МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное учреждение высшего образования

**«Московский технический университет связи и информатики»**

Лабораторная работа №3

**Работа с файлами в Python: открытие, чтение, запись, работа с исключениями**

**по дисциплине**

**«Введение в информационные технологии»**

Выполнил: студент гр. БВТ2403

Лашков Ю.Е.

Проверил: Кузнецов В.А.

Москва, 2024 г.

**Цель работы**: Освоить принципы определения и использования функций в языке программирования Python, понять механизмы передачи аргументов в функции, научиться применять функции для решения практических задач, а также изучить базовые алгоритмические конструкции.

**Оборудование:** Компьютер с установленным программным обеспечением Python.

**Введение:** Функции в Python являются важным инструментом для организации и оптимизации кода. Они упрощают чтение и повторное использование программ, разбивая сложные задачи на более управляемые части. Чтение и запись файлов — ключевые операции, позволяющие работать с внешними данными. Используя функции, мы можем гибко управлять процессом взаимодействия с файлами.

**Методы и технологии:** В рамках разработки использовались такие функции, как:

- Определение с помощью **def().**

- Использование **return** для возврата значений.

Такие конструкции, как:

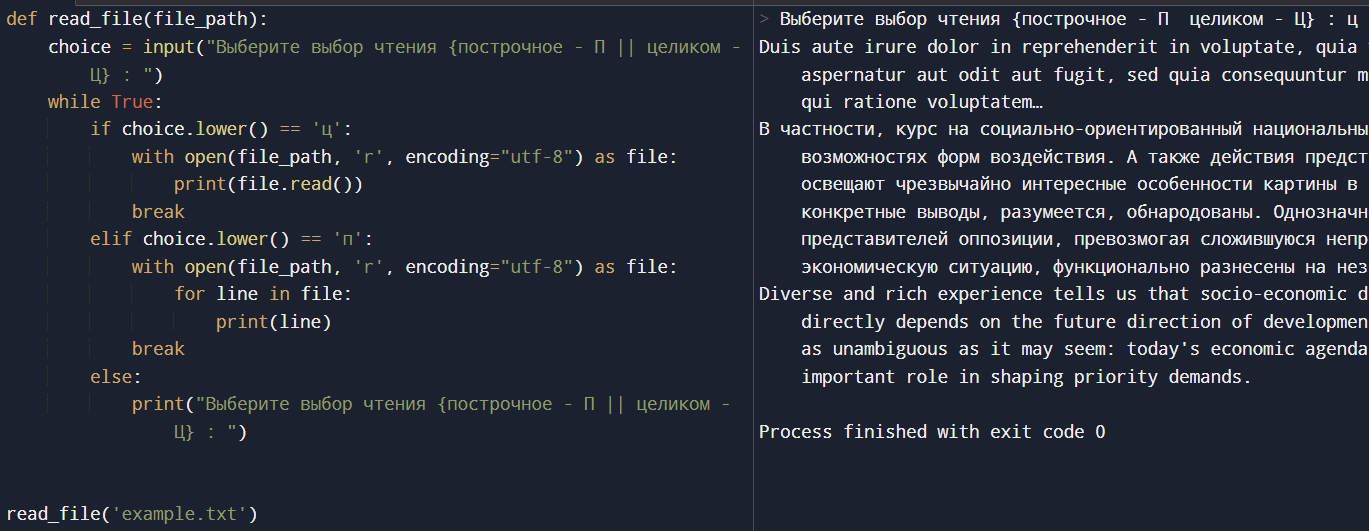
- Условные операторы: **if, elif, else** для обработки различных условий.

- Циклы: for, while для реализации повторяющихся операций.

- Обработка исключений с помощью **try-except**, что позволяет избежать сбоев при работе с файлами.

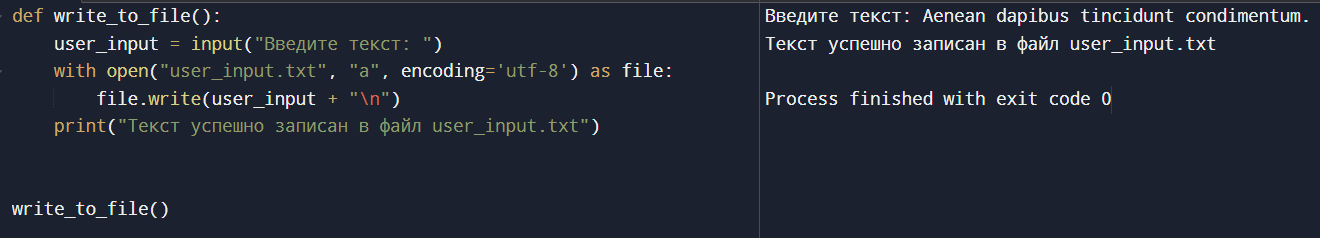
- Использование конструкции **with open as file**, что обеспечивает автоматическое закрытие файла и улучшает управление ресурсами.

