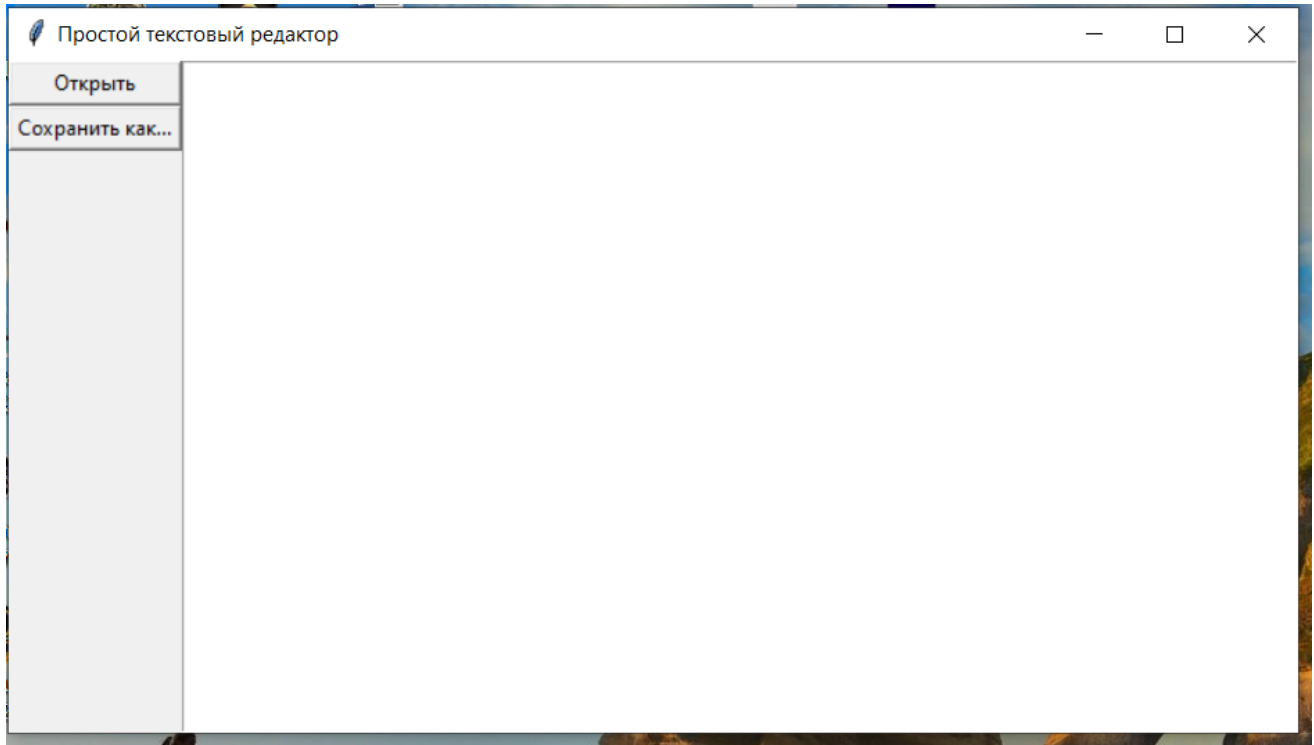


Задача:

Разработать простой текстовый редактор с возможностью сохранять и открывать текстовые файлы

Внешний вид редактора:



Ход работы:

1. Импортируйте библиотеку для работы с графическим интерфейсом
2. Создайте базовый объект окна Tk(), пропишите ему название:

```
root = Tk()
root.title("Простой текстовый редактор")
```

3. Добавьте виджет Frame() для хранения кнопок, и добавьте на него сами кнопки Button():

```
fr_buttons = Frame(root)
btn_open = Button(fr_buttons, text="Открыть")
btn_save = Button(fr_buttons, text="Сохранить как...")

btn_open.grid(row=0, column=0, sticky="ew")
btn_save.grid(row=1, column=0, sticky="ew")

fr_buttons.grid(row=0, column=0, sticky="ns")
```

