

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №2.15**  
**дисциплины «Программирование на языке Python»**

Выполнила:  
Мурашко Анастасия Юрьевна  
2 курс, группа ИТС-б-о-22-1,  
11.03.02 «Инфокоммуникационные  
технологии и системы связи»,  
направленность (профиль)  
«Инфокоммуникационные системы и  
сети», очная форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Р. А., доцент кафедры  
инфокоммуникаций

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2023 г.

**Тема:** работа с файлами на языке Python.

**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. Проработала примеры.

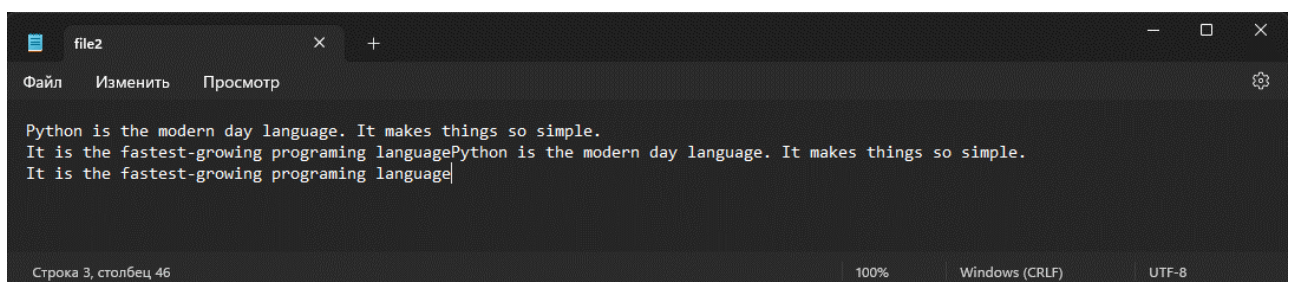
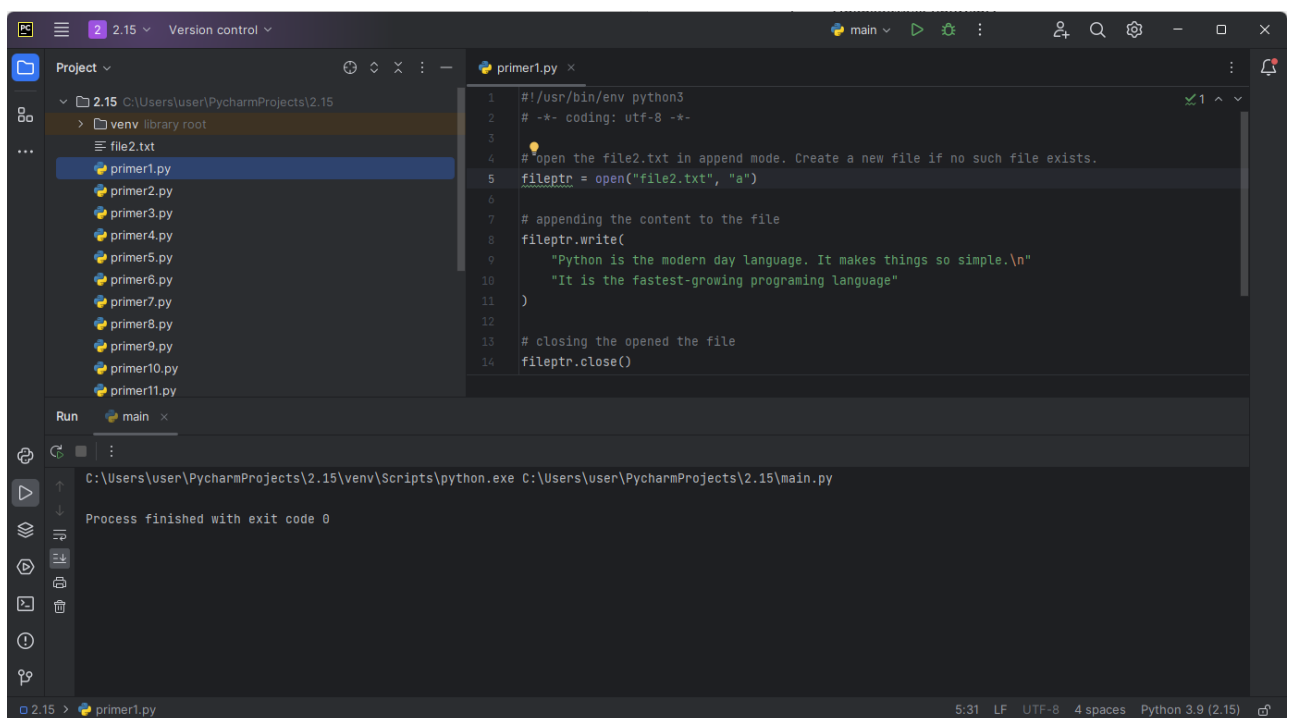


Рисунок 1. Результат работы примера 1

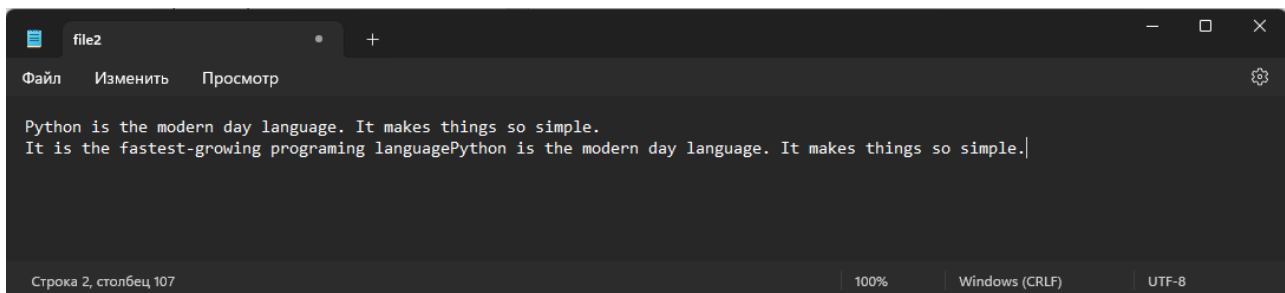
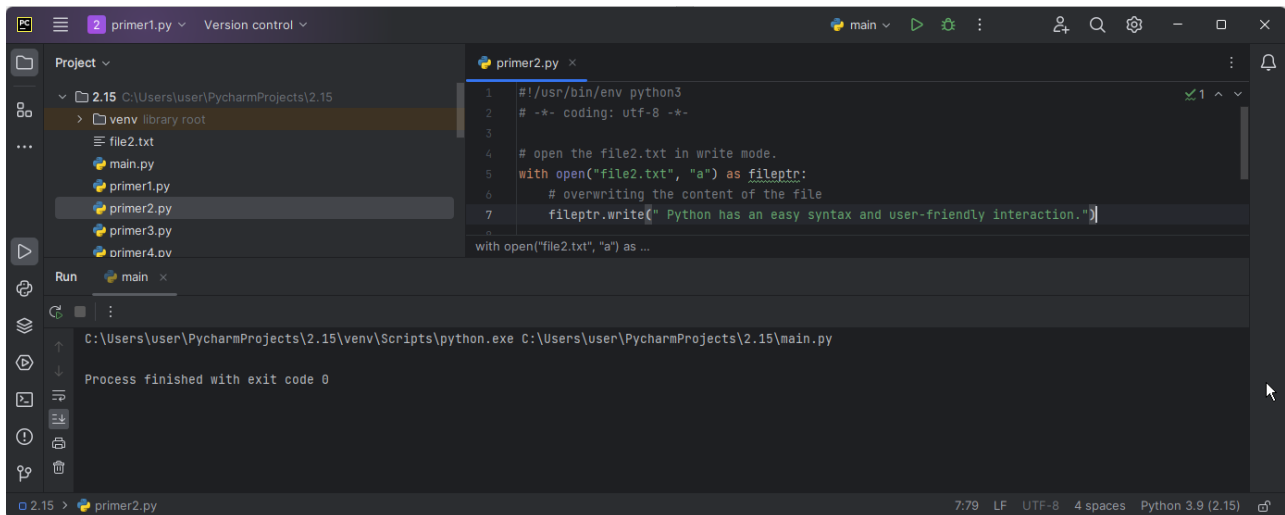