***Задача №1*** *(построчный или цельный выбор чтения)*



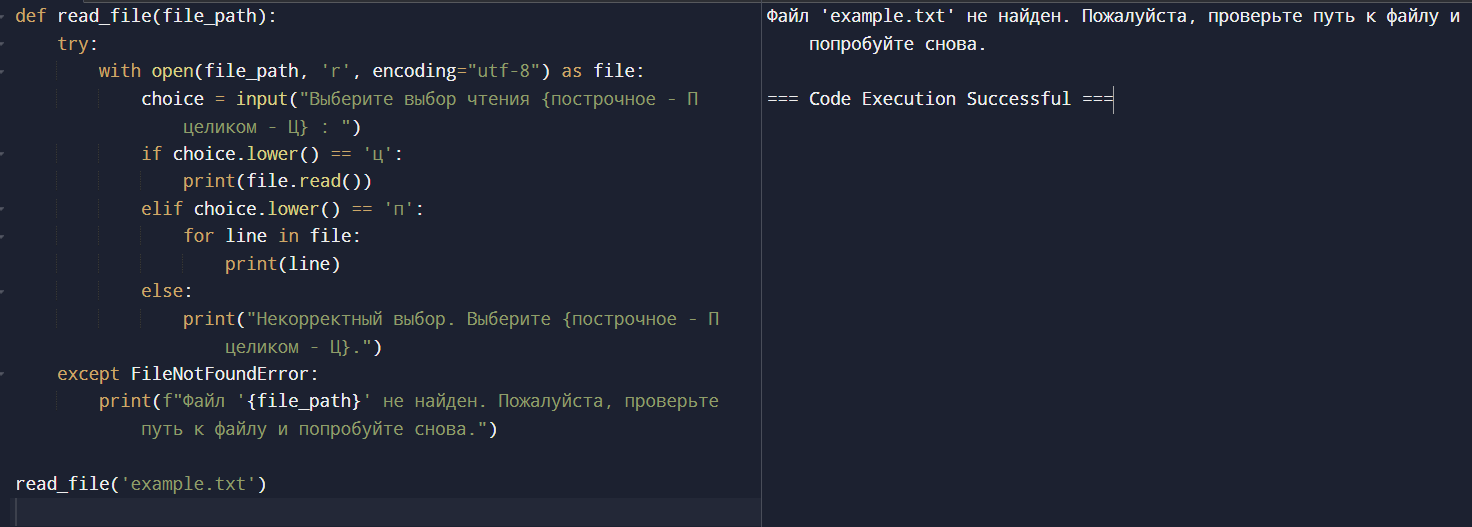
Программа реализует функцию **read\_file**, которая позволяет пользователю выбрать способ чтения содержимого файла. Пользователь вводит свой выбор: построчное чтение или чтение целиком. В зависимости от выбора открывается файл по указанному пути, и его содержимое выводится на экран. Если введено некорректное значение, программа запрашивает выбор снова. Функция завершает выполнение после успешного чтения файла.

***Задача №2*** *(добавление текста в существующий файл)*



Программа реализует функцию **write\_to\_file**, которая позволяет пользователю вводить текст для записи в файл. При запуске функции пользователю предлагается ввести текст, который затем добавляется в файл с именем "**user\_input.txt**". Запись осуществляется в режиме добавления, новый текст будет помещен в конец файла. После успешной записи программа выводит сообщение о том, что текст успешно сохранен.

***Задача №3*** *(модификация программы №1)*



Программа также реализует функцию **read\_file**, но с добавленной обработкой исключений. Она позволяет пользователю выбрать способ чтения содержимого файла, как и в первом коде. Если пользователь выбирает чтение целиком, функция выводит все содержимое файла, а если выбирает построчное чтение, файл выводится построчно. Если введено некорректное значение, программа сообщает об этом. В дополнение, если файл не найден, функция обрабатывает исключение **FileNotFoundError** и выводит сообщение о том, что файл отсутствует.

**Заключение:**

В ходе работы мы освоили основные принципы определения и использования функций в Python, а также методы работы с файлами. Мы рассмотрели, как считывать и записывать данные, используя функции, что позволяет удобно взаимодействовать с внешними источниками информации.