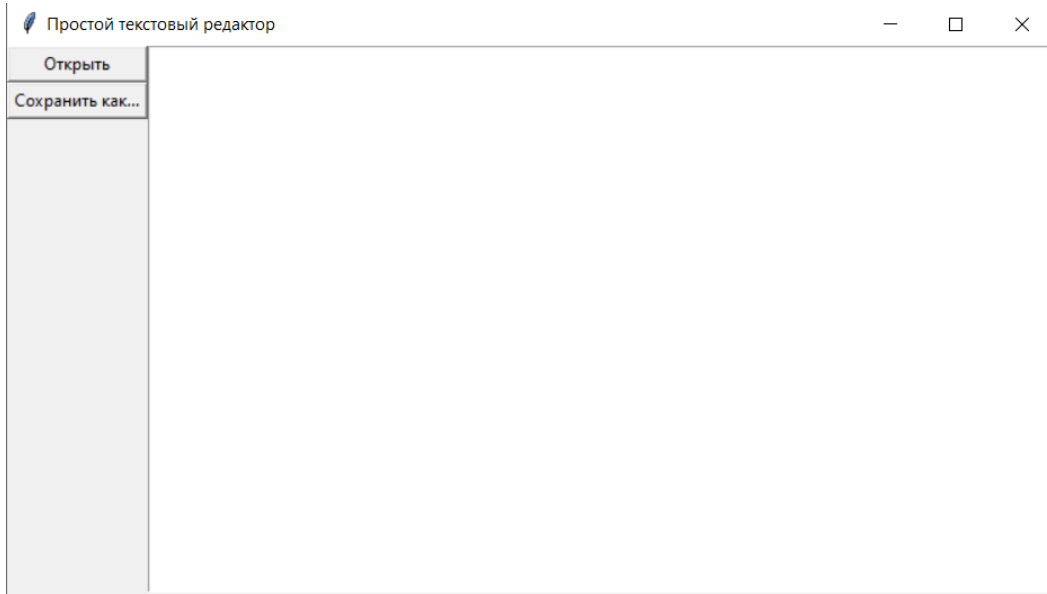
4. Добавьте виджет для редактирования текста Text():

```
txt_edit = Text(root)
txt_edit.grid(row=0, column=1, sticky="nsew")
```

5. Запустите цикл окна функцией mainloop()

```
root.mainloop()
```

6. Должно получиться окно следующего вида:



Если сейчас попытаться менять размеры окна, будет видно, что размеры элементов остаются прежними, чтобы это исправить, допишем команды:

```
root.title("Простой текстовый редактор")
root.rowconfigure(0, weight=1)
root.columnconfigure(1, weight=1)
```

Первый аргумент указывает на какой столбец/колонку применяется конфигурация, второй weight равный 1 говорит, что размеры элементов будет меняться вместе с размерами окна.

7. Следующее что нужно добавить это функции сохранения и открытия файлов:

```
def open_file(filepath):
    with open(filepath, "r", encoding='utf-8') as input_file:
        return input_file.read()
```

```
def save_file(filepath, text):
    with open(filepath, "w", encoding='utf-8') as output_file:
        output_file.write(text)
```

8. Для того чтобы открывать и сохранять файлы с помощью графического интерфейса можно использовать соответствующие диалоговые окна, сперва их нужно импортировать:

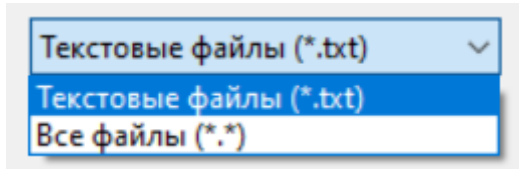
```
2 from tkinter.filedialog import askopenfilename, asksaveasfilename
```

9. Теперь пропишем функции для обработки событий нажатия кнопок и вызовем в них диалоговые окна:

```
def click_open_file():
    filepath = askopenfilename(
        filetypes=[("Текстовые файлы", "*.txt"), ("Все файлы", "*.*")]
    )
    if not filepath:
        return

def click_save_file():
    filepath = asksaveasfilename(
        defaultextension="txt",
        filetypes=[("Текстовые файлы", "*.txt"), ("Все файлы", "*.*")],
    )
    if not filepath:
        return
```

askopenfilename и asksaveasfilename, вызывают диалоговые окна и возвращают путь к выбранному файлу, аргумент filetypes принимает список кортежей в которых указано какие типы файлов будут отображаться в диалоговом окне:



10. Для того чтобы кнопки реагировали на нажатие, пропишем в них какую функцию они должны вызывать при нажатии:

```
btn_open = Button(fr_buttons, text="Открыть", command=click_open_file)
btn_save = Button(fr_buttons, text="Сохранить как...", command=click_save_file)
```

11. Сейчас функции ничего не делают, допишем в них функционал:

```
def click_open_file():
    filepath = askopenfilename(
        filetypes=[("Текстовые файлы", "*.txt"), ("Все файлы", "*.*")]
    )
    if not filepath:
        return

    text = open_file(filepath)

    txt_edit.delete("1.0", END)
    txt_edit.insert(END, text)
    root.title(f"Простой текстовый редактор - {filepath}")
```

```
def click_save_file():
    filepath = asksaveasfilename(
        defaultextension="txt",
        filetypes=[("Текстовые файлы", "*.txt"), ("Все файлы", "*.*")],
    )
    if not filepath:
        return

    text = txt_edit.get("1.0", END)
    save_file(filepath, text)
    root.title(f"Простой текстовый редактор - {filepath}")
```

Задания для самостоятельного выполнения:

1. Добавить функцию «Сохранить», которая будет сохранять файл без вызова диалогового окна, если путь к нему уже известен, в противном случае отработает аналогично кнопке «Сохранить как»
2. Перенести функционал кнопок в выпадающее меню