

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ ФГАОУ ВО  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра инфокоммуникаций

**Дисциплина «Языки программирования»**

**Отчет по практической работе № 2.15**

**Работа с файлами в языке Python**

Выполнил: студент группы ИТС-б-о-22-1  
Наумов Никита Викторович

---

(подпись)

Проверил: к.т.н., доцент  
Кафедры инфокоммуникаций  
Воронкин Р.А.

---

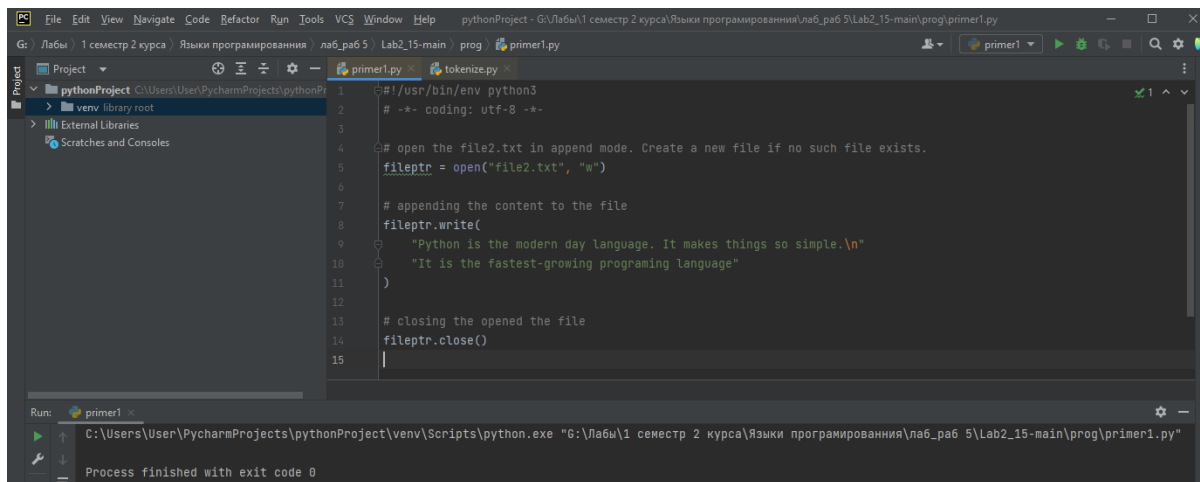
(подпись)

Ставрополь, 2023

**Цель работы:** приобретение навыков по работе с текстовыми файлами при написании программ с помощью языка программирования Python версии 3.x, изучение основных методов модуля os для работы с файловой системой, получение аргументов командной строки.

## Практическая часть:

### 1. Проработал примеры.

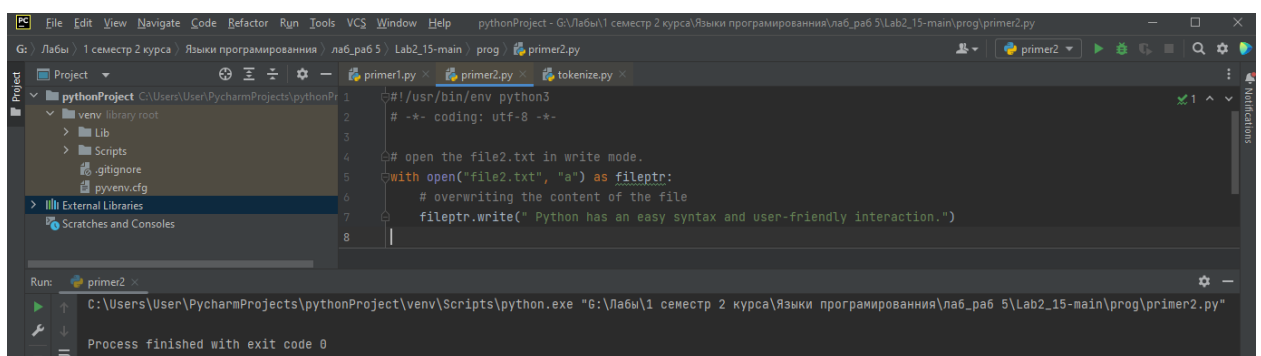


file2.txt – Блокнот

Файл Правка Формат Вид Справка

Python is the modern day language. It makes things so simple.  
It is the fastest-growing programming language

Рисунок 1. Результат работы примера 1



file2 – Блокнот

Файл Правка Формат Вид Справка

Python is the modern day language. It makes things so simple.  
It is the fastest-growing programming language Python has an easy syntax and user-friendly interaction.

