

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 №1**  
**дисциплины «Анализ данных»**

Выполнил:

Горбунов Данила Евгеньевич  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника»,  
направленность (профиль)  
«Программное обеспечение средств  
вычислительной  
техники и автоматизированных  
систем», очная форма обучения

---

(подпись)

Руководитель практики:

Воронкин Р.А., канд.  
технических наук,  
доцент кафедры  
инфокоммуникаций

---

(подпись)

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

Ставрополь, 2024 г.

**Тема:** Работа с файлами в 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.

### Выполнение работы

1. Создал общедоступный репозиторий на GitHub, в котором использована MIT лицензия и язык программирования Python. Выполнил клонирование созданного репозитория.
2. Дополнил файл .gitignore необходимыми правилами.
3. Организовал созданный репозиторий в соответствии с моделью ветвления git-flow.
4. Проработал примеры лабораторной работы. Создал для них отдельные модули языка Python. Привёл в отчёте скриншоты результата выполнения программ примеров при различных исходных данных, вводимых с клавиатуры.

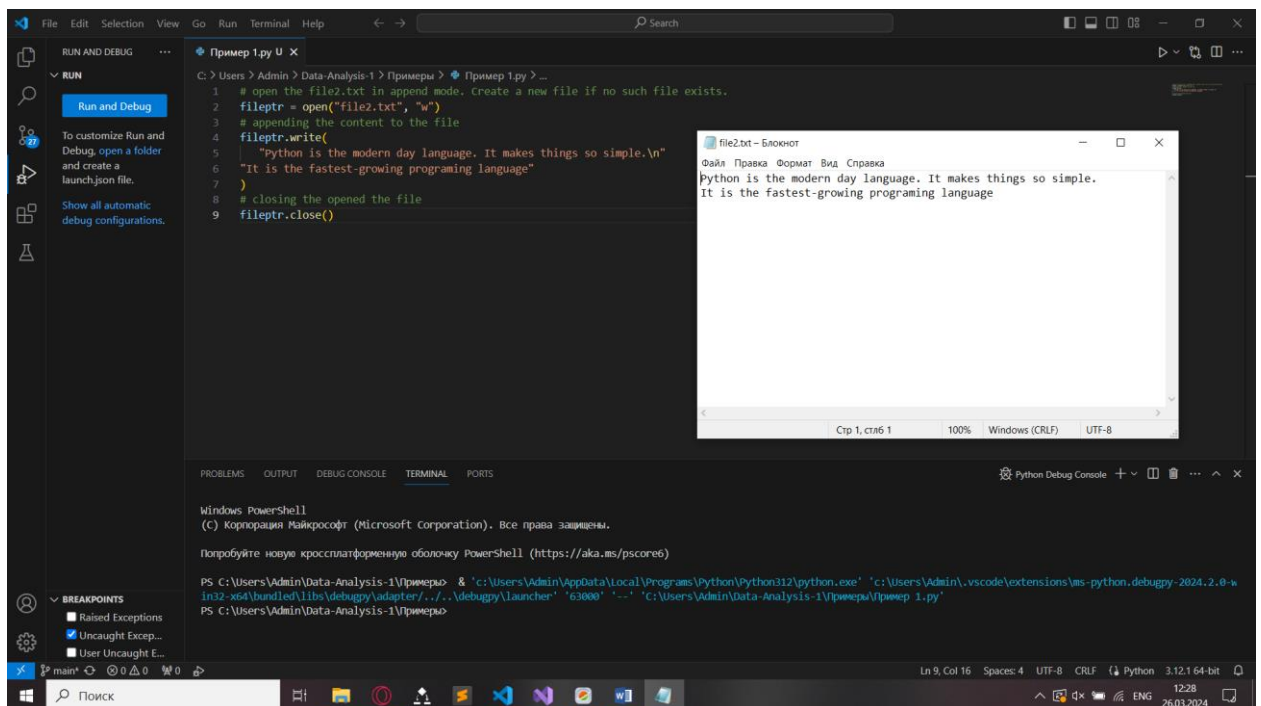


Рисунок 1. Результат работы из примера 1

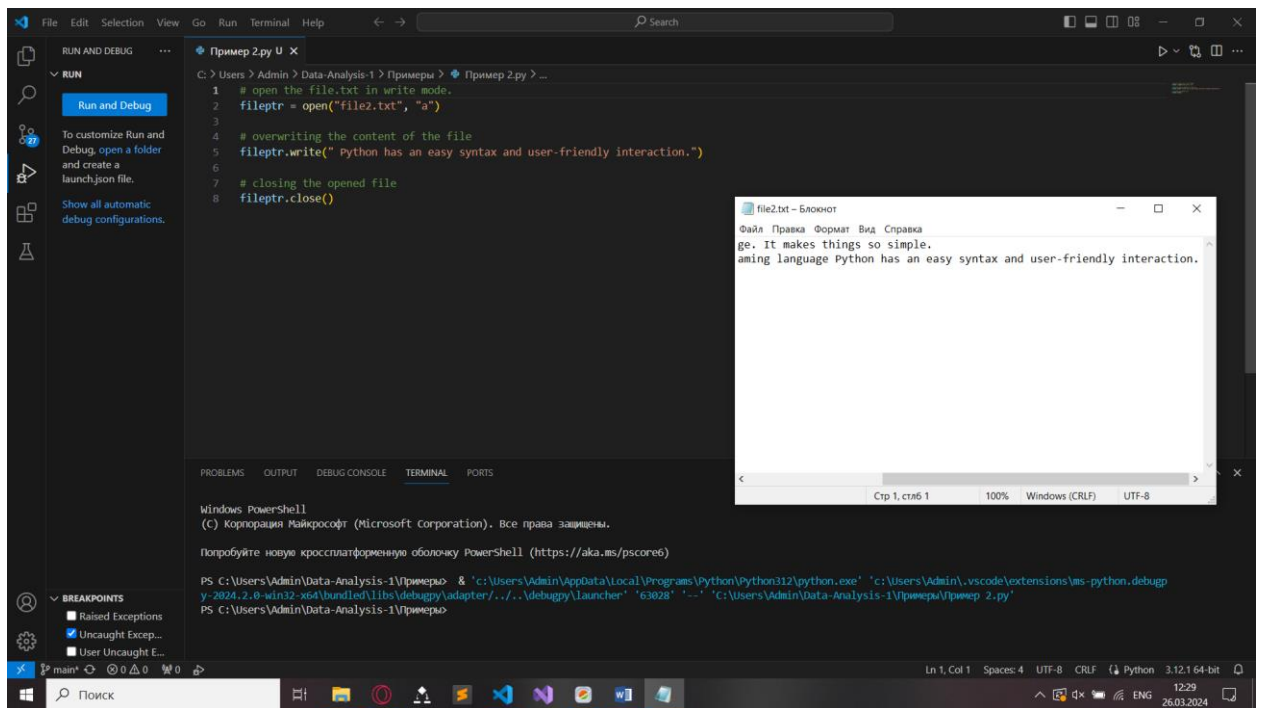


Рисунок 2. Результат работы из примера 2