Рисунок 2. Результат работы примера 2

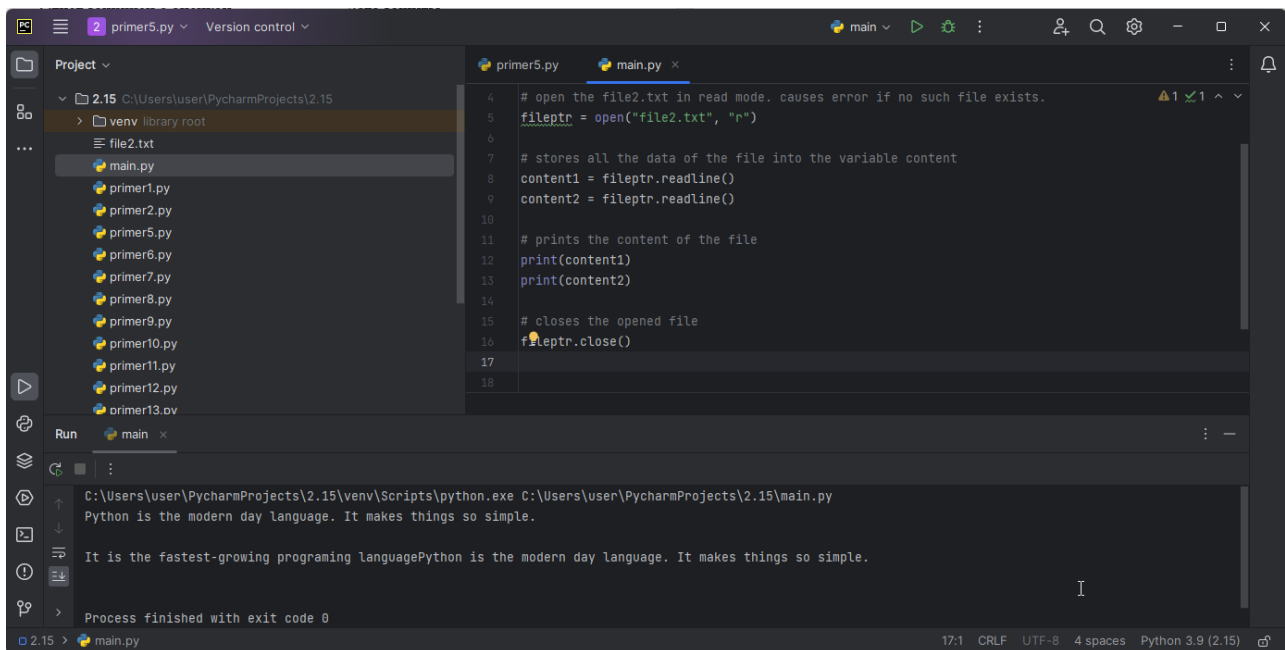


Рисунок 3. Результат работы примера 3

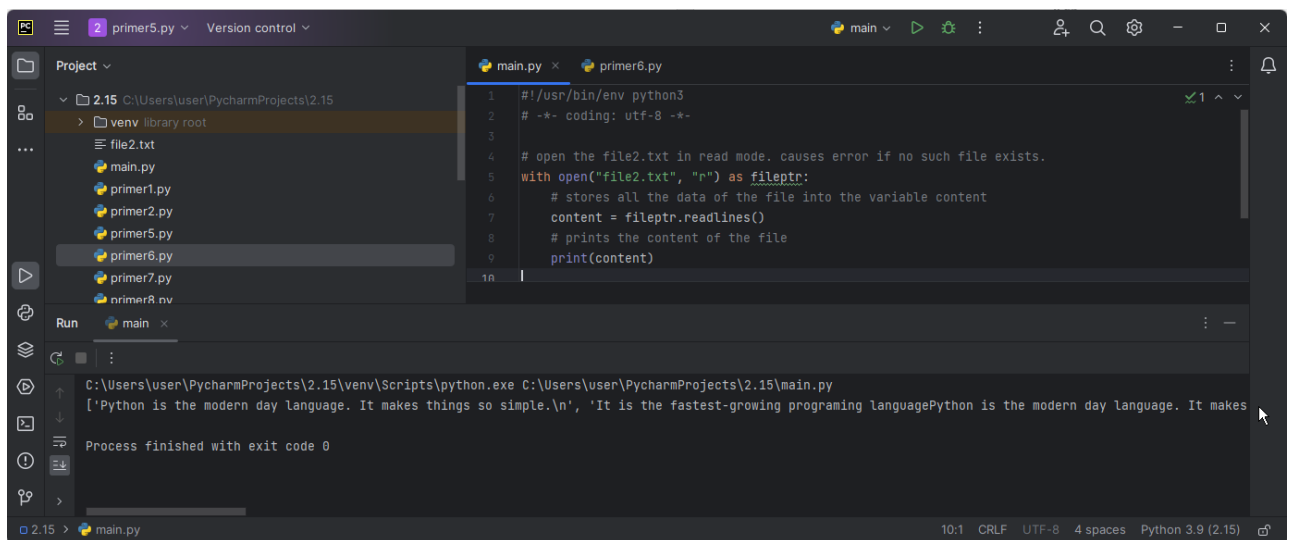


Рисунок 4. Результат работы примера 4

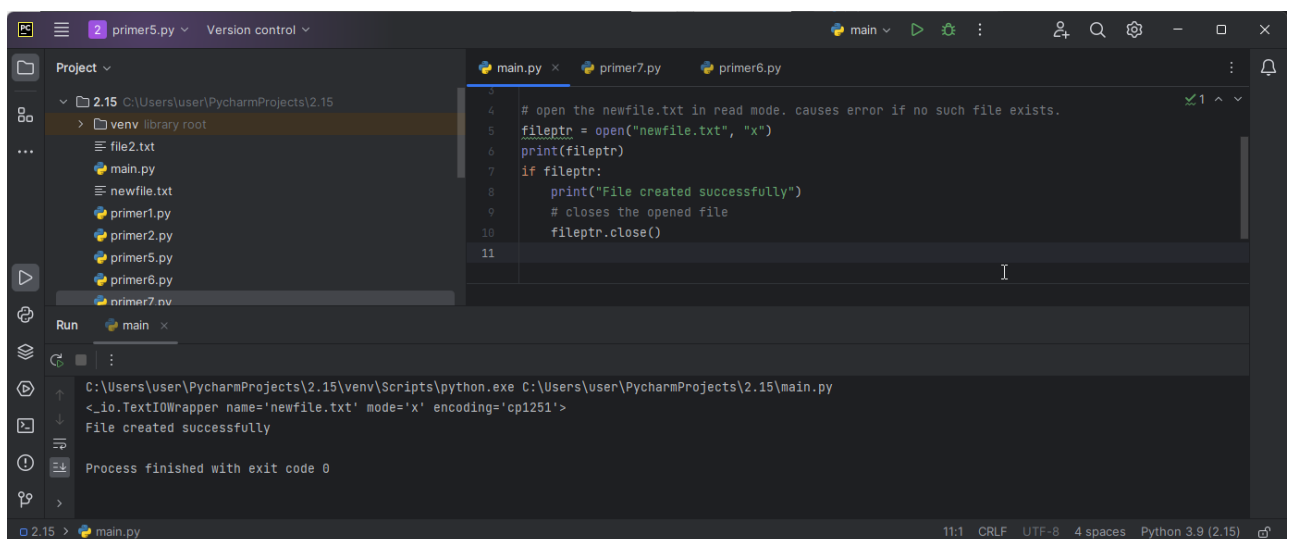
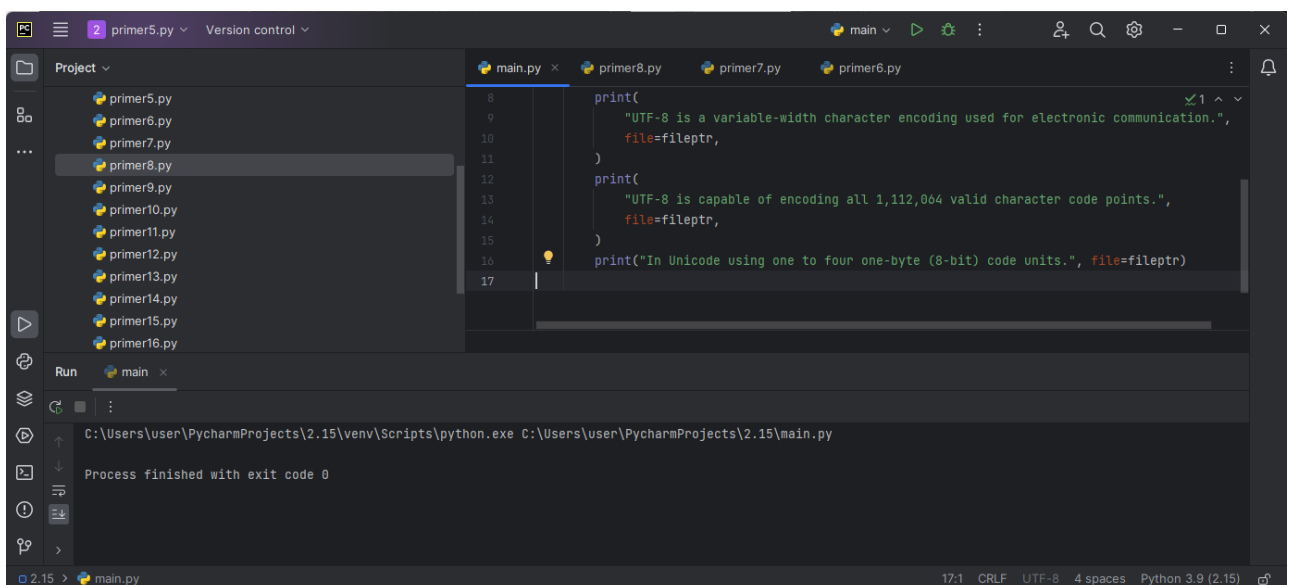