Рисунок 2. Результат работы примера 2

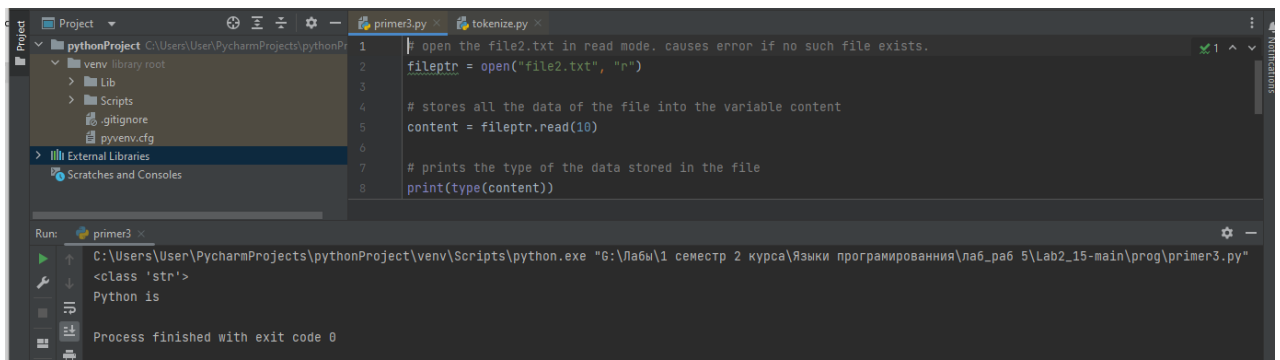


Рисунок 3. Результат работы примера 3

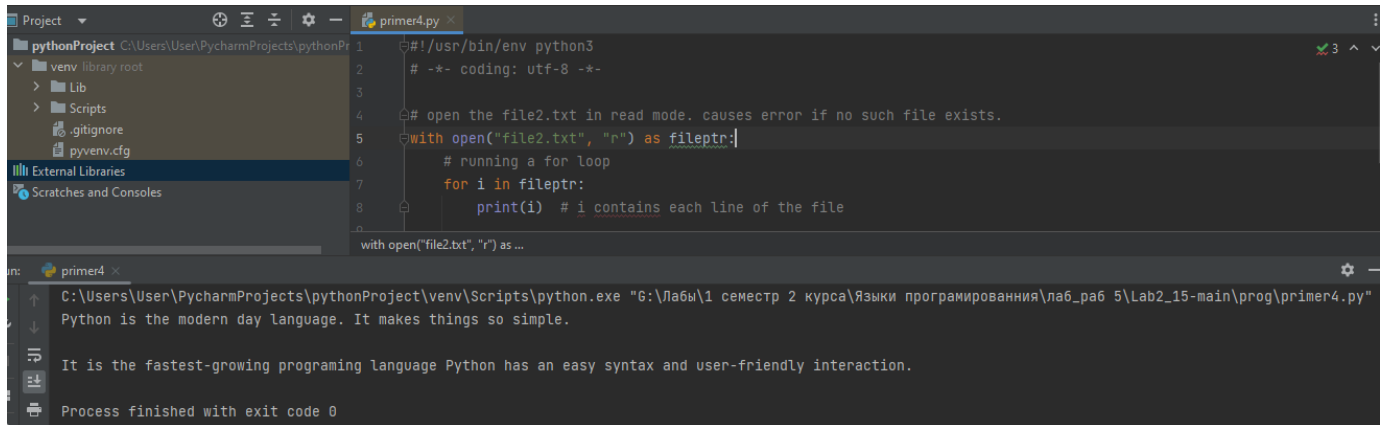


Рисунок 4. Результат работы примера 4

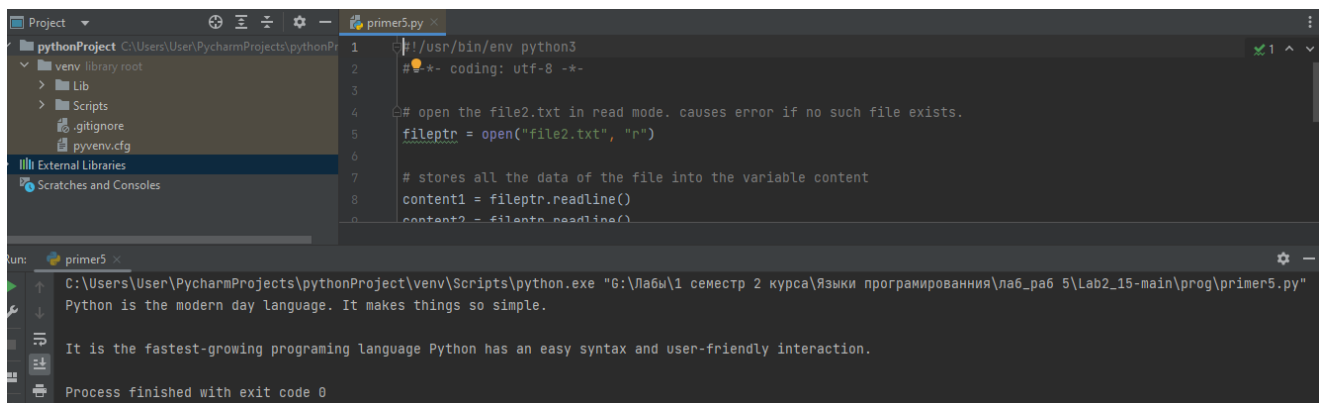


Рисунок 5. Результат работы примера 5

```
1 #!/usr/bin/env python3
2 #-*- coding: utf-8 -*-
3
4 # open the file2.txt in read mode. causes error if no such file exists.
5 with open("file2.txt", "r") as fileptr:
6     # stores all the data of the file into the variable content
7     content = fileptr.readlines()
8     # prints the content of the file
9     print(content)
```

Process finished with exit code 0

Рисунок 6. Результат работы примера 6

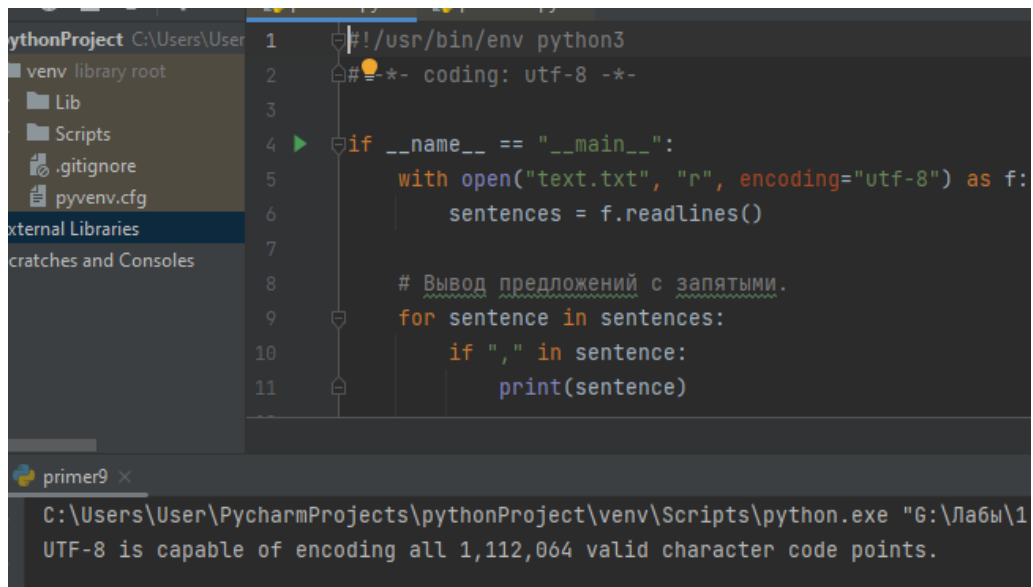
```
2 #-*- coding: utf-8 -*-
3
4 # open the newfile.txt in read mode. causes error if no such file exists.
5 fileptr = open("newfile.txt", "x")
6 print(fileptr)
7 if fileptr:
8     print("File created successfully")
9     # closes the opened file
10    fileptr.close()
```

Process finished with exit code 0

Рисунок 7. Результат работы примера 7

```
1 #!/usr/bin/env python3
2 #-*- coding: utf-8 -*-
3
4 if __name__ == "__main__":
5     # open the text.txt in append mode. Create a new file if no such file exists.
6     with open("text.txt", "w", encoding="utf-8") as fileptr:
7         # appending the content to the file
8         print(
9             "UTF-8 is a variable-width character encoding used for electronic communication.",
10            file=fileptr,
11        )
12         print(
13             "UTF-8 is capable of encoding all 1,112,064 valid character code points.",
14            file=fileptr,
15        )
16         print("In Unicode using one to four one-byte (8-bit) code units.", file=fileptr)
```

