|  |  |
| --- | --- |
| CLI | UI с графической библиотекой |
| from mod.txt\_processor import process\_txt\_file from mod.xml\_processor import process\_xml\_file from mod.json\_processor import process\_json\_file from mod.yaml\_processor import process\_yaml\_file import os  def main():  #print("Текущая рабочая директория:", os.getcwd())  file\_type = input("Введите тип входного файла (txt/xml/json/yaml): ").strip().lower()  print("Выберите метод обработки:")  print("1. Регулярные выражения")  print("2. Парсинг")  print("3. Математическая библиотека (eval)")  method = int(input("Введите номер метода (1, 2 или 3): "))   if file\_type == 'txt':  input\_file = 'data/input.txt'  output\_file = 'data/output.txt'  process\_txt\_file(input\_file, output\_file, method)  elif file\_type == 'xml':  input\_file = 'data/input.xml'  output\_file = 'data/output.xml'  process\_xml\_file(input\_file, output\_file, method)  elif file\_type == 'json':  input\_file = 'data/input.json'  output\_file = 'data/output.json'  process\_json\_file(input\_file, output\_file, method)  elif file\_type == 'yaml':  input\_file = 'data/input.yaml'  output\_file = 'data/output.yaml'  process\_yaml\_file(input\_file, output\_file, method)  else:  print("Ошибка: неподдерживаемый тип файла. Пожалуйста, используйте 'txt', 'xml', 'json', 'yaml'.")  if \_\_name\_\_ == "\_\_main\_\_":  main() | from tkinter import \* from tkinter import messagebox from mod.txt\_processor import process\_txt\_file from mod.xml\_processor import process\_xml\_file from mod.json\_processor import process\_json\_file from mod.yaml\_processor import process\_yaml\_file import os  class FileProcessorApp:  def \_\_init\_\_(self, root):  self.root = root  self.root.title("Обработка файлов")  self.root.configure(bg="#FDF4E3")   self.label\_file\_type = Label(root, text="Введите тип входного файла (txt/xml/json/yaml):",  font=("Calibri", 12, "bold"), fg="#503D33", bg="#FDF4E3")  self.label\_file\_type.pack(pady=10)   self.entry\_file\_type = Entry(root, width=20, font=("Calibri", 12))  self.entry\_file\_type.pack(pady=5)   self.label\_method = Label(root, text="Выберите метод обработки:",  font=("Calibri", 12, "bold"), bg="#FDF4E3", fg="#503D33")  self.label\_method.pack(pady=10)   self.method\_var = IntVar()  self.method\_var.set(1)   self.radio1 = Radiobutton(root, text="1. Регулярные выражения", variable=self.method\_var, value=1,  font=("Calibri", 12), indicatoron=0, width=30, relief=FLAT, bg="#F5E6CB",  activebackground="#E7C697", fg="#503D33")  self.radio1.pack(pady=5)   self.radio2 = Radiobutton(root, text="2. Парсинг", variable=self.method\_var, value=2, font=("Calibri", 12),  indicatoron=0, width=30, relief=FLAT, bg="#F5E6CB", activebackground="#E7C697",  fg="#503D33")  self.radio2.pack(pady=5)   self.radio3 = Radiobutton(root, text="3. Математическая библиотека (eval)", variable=self.method\_var,  value=3, font=("Calibri", 12), indicatoron=0, width=40, relief=FLAT, bg="#F5E6CB",  activebackground="#E7C697", fg="#503D33")  self.radio3.pack(pady=5)   self.process\_button = Button(root, text="Обработать файл", command=self.process\_file, font=("Calibri", 12),  bg="#8A6642", fg="#FDF4E3", activebackground="#6DAE81", borderwidth=0,  relief=FLAT)  self.process\_button.pack(pady=20)   def process\_file(self):  file\_type = self.entry\_file\_type.get().strip().lower()  method = self.method\_var.get()   input\_file = f'data/input.{file\_type}'  output\_file = f'data/output.{file\_type}'   try:  if file\_type == 'txt':  process\_txt\_file(input\_file, output\_file, method)  elif file\_type == 'xml':  process\_xml\_file(input\_file, output\_file, method)  elif file\_type == 'json':  process\_json\_file(input\_file, output\_file, method)  elif file\_type == 'yaml':  process\_yaml\_file(input\_file, output\_file, method)  else:  raise ValueError("Неподдерживаемый тип файла.")   messagebox.showinfo("Успех", f"Файл успешно обработан и сохранён в {output\_file}.")  except Exception as e:  messagebox.showerror("Ошибка", str(e))   if \_\_name\_\_ == "\_\_main\_\_":  root = Tk()  root.geometry("400x350+550+300")  root.resizable(False, False)  app = FileProcessorApp(root)  root.mainloop() |
| Пользователь взаимодействует с приложением через текстовый интерфейс в терминале. | Пользователь взаимодействует через графические элементы (кнопки, радиокнопки, текстовые поля) |
| Простой и быстрый способ взаимодействия, однако требует знаний командного ввода. | Более сложен в реализации, но предоставляет лучший пользовательский опыт. |
| Ввод данных осуществляется через функцию input(), где пользователь вводит тип файла и метод обработки. Обработка ошибок осуществляется через простые сообщения в терминале. | Ввод данных происходит через текстовые поля и радиокнопки. Обработка ошибок представлена в виде всплывающих окон, что делает их более заметными для пользователя. |
| **CLI** подходит для тех, кто предпочитает быстроту и простоту, а также для сценариев, где автоматизация через скрипты важна. | **GUI** предоставляет более дружелюбный интерфейс, удобен для пользователей с разным уровнем подготовки и может улучшить взаимодействие с приложением. |