

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ И МАССОВЫХ  
КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени  
Федеральное государственное бюджетное общеобразовательное  
учреждение  
высшего образования**

**“Московский технический университет связи и информатики”  
(МТУСИ)**

**Кафедра “Программная инженерия”**

**Лабораторная работа №3  
по дисциплине “Введение в информационные технологии”**

**Выполнил: Студент группы  
БПИ2503  
Яричевский Даниил**

**Москва  
2025**

## Цель работы

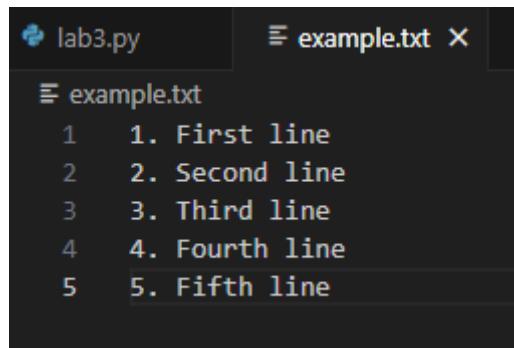
Освоить принципы определения и использования функций в языке программирования Python, понять механизмы передачи аргументов в функции, научиться применять функции для решения практических задач, а также изучить базовые алгоритмические конструкции.

## Задачи

1. Открытие и чтение файла
2. Запись в файл
3. Запись в файл

## Ход работы

- 1.1 Создаем файл example.txt и заполняем его несколькими строками текста.



```
lab3.py      example.txt x
example.txt
1. First line
2. Second line
3. Third line
4. Fourth line
5. Fifth line
```

Рис. (1.1)

- 1.2 Напишем функцию на Python, которая открывает файл example.txt в режиме чтения и выводит его содержимое на экран.

```
1 def read_file():
2     with open('example.txt', 'r') as file:
3         content = file.read()
4         print(content)
5 read_file()
```

Рис. (1.2)

1.3 Создадим функцию, использующую разные методы чтения файла: чтение всего файла сразу и построчное чтение.

Создадим переменную method, которая будет запрашивать ввод числа: 1 или 2. Добавим проверку на число: если method==1, то читаем весь файл, если method==2, то читаем файл построчно.

```
1 def read_file():
2     method = int(input("1-Полное чтение; 2-Построчное чтение\n Введите тип чтения: "))
3
4     if method == 1:
5         with open('example.txt', 'r') as file:
6             content = file.read()
7             print(content)
8
9     if method == 2:
10        with open('example.txt', 'r') as file:
11            for line in file:
12                print(line)
13
14 read_file()
```

Рис. (1.3)

2.1 Создаем функцию write\_file. С помощью режима w (write) создаем файл user\_input.txt и записываем в него текст.

```
1 def write_file():
2     with open('user_input.txt', 'w') as file:
3         file.write(input())
4
5 write_file()
```

Рис. (2.1)

2.2 По примеру из задания 1.3 создадим функцию с выбором типа записи файла с помощью переменной method. Для добавления текста в существующий файл используем режим a (append).

```
1 def write_file():
2     method = int(input("1-Запись с нуля; 2-Запись с конца\n Введите тип записи:"))
3
4     if method == 1:
5         with open('user_input.txt', 'w') as file:
6             file.write(input())
7     if method == 2:
8         with open('user_input.txt', 'a') as file:
9             file.write(input())
10    write_file()
```

Рис. (2.2)

3. Модифицируем программу из задания 1 так, чтобы она корректно обрабатывала исключение, возникающее при попытке открыть несуществующий файл.

```
lab 3.py > ...
1 def read_file():
2     method = int(input("1-Полное чтение; 2-Построчное чтение\n Введите тип чтения: "))
3
4     try:
5         if method == 1:
6             with open('example.txt', 'r') as file:
7                 content = file.read()
8                 print(content)
9
10        if method == 2:
11            with open('example.txt', 'r') as file:
12                for line in file:
13                    print(line)
14
15    except FileNotFoundError:
16        print("Ошибка: файл 'example.txt' не найден. Проверьте, что файл существует в той же папке.")
17
18    read_file()
19
```

Рис. (3)

## Вывод

Были освоены принципы определения и использования функций в языке программирования Python, поняты механизмы передачи аргументов в функции, научился применять функции для решения практических задач, а также изучать базовые алгоритмические конструкции.