МИНЕСТЕРСТВО 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. 1БВТ2403

Королев.А.А

Проверил:

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:** Работа с файлами в Python позволяет обрабатывать и анализировать большие объемы данных, проводить операции с файловой системой, создавать и модифицировать файлы различных форматов. Таким образом, работа с файлами в языке Python может быть полезной и важным навыком для специалистов в различных областях деятельности.

**Цели и задачи:**

Основной целью данной работы является разработка приложений, которое будет открывать заданный файл в режиме чтения и будет выводить его содержимое на экран. Разработка программы для записи пользовательского текста в файл. Разработка программы для обработки исключений.

1.Освоить базовые функции работы с файлами.

2.Обеспечить интерактивный ввод для работы с файлами пользователю.

3.Устранить ошибки путём обработки исключений (try-except)

**Методы и технологии:**

Для реализации приложения использовался язык программирования Python, который славится своей простотой и мощными библиотеками для работы с данными. Основные функции: def — **это** фрагмент кода, который выполняет определённые операции по инструкции из тела функции и выдаёт результат.

Чтобы открыть файл, воспользуемся конструкцией «with as» и функцией «open» - передадим в неё название файла и параметр открытия ('a' - append добавить в файл, 'r' – read чтение файла). С помощью блока file.read() можно прочитать файл полностью, а с помощью цикла for line in file выведем содержимое файла построчно.

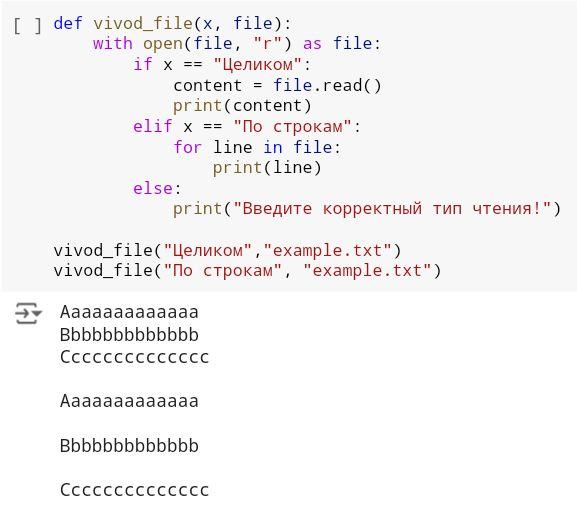
С помощью функции input пользователь будет вводить формат чтения файла для вывода в задаче 1 и 3. В задаче 2 с помощью input пользователь будет вводить свой текст, который с помощью блока file.write() добавляется в файл user\_uinput.txt.

**Реализация:**

1. **Программа для открытия и чтения содержимого файла:**

В первом выводе не был указан корректный тип чтения файла, поэтому программа запрашивает новый ввод от пользователя.

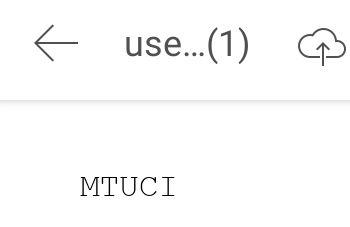
Во втором выводе был передан тип чтения «Целиком», а в третьем – «По строкам».



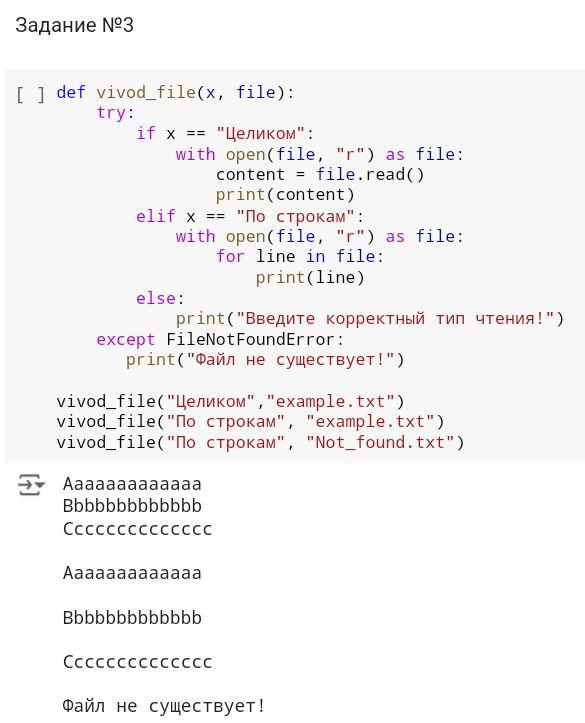
1. Программа для записи пользовательского текста в файл:

В качестве пользовательского текста было передано «MTUCI».



В итоге данный текст был записан в файл user\_input.txt.

3.Приложение для обработки исключений:



В качестве аргумента в функцию был передан несуществующий файл. Результат – вывод сообщения «Файл не существует!»

**Вывод:** данная лабораторная работа продемонстрировала возможности языка Python в разработке приложений для работы с базовыми алгоритмическими конструкциями и обработки файлов.