Process finished with exit code 0

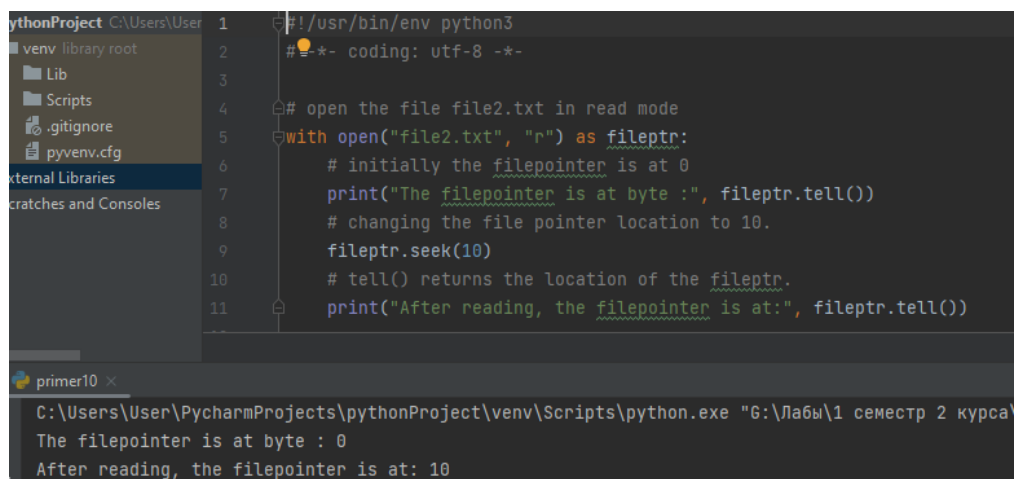


The screenshot shows the PyCharm IDE with a Python project named 'pythonProject'. The file explorer on the left shows the project structure: 'venv' (library root, Lib, Scripts), '.gitignore', and 'pyvenv.cfg'. The main editor displays a Python script with the following code:

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == "__main__":
5     with open("text.txt", "r", encoding="utf-8") as f:
6         sentences = f.readlines()
7
8     # Вывод предложений с запятыми.
9     for sentence in sentences:
10         if "," in sentence:
11             print(sentence)
```

The console output at the bottom shows the command executed: `C:\Users\User\PycharmProjects\pythonProject\venv\Scripts\python.exe "G:\Лабы\1"` and the output: `UTF-8 is capable of encoding all 1,112,064 valid character code points.`

Рисунок 8. Результат работы примера 9

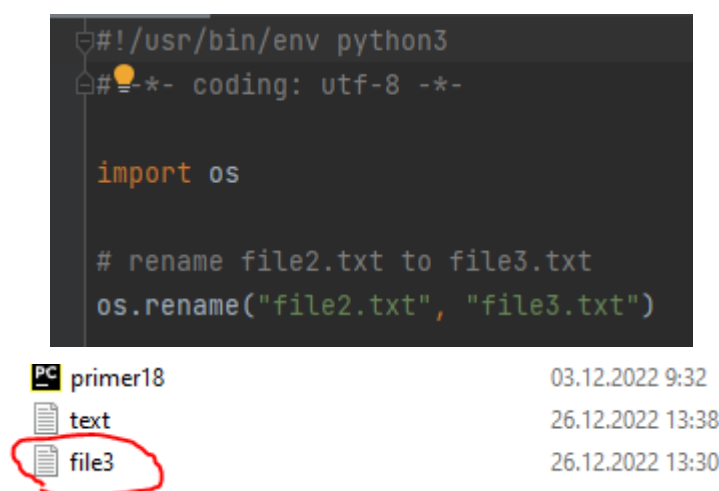


The screenshot shows the PyCharm IDE with a Python project named 'pythonProject'. The file explorer on the left shows the project structure: 'venv' (library root, Lib, Scripts), '.gitignore', and 'pyvenv.cfg'. The main editor displays a Python script with the following code:

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 # open the file file2.txt in read mode
5 with open("file2.txt", "r") as fileptr:
6     # initially the filepointer is at 0
7     print("The filepointer is at byte :", fileptr.tell())
8     # changing the file pointer location to 10.
9     fileptr.seek(10)
10    # tell() returns the location of the fileptr.
11    print("After reading, the filepointer is at:", fileptr.tell())
```

The console output at the bottom shows the command executed: `C:\Users\User\PycharmProjects\pythonProject\venv\Scripts\python.exe "G:\Лабы\1 семестр 2 курса"` and the output: `The filepointer is at byte : 0` and `After reading, the filepointer is at: 10`

Рисунок 9. Результат работы примера 10



The screenshot shows the PyCharm IDE with a Python project named 'pythonProject'. The file explorer on the left shows the project structure: 'venv' (library root, Lib, Scripts), '.gitignore', and 'pyvenv.cfg'. The main editor displays a Python script with the following code:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import os