Рисунок 5. Результат работы примера 5



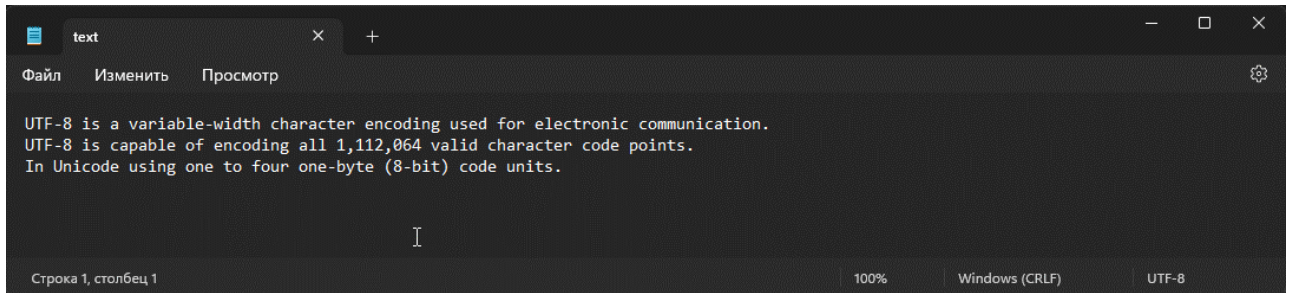


Рисунок 6. Результат работы примера 6

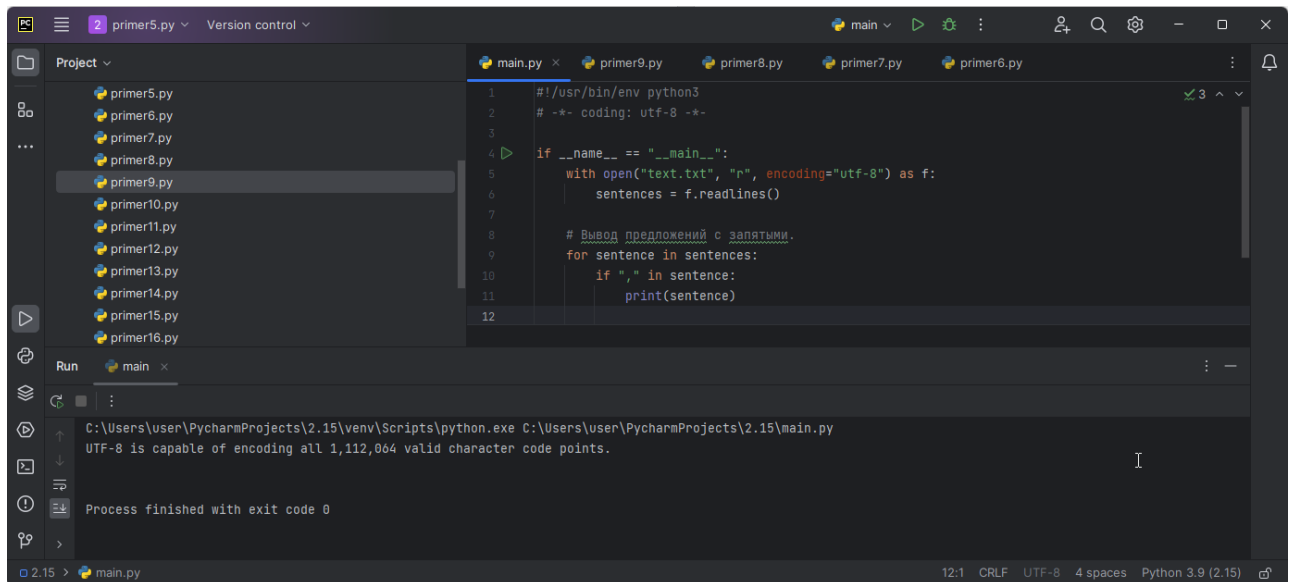


Рисунок 7. Результат работы примера 7

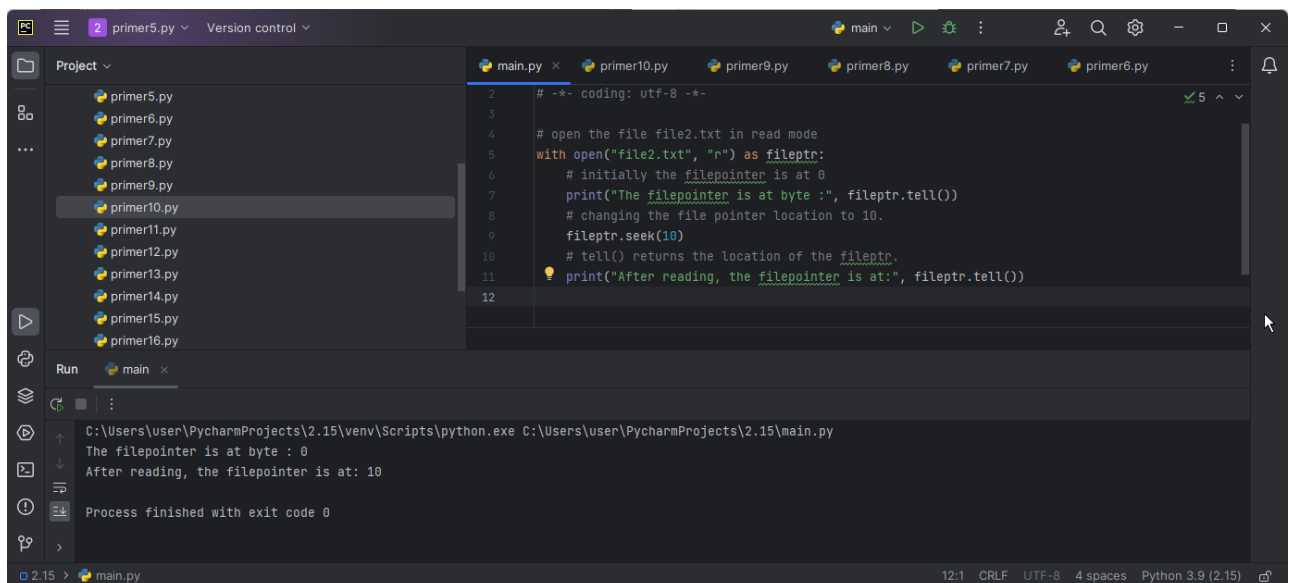
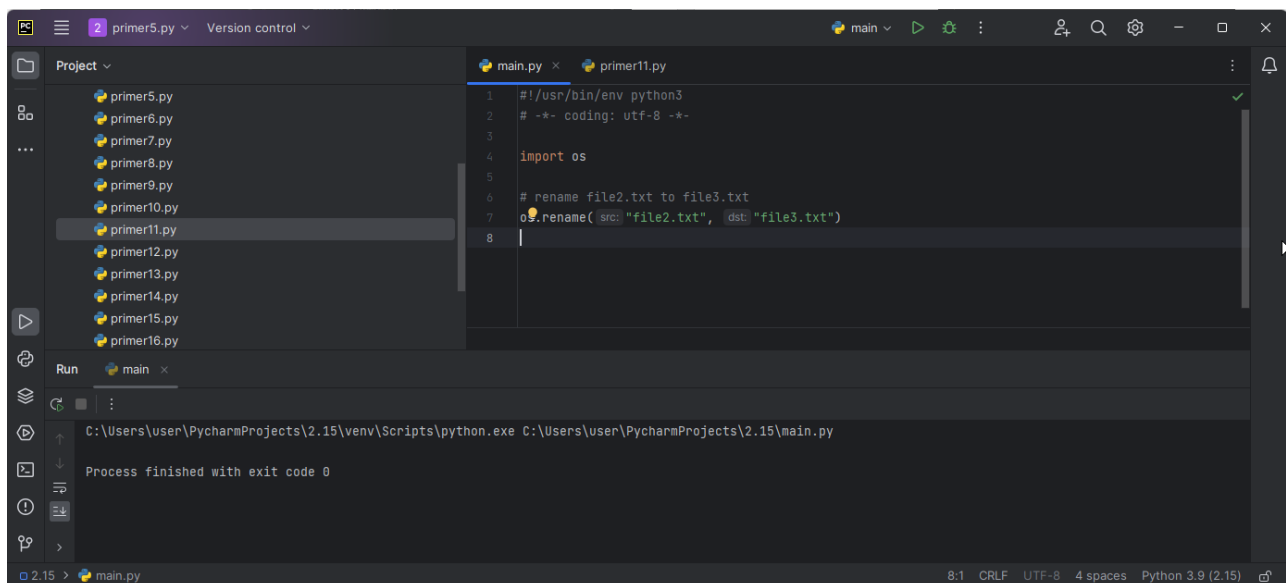


