**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

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

по дисциплине «Введение в информационные технологии»

Выполнил: студент группы

БПИ2501

Карпеко Никита Александрович

Проверила: Мосева Марина Сергеевна

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.

**Задания**

**1. Открытие и чтение файла**

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, реализуйте выбор типа чтения в принимаемых аргументах функции.

**2. Запись в файл**

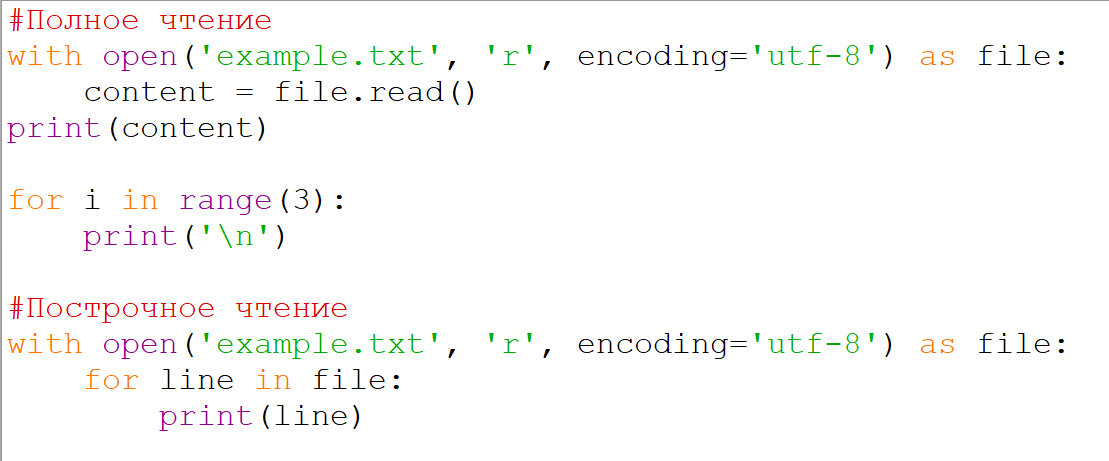
2.1. Написать программу, которая запрашивает у пользователя текст и записывает его в новый файл user\_input.txt.

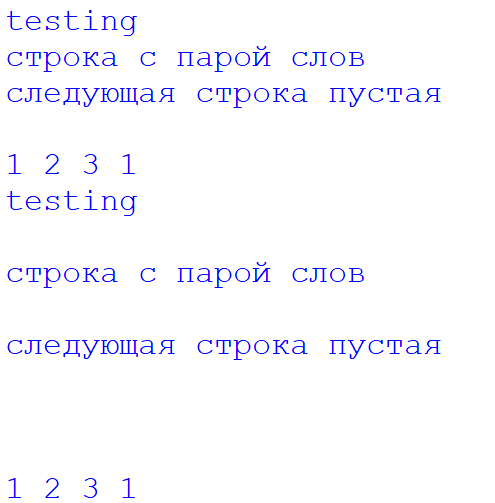
2.2. Реализовать функционал добавления текста в существующий файл, не удаляя его предыдущее содержимое.