# rename file2.txt to file3.txt
os.rename("file2.txt", "file3.txt")
```

The console output at the bottom shows the command executed: `C:\Users\User\PycharmProjects\pythonProject\venv\Scripts\python.exe "G:\Лабы\1 семестр 2 курса"` and the output: `The filepointer is at byte : 0` and `After reading, the filepointer is at: 10`

Below the code, there is a file explorer showing the files in the project: 'text' and 'file3'. The file 'file3' is circled in red.

Рисунок 10. Результат работы примера 11

Возвращает наш 3 файл

Имя	Дата изменения	И
individual1.py	29.11.2022 22:42	И
individual2.py	29.11.2022 22:42	И
newfile.txt	29.11.2022 22:43	Т
primer1.py	29.11.2022 22:29	И
primer2.py	29.11.2022 22:32	И

Рисунок 11. Результат работы примера 12

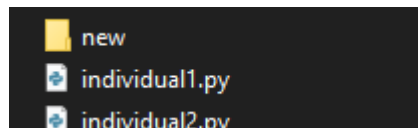
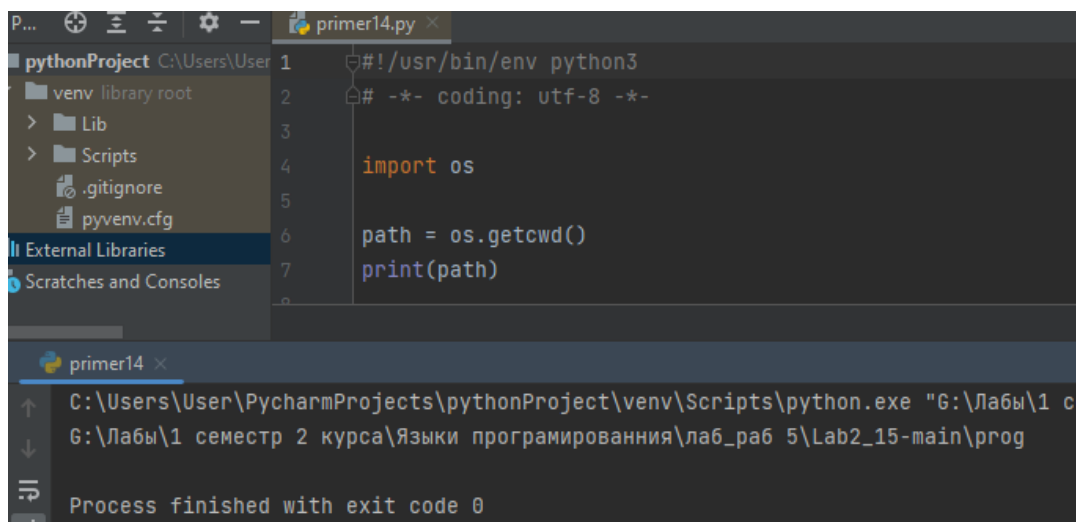


Рисунок 12. Результат работы примера 13

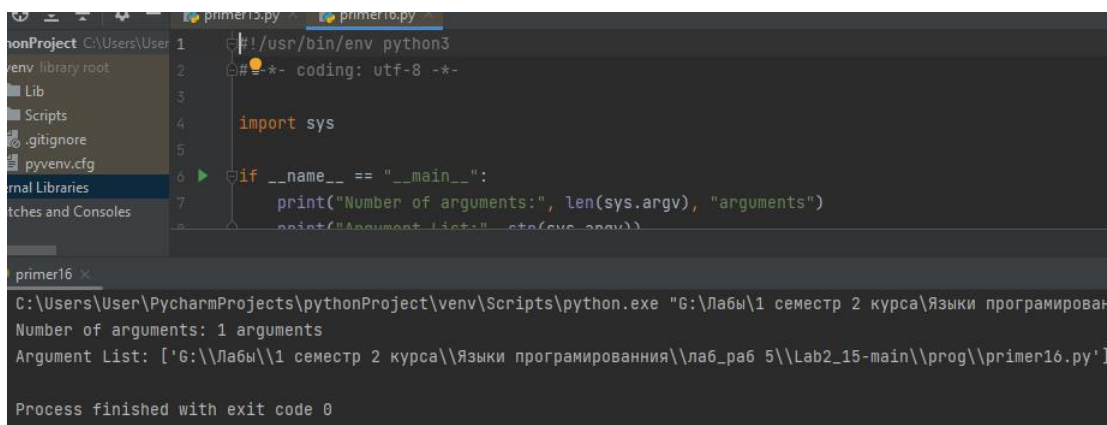
A screenshot of the PyCharm IDE. The top editor shows a file named 'primer14.py' with the following code:

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import os
5
6 path = os.getcwd()
7 print(path)
```

