

**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 “Математическая кибернетика и информационные технологии”

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 БВТ2505

Хардилов Владислав Дмитриевич

Проверил: Павликов. А.Е.

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

Задание 1:

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

Задание 2:

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

Задание 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`.

Исходный код программы:

```
import os

def edit_add_file(file_path, content):
    with open(file_path, "a", encoding="utf-8") as file:
        file.write(content + "\n")
    print(f"content added to {file_path}")

def choose_file_in_directory(directory):
    try:
        files = [f for f in os.listdir(directory) if os.path.isfile(os.path.join(directory, f))]
        if not files:
            print("\nno files found in the directory.")
            return None
        print("\navailable files:")
        for i, file_name in enumerate(files, start=1):
```

```

    print(f"{i}. {file_name}")

try:
    choice = int(input(f"\nEnter file number(1 - {len(files)}): "))
    if 1 <= choice <= len(files):
        chosen_file = os.path.join(directory, files[choice - 1])
        print(f"\nYou choose file: {files[choice - 1]}")
        return chosen_file
    else:
        print("error: number out of range.")
        return None
except ValueError:
    print("error: invalid input, please enter a number.")
    return None
except FileNotFoundError:
    print("error: directory not found.")
    return None

def user_read_type(read_type, file_path):
    with open(file_path, "r", encoding="utf-8") as file:
        if read_type == 'all':
            content = file.read()
            return content
        elif read_type == 'line':
            lines = file.readlines()
            try:
                line_number = int(input(f"Enter line number (1-{len(lines)}): "))
                if 1 <= line_number <= len(lines):
                    print(lines[line_number - 1].strip())
                else:
                    print("line number out of range")
            except ValueError:
                print("invalid input, please enter a number")
            return

```

```

    else:
        return "invalid choice"

user_choice_directory = input("enter directory path: ").strip()
file_path = choose_file_in_directory(user_choice_directory)
if file_path:
    action = input("do you want to (r)ead or (a)dd content to the file? (r/a): ").strip().lower()
    if action == 'a':
        content_to_add = input("enter content to add: \n").strip()
        edit_add_file(file_path, content_to_add)
    elif action == 'r':
        read_type = input("do you want to read the (a)ll content or a specific (l)ine? (a/l): ").strip().lower()
        if read_type == 'a':
            file_content = user_read_type('all', file_path)
            print("\nfile content:\n")
            print(file_content)
        elif read_type == 'l':
            user_read_type('line', file_path)
        else:
            print("invalid choice")
    else:
        print("invalid choice")

```

Заключение

В ходе лабораторной работы были изучены основные операции работы с файлами в Python: открытие, чтение, запись и добавление данных. Были рассмотрены различные режимы работы с файлами и способы чтения содержимого, включая построчное чтение и чтение всего файла целиком.

Также были получены навыки обработки ошибок с использованием конструкции try-except. Реализована корректная реакция программы на попытку открыть несуществующий файл, что повышает надежность и устойчивость программного кода.