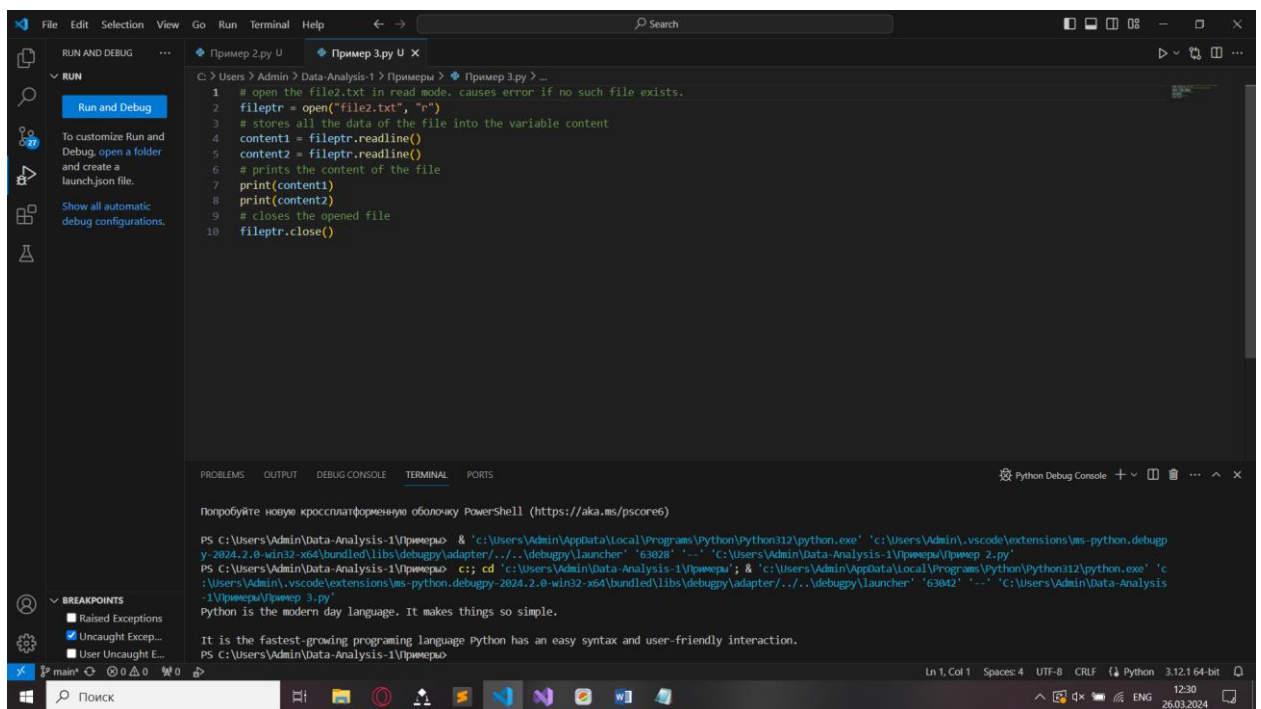


Рисунок 3. Результат работы из примера 3

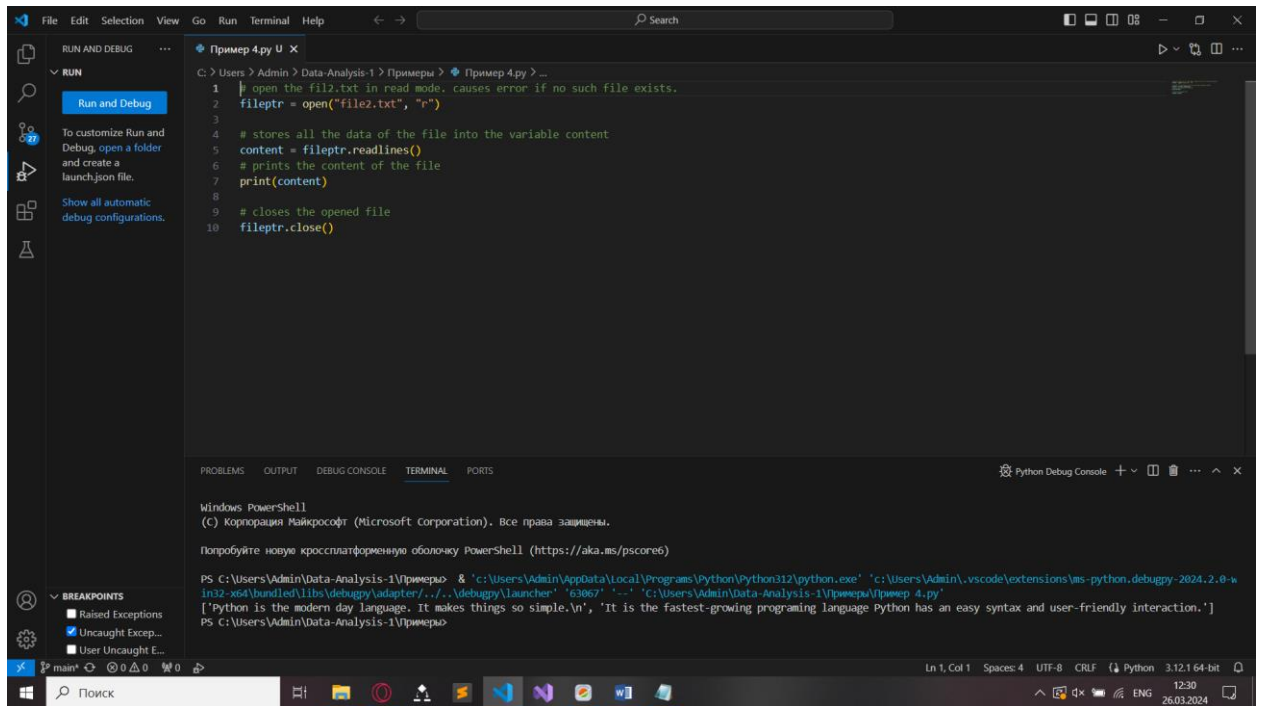


Рисунок 4. Результат работы из примера 4

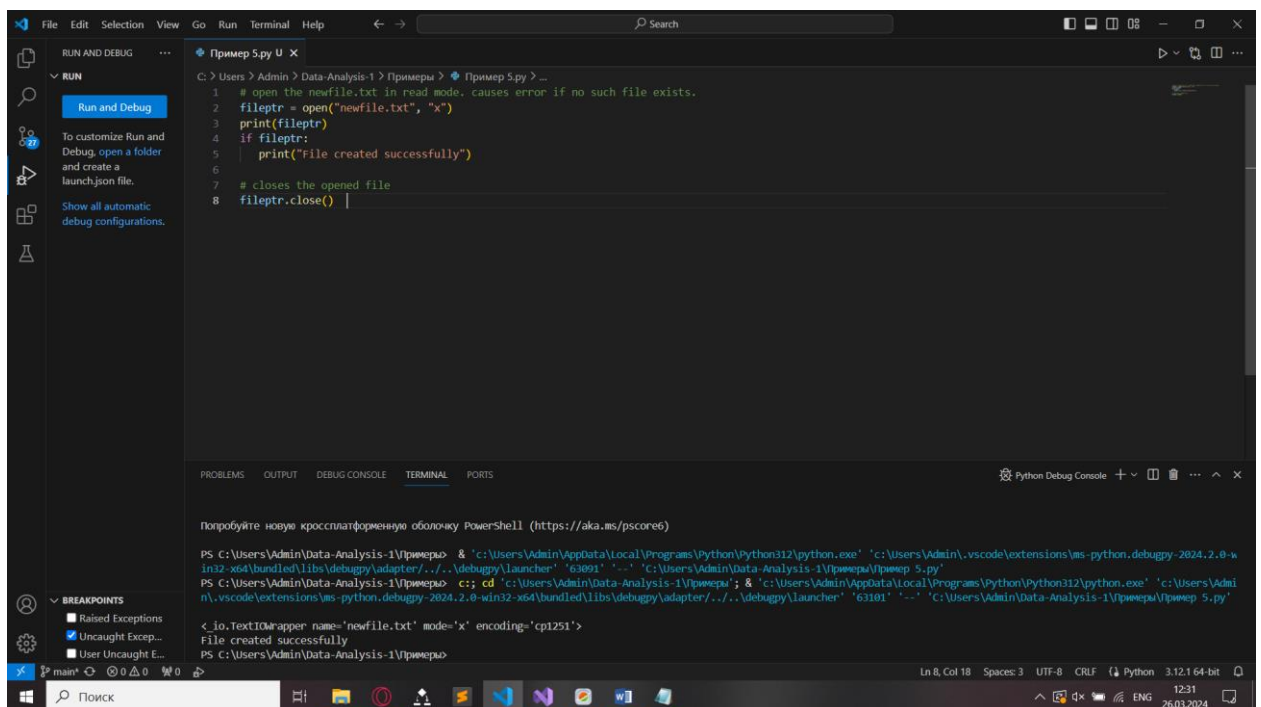


Рисунок 5. Результат работы из примера 5