Рисунок 8. Результат работы примера 8






	.idea	30.09.2023 10:53	Папка с файлами	
	venv	30.09.2023 10:30	Папка с файлами	
	file3	30.09.2023 10:53	Текстовый докум...	2 КБ

Рисунок 9. Результат работы примера 9

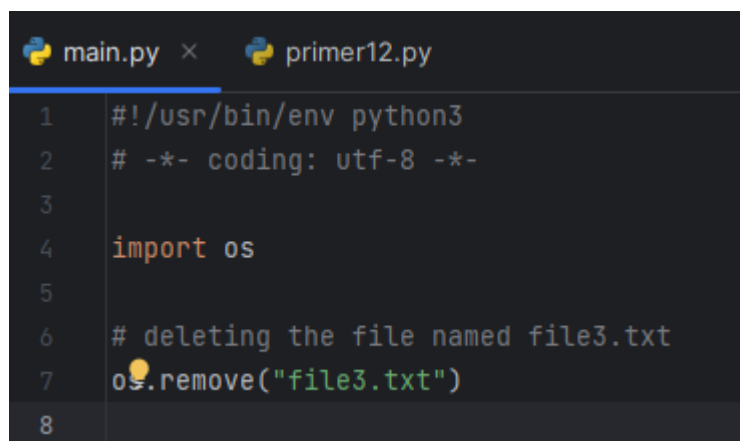
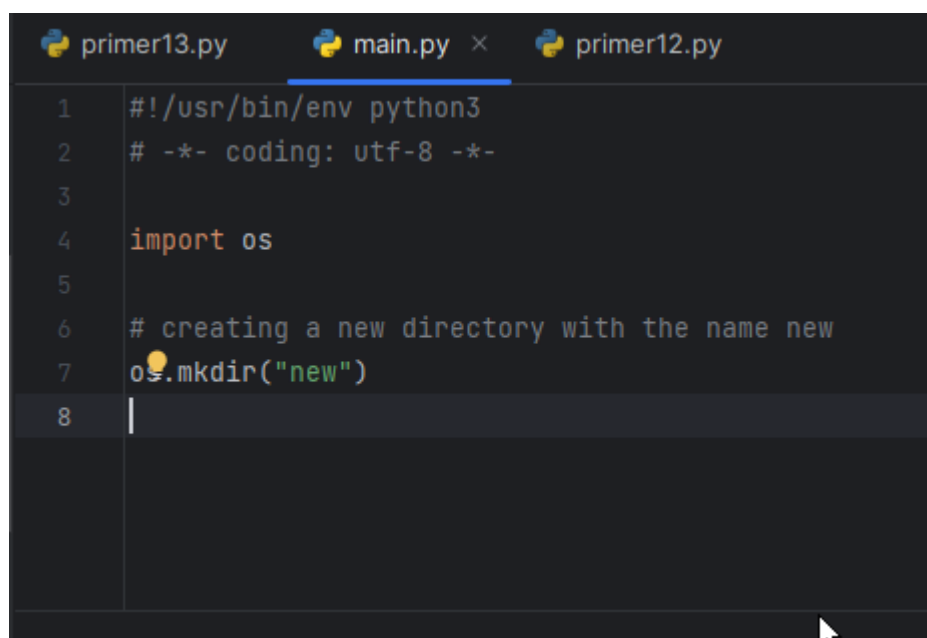


