

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ,  
СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение  
высшего образования**

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

**Факультет «Информационные технологии»**

**Кафедра «Математическая кибернетика и информационные технологии»**

**Дисциплина: «Введение в информационные технологии»**

**Лабораторная работа №3**

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

**Выполнил:**

**Студент группы БВТ2402**

**Чимитов Намжил**

**Москва**

**2024**

## Цель работы

Освоить работу с файлами в Python, научиться открывать, читать файлы, записывать в них информацию, а также обрабатывать исключения.

## Задачи

### 1. Открытие и чтение файла.

- Создать текстовый файл *example.txt* и заполнить его несколькими строками текста.
- Написать функцию на Python, которая открывает файл *example.txt* в режиме чтения и выводит его содержимое на экран.
- Использовать разные методы чтения файла: чтение всего файла сразу, построчное чтение, реализовать выбор типа чтения в принимаемых аргументах функции.

### 2. Запись в файл.

- Написать программу, которая запрашивает у пользователя текст и записывает его в новый файл *user\_input.txt*.
- Реализовать функционал добавления текста в существующий файл, не удаляя его предыдущее содержимое.

### 3. Запись в файл с обработкой исключения.

- Модифицировать программу из Задачи 1 так, чтобы она корректно обрабатывала исключение, возникающее при попытке открыть несуществующий файл.
- Вместо вывода ошибки программа должна выводить пользователю понятное сообщение.
- Использовать в блоке *try except* следующий класс исключений: *FileNotFoundError*.

## Ход работы

**Задача 1.** Создан файл *example.txt*. Написана функция, которая открывает файл *example.txt* в режиме чтения и выводит его содержимое на экран. Реализованы разные методы чтения файла: чтение всего файла сразу и построчное чтение.

```
def file_viewer(name='', style='all'):  
    with open(name, 'r') as f:  
        if style == 'all':  
            print(f.read())  
        elif style == 'str':  
            for line in f:  
                print(line, end='')  
        else:  
            print('(!) Неизвестный метод чтения :(')  
  
name = input('Введите название файла с расширением: ')  
style = input('Введите метод чтения: all - f.read() / str-by-str -  
for loop: ')  
file_viewer(name, style)
```

**Задача 2.** Написана программа, которая запрашивает у пользователя текст и записывает его в новый файл *user\_input.txt*. Реализован функционал добавления текста в существующий файл без удаления его предыдущего содержимого.

```
def writer(text=''):  
    with open('user_input.txt', 'w') as f:  
        f.write(text)  
  
def editor(name, text=''):  
    with open(name, 'a') as f:  
        f.write('\n')  
        f.write(text)  
  
choice = int(input('Новый файл (1) или изменить существующий (2)? ---  
'))  
if choice == 1:  
    writer(input('Введите строку ниже:\n'))  
if choice == 2:  
    editor(input('Введите название файла: '), input('Введите строку  
ниже:\n'))  
else:  
    exit()
```

**Задача 3.** Модифицирован код из Задачи 1 – теперь он корректно обрабатывает исключение, возникающее при попытке открыть несуществующий файл. Вместо вывода ошибки код выводит пользователю понятное сообщение. В блоке *try except* использован класс исключений *FileNotFoundError*.

```
def file_viewer(name='', style='all'):
    try:
        with open(name, 'r') as f:
            if style == 'all':
                print(f.read())
            elif style == 'str':
                for line in f:
                    print(line, end='')
            else:
                print('(!) Неизвестный метод чтения :(')
    except FileNotFoundError:
        print('(!) Файл с таким названием не найден :(')

name = input('Введите название файла с расширением: ')
style = input('Введите метод чтения: all - f.read() / str - for loop: ')
file_viewer(name, style)
```

## Вывод

В результате выполнения лабораторной работы была освоена работа с файлами в Python, открытие и чтение файлов, запись в них информацию, а также обработка исключений.