

Задание:

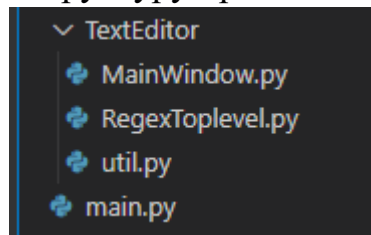
Улучшить структуру кода текстового редактора, применить принципы ООП при написании GUI.

Добавить функционал регулярных выражений к текстовому редактору.

Познакомиться с созданием дополнительных окон

Ход работы:

1. Для начала создадим новую структуру проекта:



В основной папке проекта создадим файл main.py и папку TextEditor, в которой создадим файлы MainWindow.py, RegexToplevel.py и util.py

2. Сперва перенесём наши функции open_file и save_file в файл util.py:

```
TextEditor > util.py > save_file
1  def save_file(filepath, text):
2      with open(filepath, "w", encoding='utf-8') as output_file:
3          output_file.write(text)
4
5  def open_file(filepath):
6      with open(filepath, "r", encoding='utf-8') as input_file:
7          return input_file.read()
```

3. Затем перепишем уже имеющийся код текстового редактора в ООП стиле, в файле MainWindow напишем следующий код:

```

1  from tkinter import *
2  from tkinter.filedialog import askopenfilename, asksaveasfilename
3
4  from TextEditor.util import *
5
6  class MainWindow(Tk):
7      def __init__(self):
8          super().__init__()
9          self.title("Простой текстовый редактор")
10         self.rowconfigure(0, weight=1)
11         self.columnconfigure(1, weight=1)
12
13         self.fr_buttons = Frame(self)
14
15         self.btn_open = Button(self.fr_buttons, text="Открыть", command=self._click_open_file)
16         self.btn_open.grid(row=0, column=0, sticky="ew")
17
18         self.btn_save = Button(self.fr_buttons, text="Сохранить как...", command=self._click_save_file)
19         self.btn_save.grid(row=1, column=0, sticky="ew")
20
21         self.fr_buttons.grid(row=0, column=0, sticky="ns")
22
23         self.txt_edit = Text(self, undo=True)
24         self.txt_edit.grid(row=0, column=1, sticky="nsew")
25
26         self.mainloop()
27
28     def _click_open_file(self):
29         filepath = askopenfilename(
30             | filetypes=[("Текстовые файлы", "*.txt"), ("Все файлы", "*.*")]
31         )
32         if not filepath:
33             return
34
35         text = open_file(filepath)
36
37         self.txt_edit.delete("1.0", END)
38         self.txt_edit.insert(END, text)
39         self.title(f"Простой текстовый редактор - {filepath}")
40
41
42     def _click_save_file(self):
43         filepath = asksaveasfilename(
44             | defaultextension="txt",
45             | filetypes=[("Текстовые файлы", "*.txt"), ("Все файлы", "*.*")],
46         )
47         if not filepath:
48             return
49
50         text = self.txt_edit.get("1.0", END)
51         save_file(filepath, text)
52         self.title(f"Простой текстовый редактор - {filepath}")
53

```

В данном случае вместо создания отдельного экземпляра класса Tk мы создаём собственный класс, в данном случае MainWindow, и наследуем его от класса Tk, таким образом наш класс будет дополнять стандартный класс своим функционалом. Внутри конструктора __init__ мы, во-первых, вызываем инициализацию родителя, и затем начинаем описывать структуру нашего окна, вы можете заметить, что код идентичен тому что мы писали ранее, за тем исключением, что виджеты добавляются как поля нашего класса, через оператор «self», который так же передаётся как корневой виджет при создании новых.

В данном примере приведён базовый код текстового редактора без улучшений что вы добавляли в первом задании, перепишите свой вариант решения в виде класса, с добавлением кнопки сохранить и переносом функций в верхнее меню.

4. Теперь создадим класс дополнительного окна, в файле `RegexToplevel` напишем следующий код:

```
1  from tkinter import *
2  import re
3
4  class RegexToplevel(Toplevel):
5      def __init__(self, root):
6          super().__init__(root)
7          self.root = root
8          self.title('Дополнительное окно')
9          self.columnconfigure(0, weight=1)
10         self.rowconfigure(0, weight=1)
11         self.resizable(False, False)
12
13         self.fr_functions = Frame(self)
14         self.fr_functions.columnconfigure(0, weight=1)
15         self.fr_functions.rowconfigure(2, weight=1)
16
17         self.ent_regex = Entry(self.fr_functions)
18         self.ent_regex.grid(column=0, row=0, sticky=EW, padx=2, pady=2)
19
20         self.btn_findall = Button(self.fr_functions, text='Найти', command=self._click_findall)
21         self.btn_findall.grid(column=0, row=1, sticky=EW, padx=2, pady=2)
22
23         self.list_result = Listbox(self.fr_functions, height=10)
24         self.list_result.grid(column=1, row=0, rowspan=2, sticky=NSEW, padx=2, pady=2)
25
26         self.fr_functions.grid(column=0, row=0, sticky=NS)
27
28     def _click_findall(self):
29         self.list_result.delete(0, END)
30         pattern = re.compile(self.ent_regex.get())
31         result = pattern.findall(self.root.txt_edit.get('1.0', END))
32         for i in result:
33             self.list_result.insert(END, i)
```

Дополнительные окна создаются, схожим образом с основным разница лишь в том что вместо класса `Tk` мы используем `Toplevel`, и отсутствует функция `mainloop`, открытие окна будет производиться в главном окне. Обратите внимание что для работы с элементами главного окна, в конструкторе мы сохраняем, корневое окно в отдельное поле (`root`).

5. Для того чтобы открыть дополнительное окно дополним код в файле `MainWindow`.

Импортируем класс из модуля с новым окном:

```
4  from TextEditor.RegexToplevel import RegexToplevel
5  from TextEditor.util import *
```

Добавим кнопку по которой будет вызываться новое окно:

```
self.btn_open_re_tl = Button(self.fr_buttons, text='Анализатор', command=self._click_open_re_tl)
self.btn_open_re_tl.grid(row=2, column=0, sticky='ew')
```

Напишем функцию в которой и будем вызывать новое окно:

```
58     def _click_open_re_tl(self):
59         tl_reg = RegexToplevel(self)
60         #tl_reg.grab_set()
61
```

Функцией `grab_set` мы можем заблокировать основное окно пока открыто основное, попробуйте оба варианта (с вызовом функции и без) и решите какой подходит лучше в данной ситуации.

Задания:

1. Напишите новое приложение с учётом прошлого опыта, кнопки уберите и создайте верхнее меню, команды для сохранения и открытия файла уберите во вкладку «Файл», для команды «Анализатор», создайте вкладку «Правка». Окно для работы с регулярными выражениями также дополните функционалом с прошлого урока, поэкспериментируйте с дизайном, выберите наиболее подходящий для вас.
2. Добавьте новую команду «О нас» в верхнее меню, опишите для неё новый `Toplevel` в котором опишите информацию о себе: ФИО, № группы, e-mail и т.д. Самостоятельно изучите как можно вставить изображение в `tkinter`, и добавьте в окно «О нас» эмблему на свой выбор.