The bottom console shows the execution output:

```
C:\Users\User\PycharmProjects\pythonProject\venv\Scripts\python.exe "G:\Лабы\1 семестр 2 курса\Языки программирования\лаб_раб 5\Lab2_15-main\prog
G:\Лабы\1 семестр 2 курса\Языки программирования\лаб_раб 5\Lab2_15-main\prog
Process finished with exit code 0
```

Рисунок 13. Результат работы примера 14

A screenshot of the PyCharm IDE. The top editor shows a file named 'primer16.py' with the following code:

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import sys
5
6 if __name__ == "__main__":
7     print("Number of arguments:", len(sys.argv), "arguments")
8     print("Argument List:" + str(sys.argv))
```

The bottom console shows the execution output:

```
C:\Users\User\PycharmProjects\pythonProject\venv\Scripts\python.exe "G:\Лабы\1 семестр 2 курса\Языки программирования\лаб_раб 5\Lab2_15-main\prog\primer16.py"
Number of arguments: 1 arguments
Argument List: ['G:\Лабы\1 семестр 2 курса\Языки программирования\лаб_раб 5\Lab2_15-main\prog\primer16.py']
Process finished with exit code 0
```

Рисунок 14. Результат работы примера 16

2. Выполнил индивидуальные задания.

Задание 1. Написать программу, которая считывает текст из файла и выводит на экран только цитаты, то есть предложения, заключенные в кавычки.

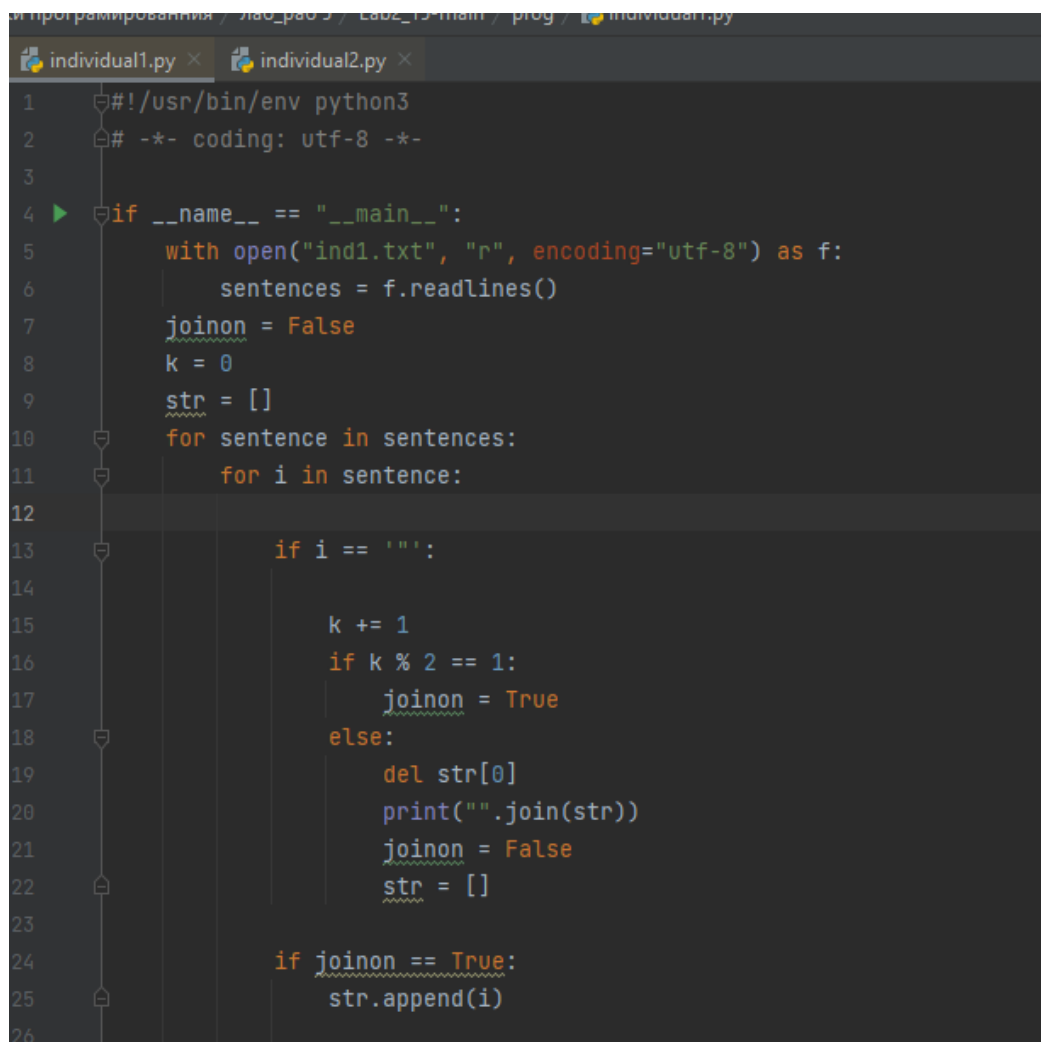
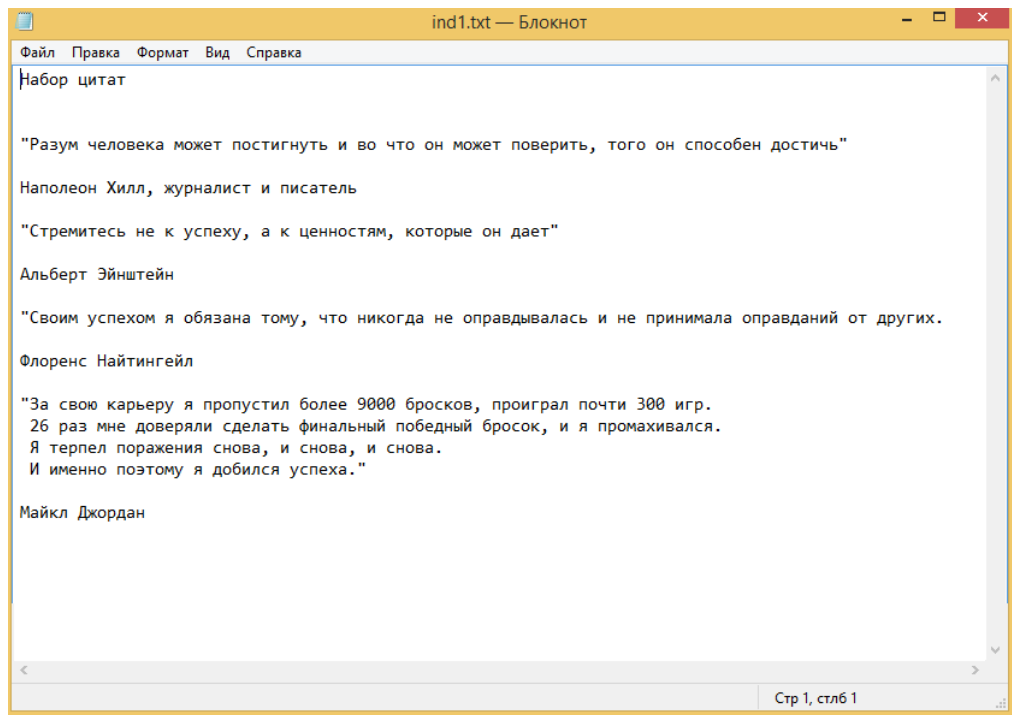
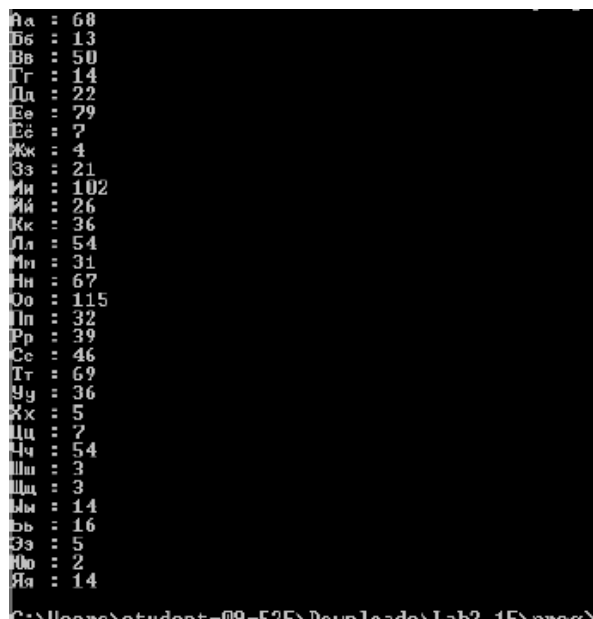


Рисунок 16. Индивидуальное задание 1

Задание 2. Как вы знаете, в языке Python для создания комментариев в коде используется символ #. Комментарий начинается с этого символа и продолжается до конца строки – без возможности остановить его раньше. В данном упражнении вам предстоит написать программу, которая будет удалять все комментарии из исходного файла с кодом на языке Python. Пройдите по всем строкам в файле на предмет поиска символа #. Обнаружив его, программа должна удалить все содержимое, начиная с этого символа и до конца строки. Для простоты не будем рассматривать ситуации, когда знак решетки встречается в середине строки. Сохраните новое содержимое в созданном файле. Имена файла источника и файла назначения должны быть запрошены у пользователя. Удостоверьтесь в том, что программа корректно обрабатывает возможные ошибки при работе с обоими файлами.