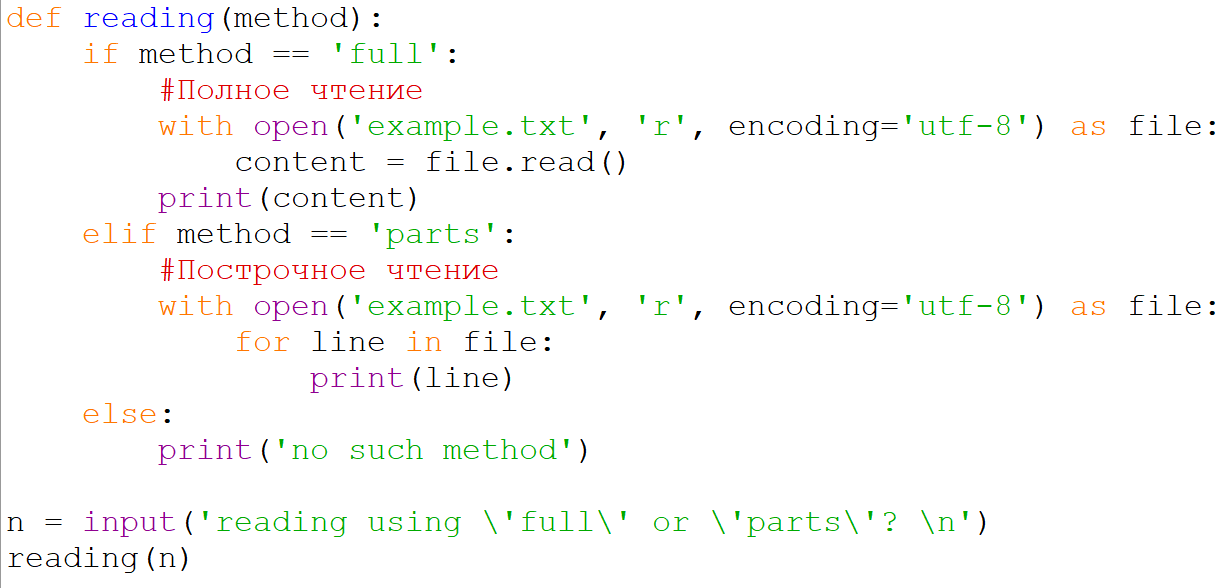
**3. Запись в файл**

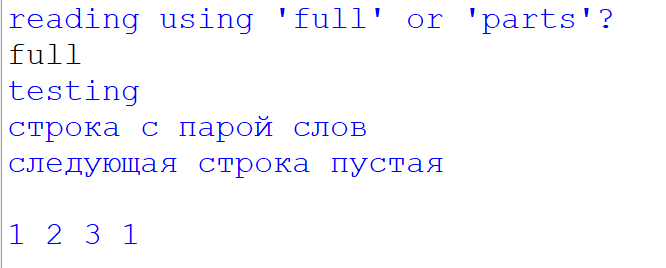
Модифицировать программу из Задания 1 так, чтобы она корректно обрабатывала исключение, возникающее при попытке открыть несуществующий файл. Вместо вывода ошибки программа должна выводить пользователю понятное сообщение. Использовать в блоке try except следующий класс исключений: **FileNotFoundError.**

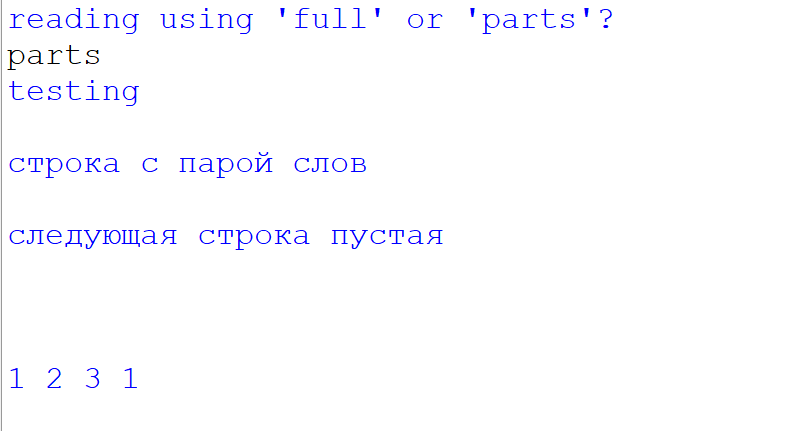
**Скриншоты выполнения (группы – код и вывод выполнения)**

