или PNG.
4. **Улучшите дизайн:** Экспериментируйте с размещением элементов и их размерами, чтобы сделать приложение более красивым и удобным в использовании.