

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

**Ордена Трудового Красного Знамени федеральное государственное бюджетное
образовательное учреждение высшего образования**

«Московский технический университет связи и информатики»

Кафедра Программная инженерия

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Введение в информационные технологии»

Тема: «Работа с файлами в Python: открытие, чтение, запись, работа с исключениями»

Выполнил: студент группы БВТ2504
Гапеев Алексей Денисович

Проверил: Харрасов К.Р.

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.

Индивидуальное задание

Задание 1:

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

Задание 2:

- 2.1. Написать программу, которая запрашивает у пользователя текст и записывает его в новый файл user_input.txt.
- 2.2. Реализовать функционал добавления текста в существующий файл, не удаляя его предыдущее содержимое.

Задание 3:

- 3.1. Модифицировать программу из Задания 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.

Скриншоты выполнения

Задание 1:

Выберите вариант чтения файла:

1. Чтение всего файла
2. Построчное чтение файла:

1

Выбран вариант 1.

```
d8c53dd55711eb6f614def4abc898cc3 /tmp/builtin-events-scheme7232674648889925659.json
d41d8cd98f00b204e9800998ecf8427e /tmp/gdm3-config-err-W4upf6
8ebaf94c5795c640f051881c5dc0f5ce /tmp/libcatboost4j-prediction4635869575473099183.so
```

Выбрано задание 1.

Выберите вариант чтения файла:

1. Чтение всего файла

2. Построчное чтение файла:

2

Выбран вариант 2.

d8c53dd55711eb6f614def4abc898cc3 /tmp/builtin-events-scheme7232674648889925659.json

d41d8cd98f00b204e9800998ecf8427e /tmp/gdm3-config-err-W4upf6

8ebaf94c5795c640f051881c5dc0f5ce /tmp/libcatboost4j-prediction4635869575473099183.so

Задание 2:

The screenshot shows the PyCharm interface. In the code editor, there is a file named 'Лабораторная 3.py' containing the following code:

```
asd;sldfgj;slkfdgj;sldfkgjHello, World!
```

In the 'Run' tab, the configuration 'Лабораторная 3' is selected. The terminal output shows:

```
/home/diaden/PycharmProjects/PythonProject1/.venv/bin/python /home/diaden/PycharmProjects/PythonProject1/Лабораторная 3.py
Выберите номер задания:
1. Открытие и чтение файла
2. Запись в файл
3. Открытие и чтение файла, часть 2
2
Выбрано задание 2.
Введите значение для записи в файл: Hello, World!
Process finished with exit code 0
```

Задание 3:

The screenshot shows the PyCharm interface. In the code editor, there is a file named 'Лабораторная 3.py' containing the following code:

```
# Задание 3. Запись в файл с блоком Try - Except
def open_file_updated():
    try:
        open_file()
    except FileNotFoundError:
        print("Файл не найден!")
```

In the 'Run' tab, the configuration 'Лабораторная 3' is selected. The terminal output shows:

```
/home/diaden/PycharmProjects/PythonProject1/.venv/bin/python /home/diaden/PycharmProjects/PythonProject1/Лабораторная 3.py
Выберите номер задания:
1. Открытие и чтение файла
2. Запись в файл
3. Открытие и чтение файла, часть 2
3
Выбрано задание 3.
Файл не найден!
Process finished with exit code 0
```

Исходный код программы

```
# Лабораторная работа №3. Работа с файлами в Python: открытие, чтение, запись, работа с исключениями

# Задание 1. Открытие и чтение файла
# Файл заполнен КС файлов каталога /etc/ssh/*
def open_file():
    with open('example.txt', 'r') as file:
        match int(input("Выберите вариант чтения файла:\n1. Чтение всего файла\n2. Построчное чтение файла:\n")):
            case 1:
                print("Выбран вариант 1.")
                content = file.read()
                print(content)
            case 2:
                print("Выбран вариант 2.")
                with open('example.txt', 'r') as file:
                    for line in file:
                        print(line)

# Задание 2. Запись в файл. Часть 1.

def user_input(user_content):
    with open ('user_input.txt', 'a') as file:      # Запись в конце файла без затирки прошлых данных
        file.write(user_content)
        #content = file.read()
        #?

# Задание 3. Запись в файл с блоком Try - Except

def open_file_updated():
    try:
        open_file()
    except FileNotFoundError:
        print("Файл не найден!")

# Основной блок кода
match int(input('Выберите номер задания:\n1. Открытие и чтение файла\n2. Запись в файл\n3. Открытие и чтение файла, часть 2\n')):
    case 1:
        print("Выбрано задание 1.")
        open_file()
    case 2:
        print("Выбрано задание 2.")
        user_input(input("Введите значение для записи в файл: "))
```

```
case 3:  
    print("Выбрано задание 3.")  
    open_file_updated()  
case _:  
    print("Введено неверное значение!!!")
```

Заключение

В ходе выполнения лабораторной работы были успешно решены следующие задачи:

1. Освоены принципы определения и использования функций в языке программирования Python, механизмы передачи аргументов в функции а также изучены базовые алгоритмические конструкции
2. Применены функции для решения практических задач.