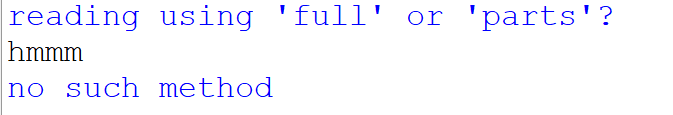
****

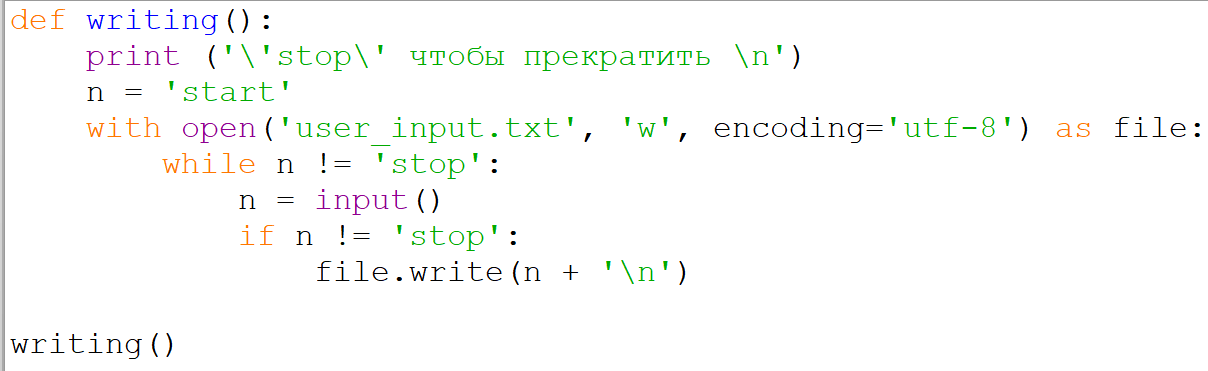
****

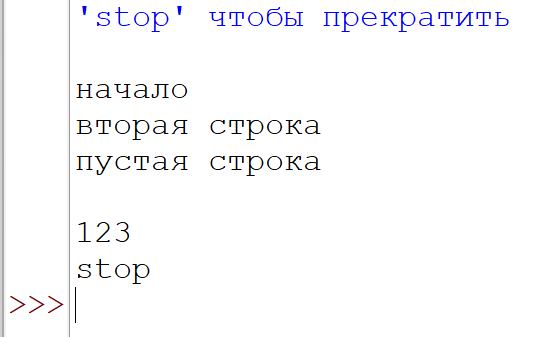
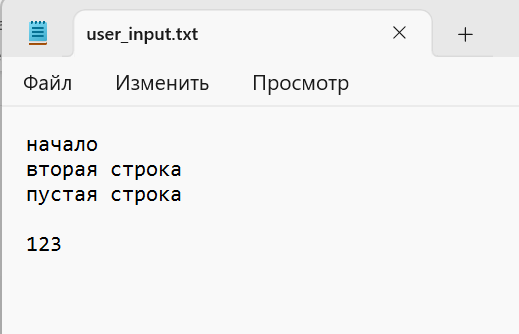






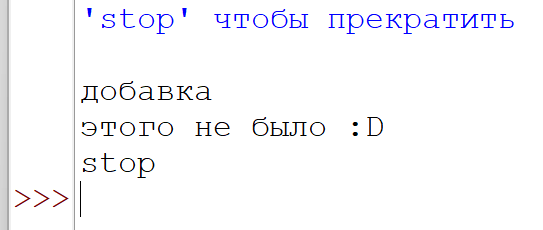
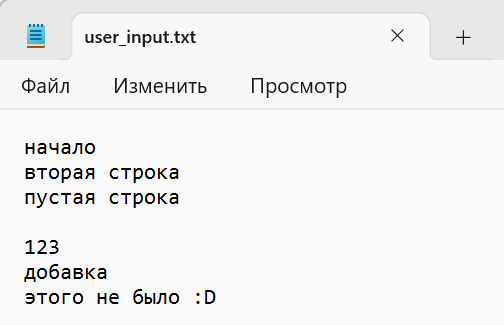