Рисунок 10. Результат работы примера 10






	new	30.09.2023 11:02
	venv	30.09.2023 10:30
	main	30.09.2023 11:02

Рисунок 11. Результат работы примера 11

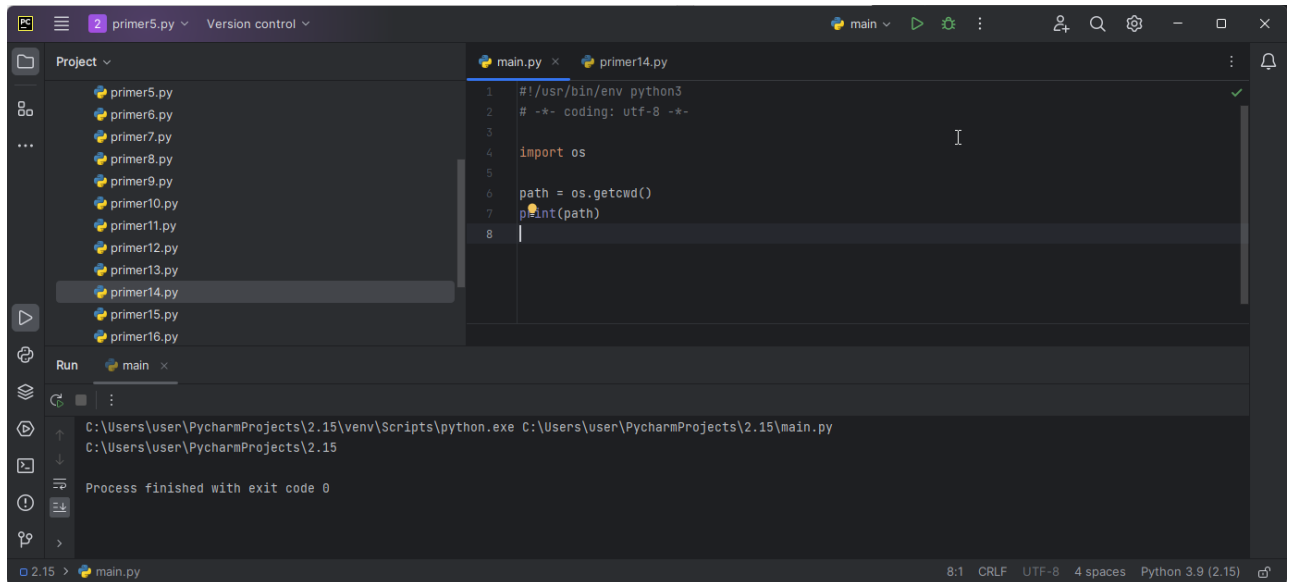


Рисунок 12. Результат работы примера 12

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import os
5
6 # Changing current directory with the new directory
7 os.chdir("C:\\Windows")
8 #It will display the current working directory
9 print(os.getcwd())

```

Рисунок 13. Результат работы примера 13

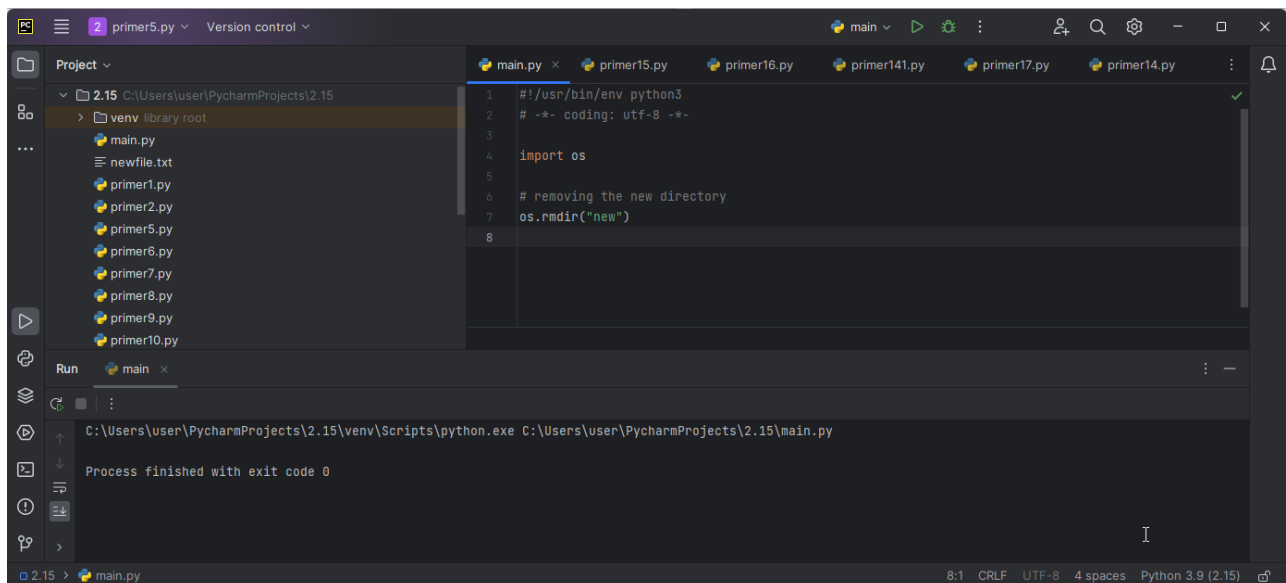


Рисунок 14. Результат работы примера 14

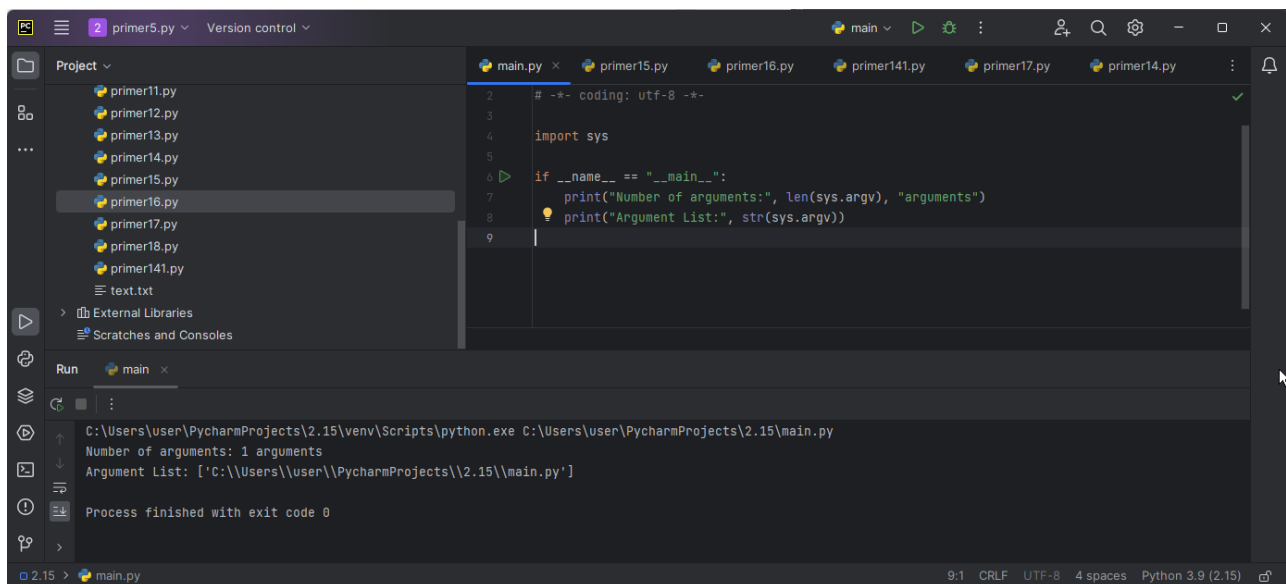


Рисунок 15. Результат работы примера 15



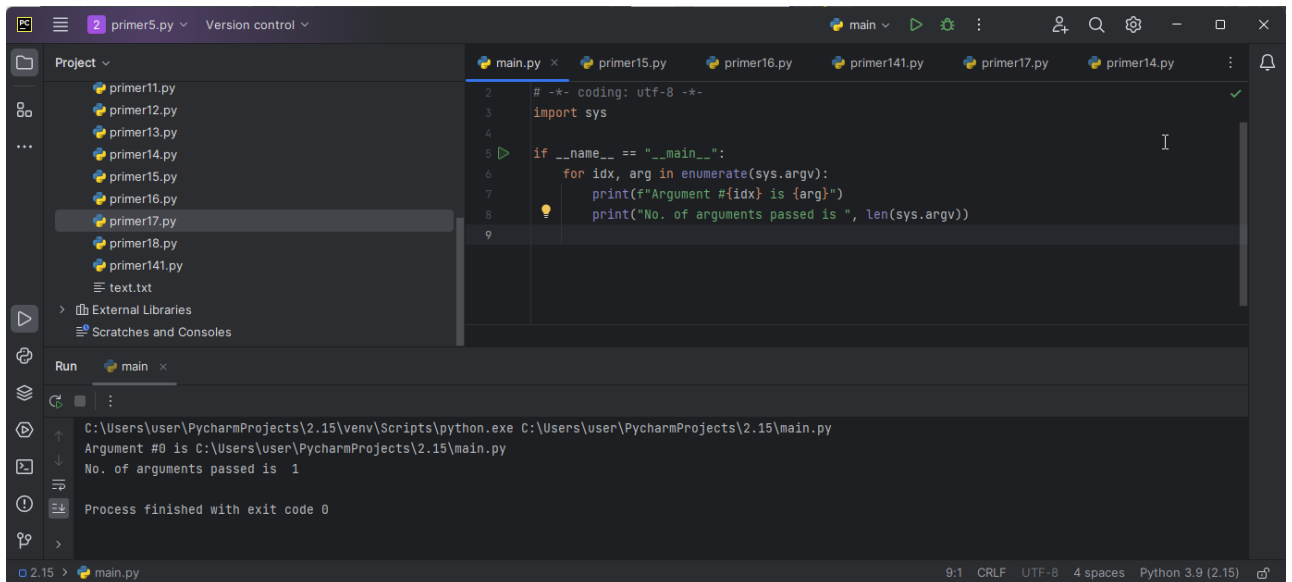


Рисунок 16. Результат работы примера 16

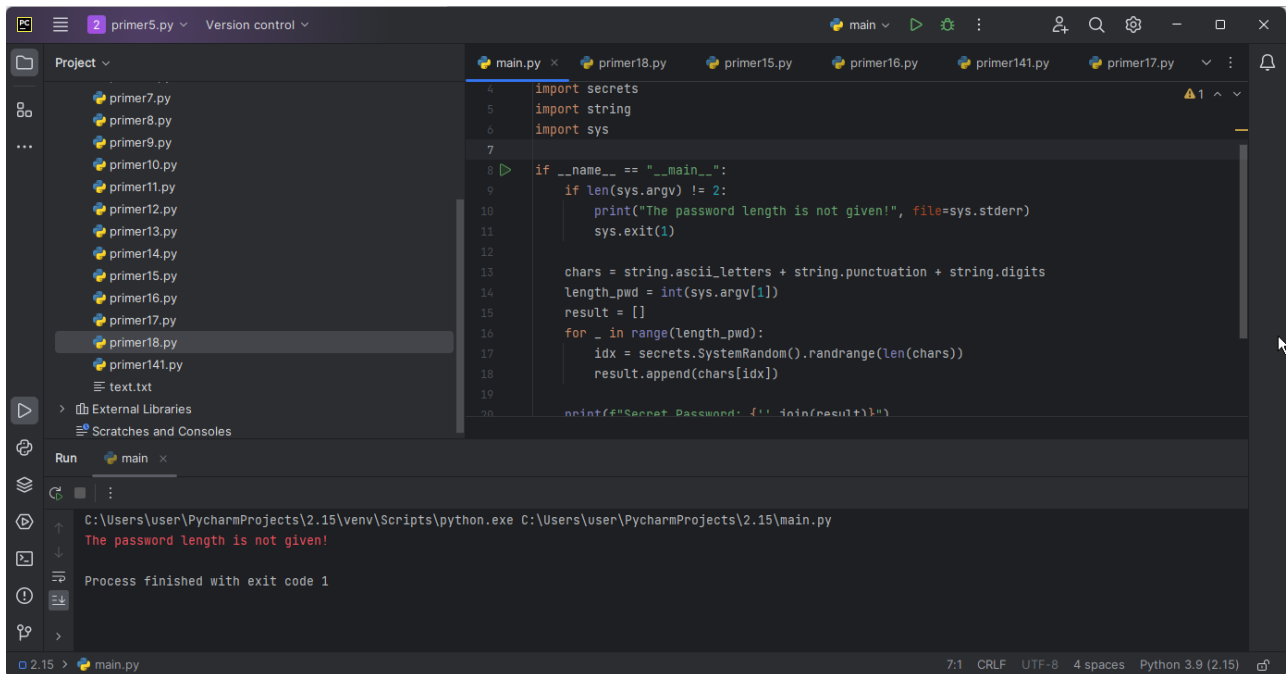


Рисунок 17. Результат работы примера 17