```
Аа : 68
Бб : 13
Вв : 50
Гг : 14
Дд : 22
Ее : 79
Ёё : 7
Жж : 4
Зз : 21
Ии : 102
Йй : 26
Кк : 36
Лл : 54
Мм : 31
Нн : 67
Оо : 115
Пп : 32
Рр : 39
Сс : 46
Тт : 69
Уу : 36
Хх : 5
Цц : 7
Чч : 54
Шш : 3
Щщ : 3
Ыы : 14
Ьь : 16
Ээ : 5
Юю : 2
Яя : 14
C:\Users\student=08-525\Downloads\Lab2_15\prog>
```

Рисунок 17. Индивидуальное задание 2

### Контрольные вопросы:

1. Как открыть файл в языке Python только для чтения?

r – открывает файл в режиме только для чтения. Указатель файла существует в начале. Файл по умолчанию открывается в этом режиме, если не передан режим доступа.

Например: fileptr = open("file.txt", "r").



## 2. Как открыть файл в языке Python только для записи?

w – только для записи. Он перезаписывает файл, если он существовал ранее, или создает новый, если файл с таким именем не существует. Указатель имеется в начале файла.

## 3. Как прочитать данные из файла в языке Python?

Чтобы прочитать файл с помощью сценария Python, Python предоставляет метод read(). Метод read() считывает строку из файла. Он может читать данные как в текстовом, так и в двоичном формате.

Синтаксис метода read():fileobj.read(<count>).

Мы можем прочитать файл построчно, используя цикл for .

Python упрощает чтение файла построчно с помощью метода readline() . Метод readline() читает строки файла с самого начала, т. е. если мы используем его два раза, мы можем получить первые две строки файла.

## 4. Как записать данные в файл в языке Python?

Чтобы записать текст в файл, нам нужно открыть файл с помощью метода open с одним из следующих режимов доступа.

'w': он перезапишет файл, если какой-либо файл существует. Указатель файла находится в начале файла.

'a': добавит существующий файл. Указатель файла находится в конце файла. Он создает новый файл, если файл не существует.

## 5. Как закрыть файл в языке Python?

После того, как все операции будут выполнены с файлом, мы должны закрыть его с помощью нашего скрипта Python, используя метод close() . Любая незаписанная информация уничтожается после вызова метода close() для файлового объекта.

6. Изучите самостоятельно работу конструкции with ... as. Каково ее назначение в языке

Python? Где она может быть использована еще, помимо работы с файлами?

Конструкция `with ... as` используется для оборачивания выполнения блока инструкций менеджером контекста.

Синтаксис конструкции `with ... as`:

```
"with" expression ["as" target] ("," expression ["as" target])* ":"  
suite
```

Теперь по порядку о том, что происходит при выполнении данного блока:

- 1) Выполняется выражение в конструкции `with ... as`.
- 2) Загружается специальный метод `__exit__` для дальнейшего использования.
- 3) Выполняется метод `__enter__`. Если конструкция `with` включает в себя слово `as`, то возвращаемое методом `__enter__` значение записывается в переменную.
- 4) Выполняется `suite`.
- 5) Вызывается метод `__exit__`, причём неважно, выполнилось ли `suite` или произошло исключение. В этот метод передаются параметры исключения, если оно произошло, или во всех аргументах значение `None`, если исключения не было.

Если в конструкции `with - as` было несколько выражений, то это эквивалентно нескольким вложенным конструкциям.

7. Изучите самостоятельно документацию Python по работе с файлами. Какие помимо рассмотренных существуют методы записи/чтения информации из файла?

8. Какие существуют, помимо рассмотренных, функции модуля `os` для работы с файловой системой?

Функции модуля `os`:

- Получение информации об ОС Чтобы узнать имя текущей ОС, достаточно воспользоваться методом `name`. ...
- Изменение рабочей директории ...

- Проверка существования пути ...
- Создание директорий ...
- Удаление файлов и директорий ...
- Запуск на исполнение ...
- Получение имени файла и директории ...
- Вычисление размера

**Вывод:** в результате выполнения работы были приобретены навыки по работе с текстовыми файлами при написании программ с помощью языка программирования Python версии 3.x, изучение основных методов модуля os для работы с файловой системой, получение аргументов командной строки.