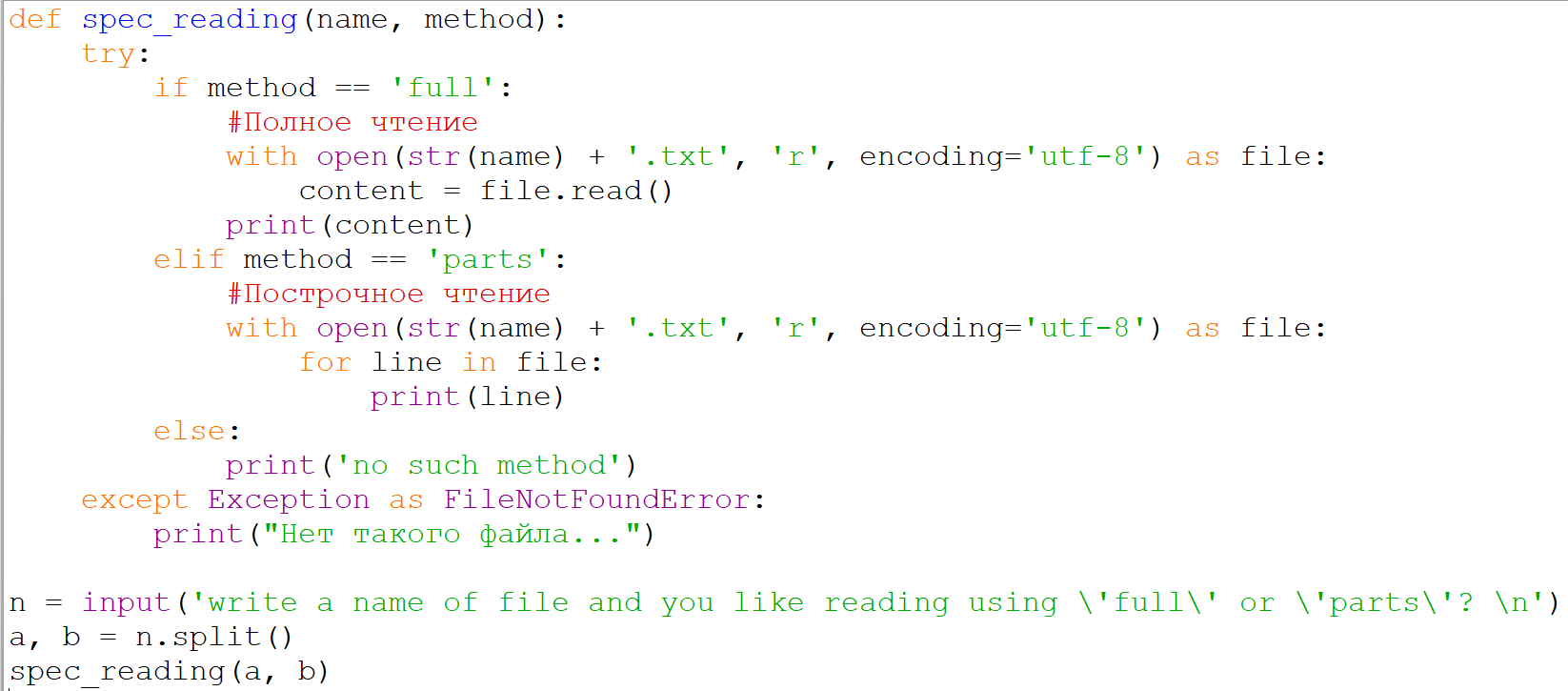


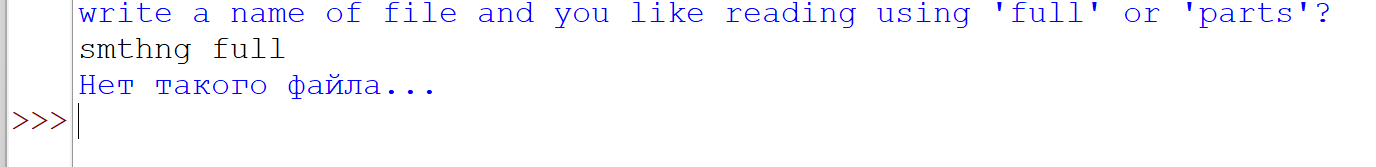
 

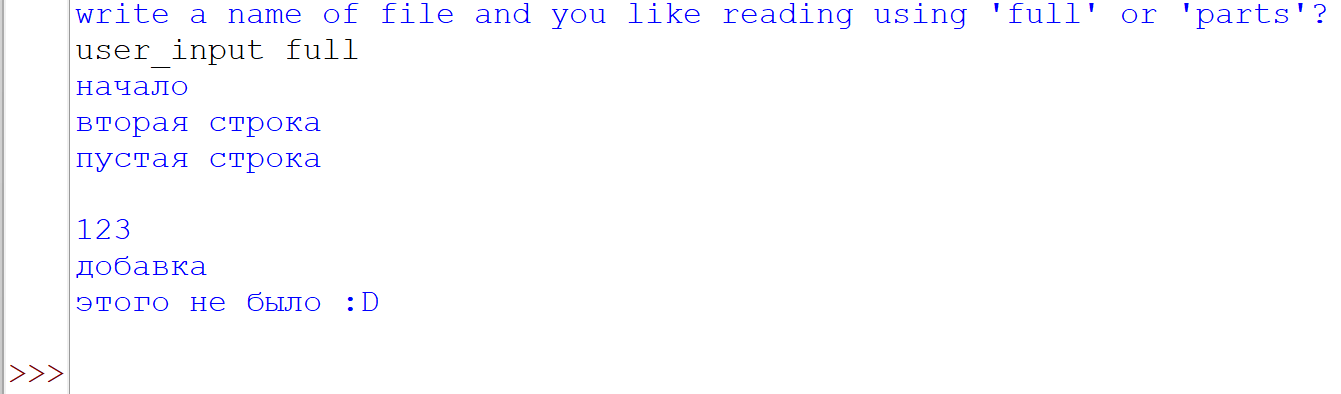
Заменив в строке with open('user\_input.txt', 'w', encoding='utf-8') as file:

W 🡪 A вместо записи заново, строки будут добавляться

**** ****

****

****

****

**Исходный код**

#Полное чтение

with open('example.txt', 'r', encoding='utf-8') as file:

content = file.read()

print(content)

#Построчное чтение

with open('example.txt', 'r', encoding='utf-8') as file:

for line in file:

print(line)

def reading(method):

if method == 'full':

#Полное чтение

with open('example.txt', 'r', encoding='utf-8') as file:

content = file.read()

print(content)

elif method == 'parts':

#Построчное чтение

with open('example.txt', 'r', encoding='utf-8') as file:

for line in file:

print(line)

else:

print('no such method')

n = input('reading using \'full\' or \'parts\'? \n')

reading(n)

def writing():

print ('\'stop\' чтобы прекратить \n')

n = 'start'

with open('user\_input.txt', 'a', encoding='utf-8') as file:

while n != 'stop':

n = input()

if n != 'stop':

file.write(n + '\n')

writing()

def spec\_reading(name, method):

try:

if method == 'full':

#Полное чтение

with open(str(name) + '.txt', 'r', encoding='utf-8') as file:

content = file.read()

print(content)

elif method == 'parts':

#Построчное чтение

with open(str(name) + '.txt', 'r', encoding='utf-8') as file:

for line in file:

print(line)

else:

print('no such method')

except Exception as FileNotFoundError:

print("Нет такого файла...")

n = input('write a name of file and you like reading using \'full\' or \'parts\'? \n')

a, b = n.split()

spec\_reading(a, b)

**Заключение**

Были освоены принципы определения и использования функций в языке программирования Python для открытия, чтения, редактирования файлов, поняты механизмы передачи аргументов в функции, связанные с этим, также было изучено - как применять функции для решения практических задач по записи текстовых данных в файлы.