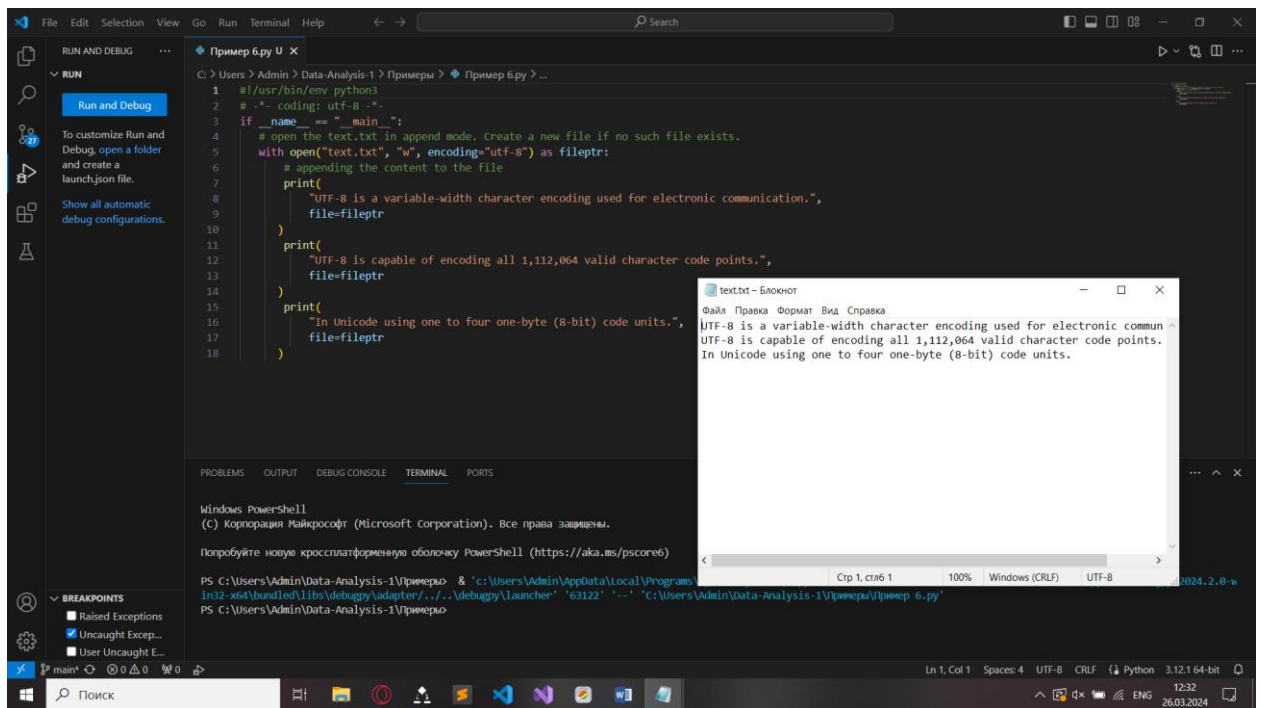


Рисунок 6. Результат работы из примера 6

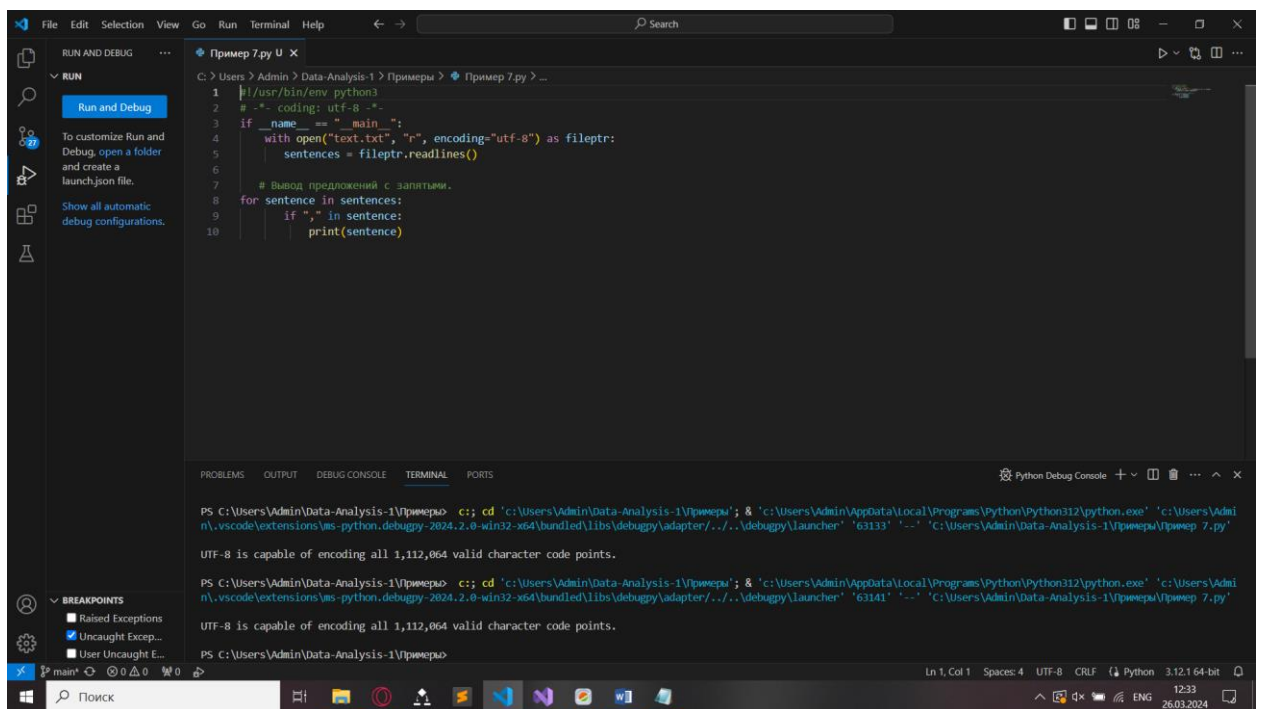


Рисунок 7. Результат работы из примера 7





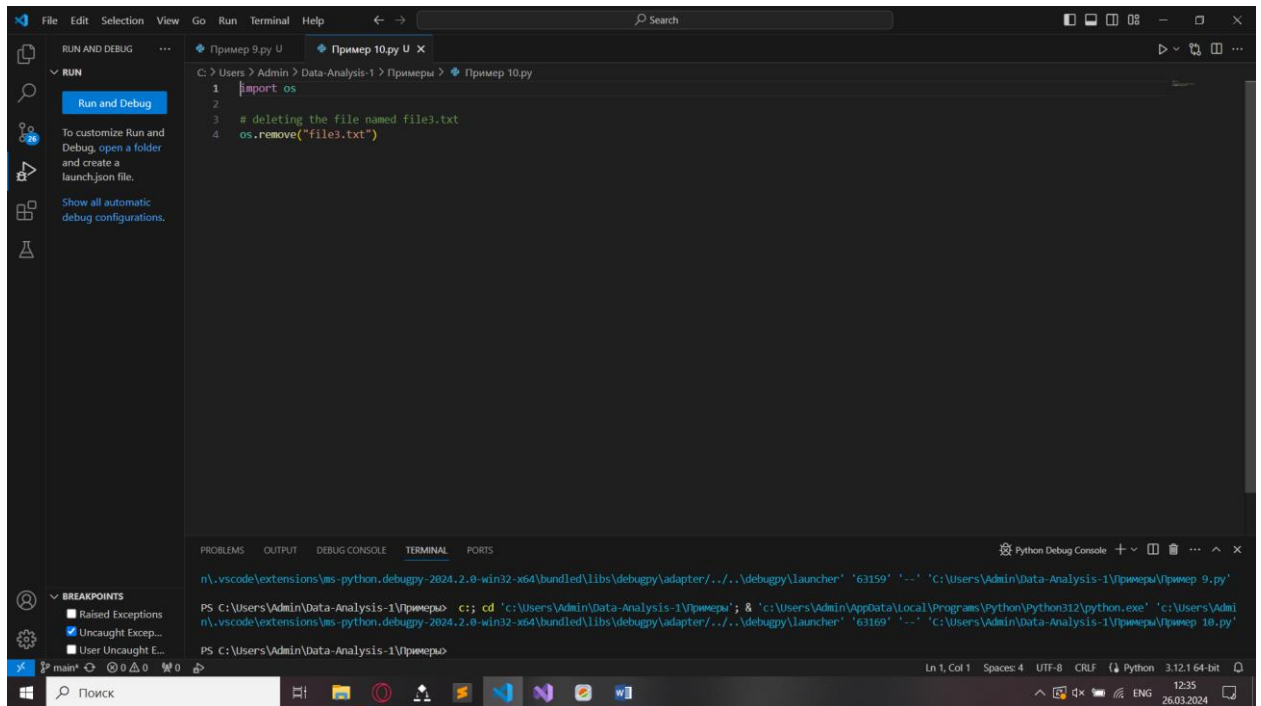


Рисунок 10. Результат работы из примера 10