## 2. Выполнил индивидуальные задания.

Задание 1. Написать программу, которая считывает английский текст из файла и выводит на экран слова текста, начинающиеся и оканчивающиеся на гласные буквы.

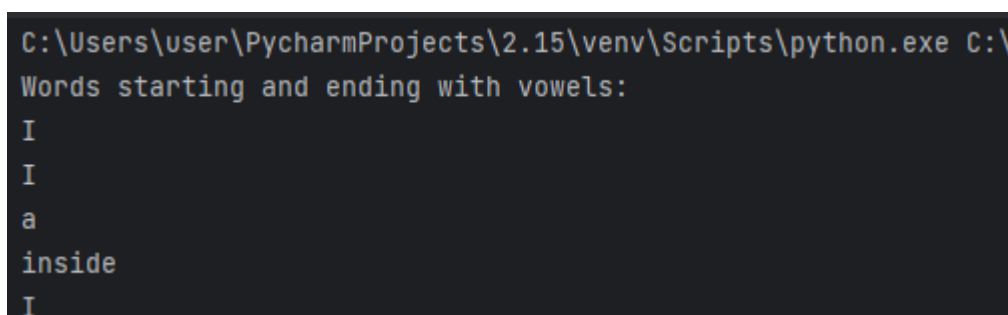
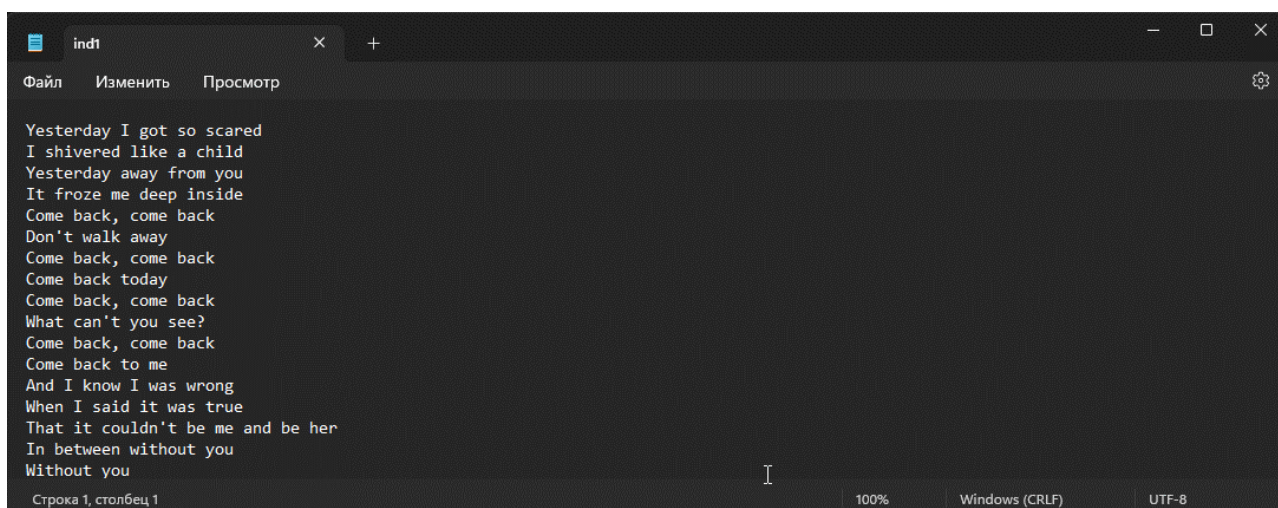
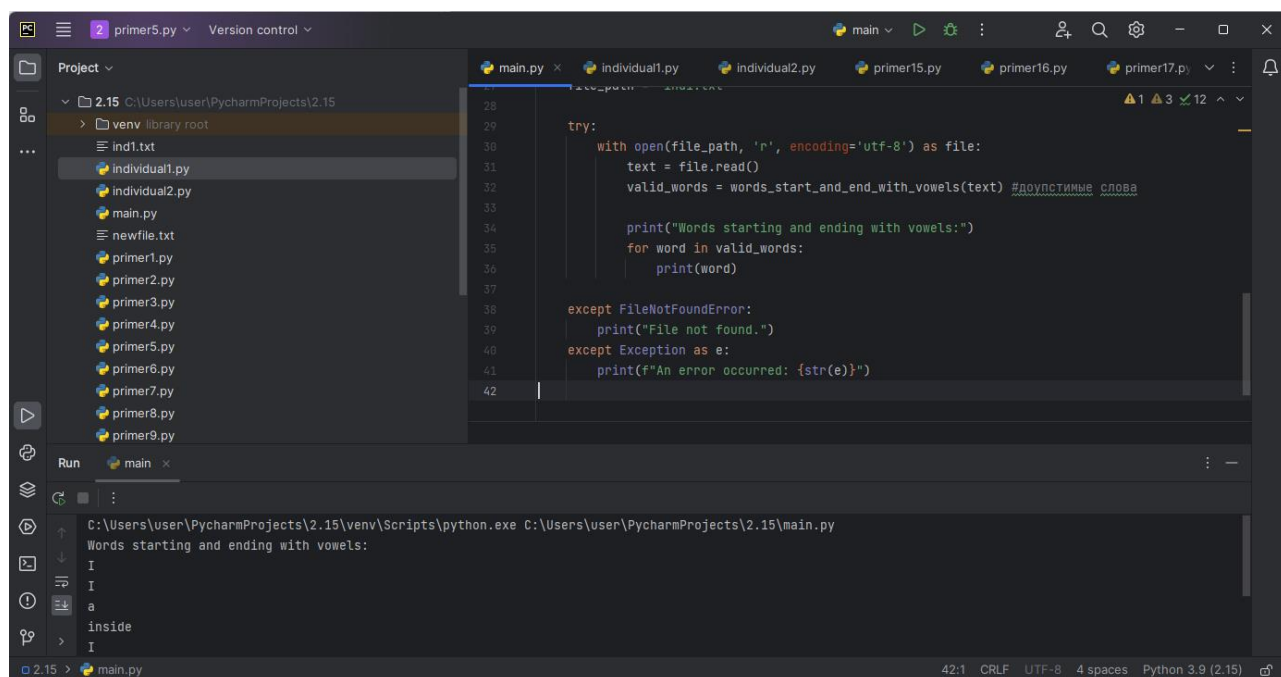


Рисунок 18. Индивидуальное задание 1

Задание 2. Создание пароля посредством генерирования случайных символов может обернуться сложностью в запоминании полученной относительно надежной последовательности. Некоторые системы создания паролей рекомендуют сцеплять вместе два слова на английском языке, тем самым упрощая запоминание заветного ряда символов — правда, в ущерб его надежности. Напишите программу, которая будет открывать файл со списком

слов, случайным образом выбирать два из них и сцеплять вместе для получения итогового пароля. При создании пароля исходите из следующего требования: он должен состоять минимум из восьми символов и максимум из десяти, а каждое из используемых слов должно быть длиной хотя бы в три буквы. Кроме того, сделайте заглавными первые буквы обоих слов, чтобы легко можно было понять, где заканчивается одно и начинается другое. По завершении процесса полученный пароль должен быть отображен на экране.

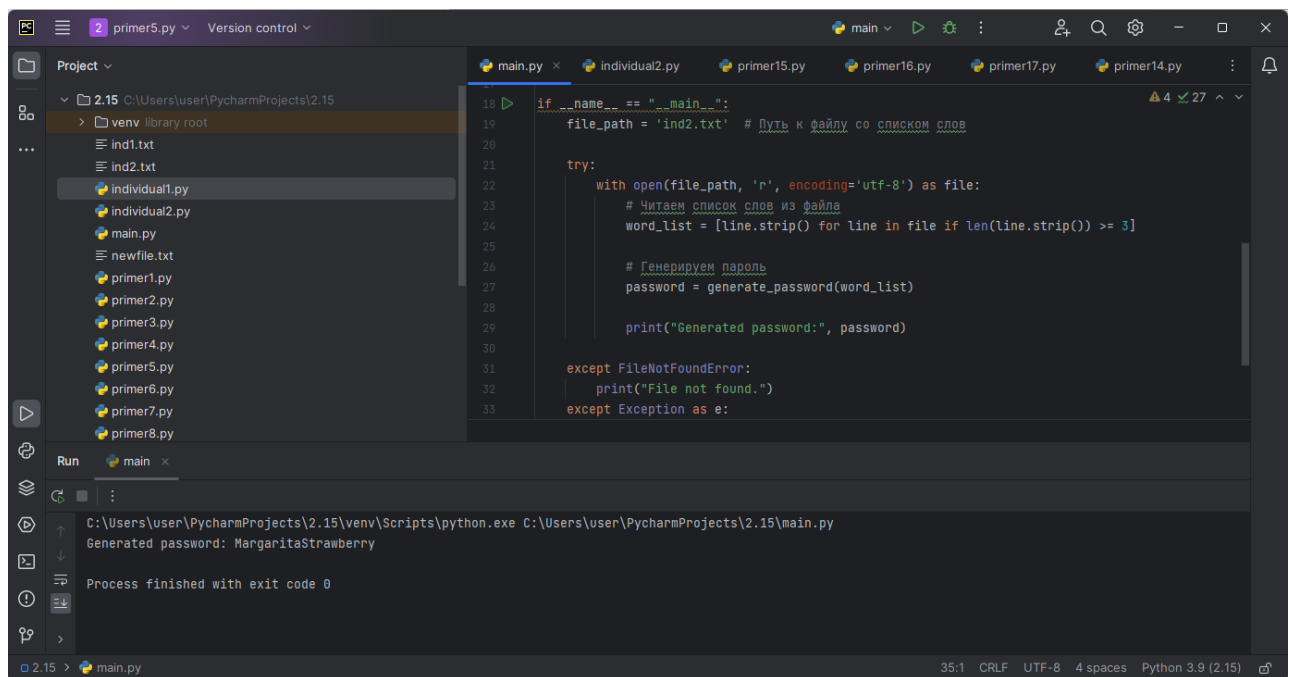


Рисунок 19. Индивидуальное задание 2

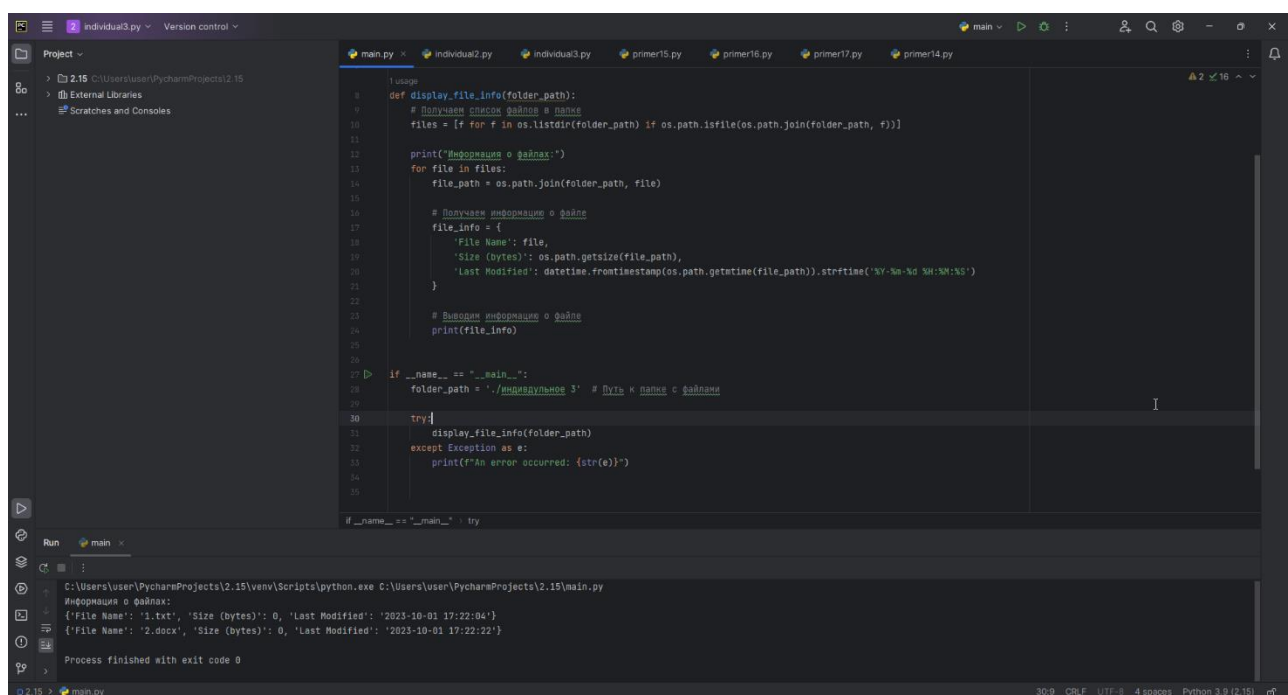
3. Самостоятельно подберите или придумайте задачу для работы с изученными функциями модуля `os`. Приведите решение этой задачи.

4. Задание: Необходимо создать программу, которая перебирает все файлы в указанной папке и выводит на экран информацию о файлах: имя файла, его размер и дата последнего изменения.

3. Решение: Получаем список всех файлов в указанной папке. Для каждого файла получаем информацию о его имени, размере и дате последнего изменения. Выводим эту информацию на экран. `file_path = os.path.join(folder_path, file)`: В этой строке используется функция `os.path.join()`, чтобы создать полный путь к файлу. Функция `os.path.join()` соединяет аргументы в пути с учётом правильного разделителя в зависимости от

операционной системы. `folder_path` содержит путь к основной папке, а `file` - имя файла из списка файлов. Таким образом, `file_path` представляет полный путь к текущему файлу, который перебирается в цикле.

Таким образом, в данном контексте каждая итерация цикла `for` берет имя файла из списка `files`, создает полный путь к этому файлу с помощью `os.path.join()`, и сохраняет этот путь в переменной `file_path`. Это далее используется для получения информации о файле, такой как его размер и дата последнего изменения.



Имя	Дата изменения	Тип	Размер
1	01.10.2023 17:22	Текстовый докум...	0 КБ
2	01.10.2023 17:22	Документ Micros...	0 КБ

Рисунок 18. Индивидуальное задание 3.

## 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.