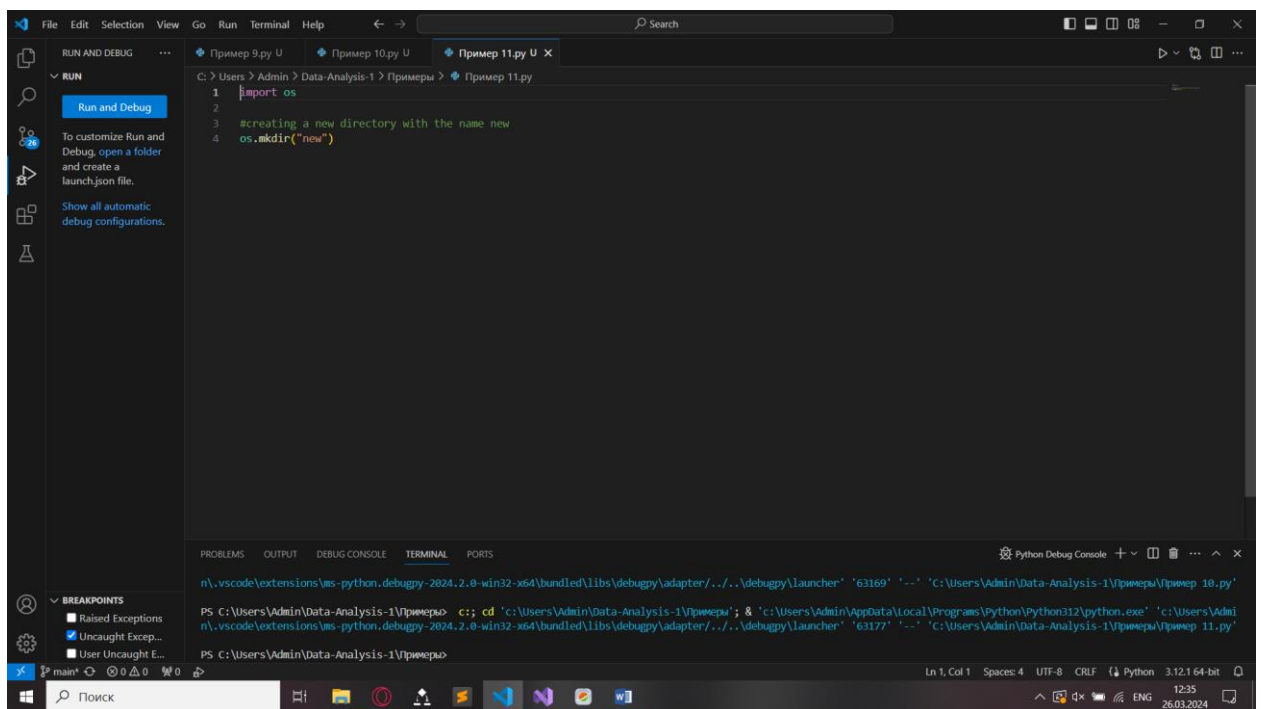


Рисунок 11. Результат работы из примера 11

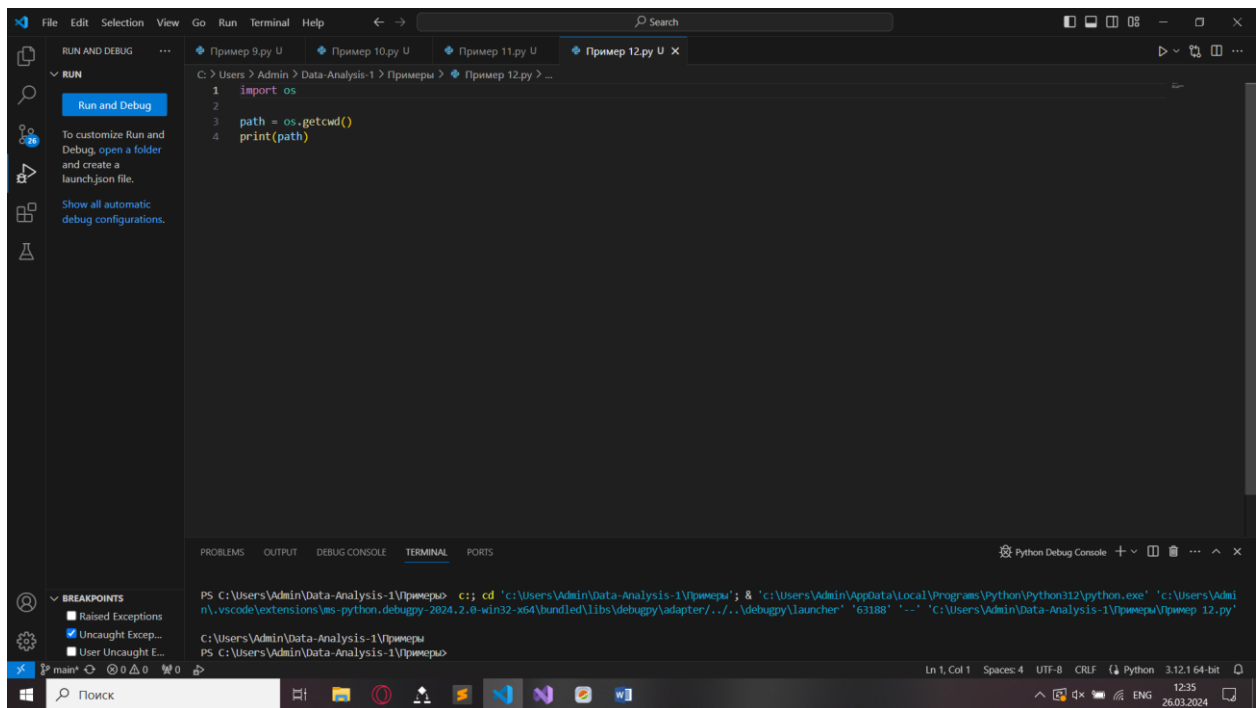


Рисунок 12. Результат работы из примера 12

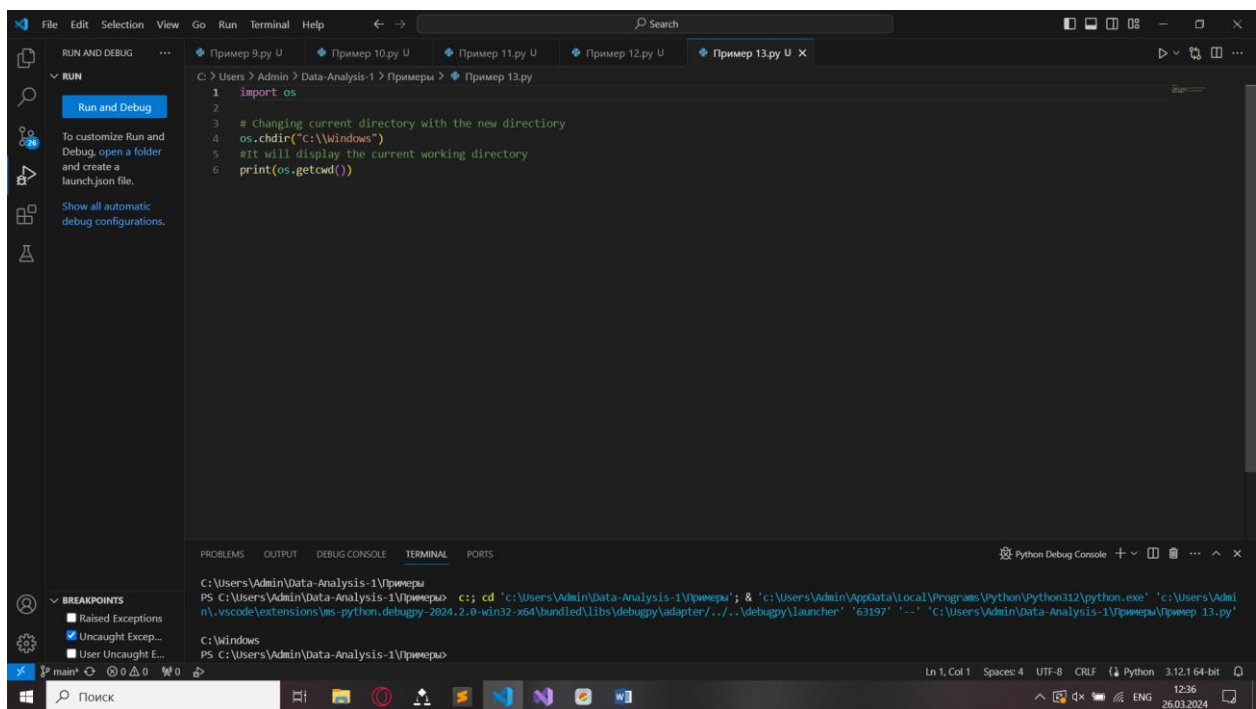


Рисунок 13. Результат работы из примера 13





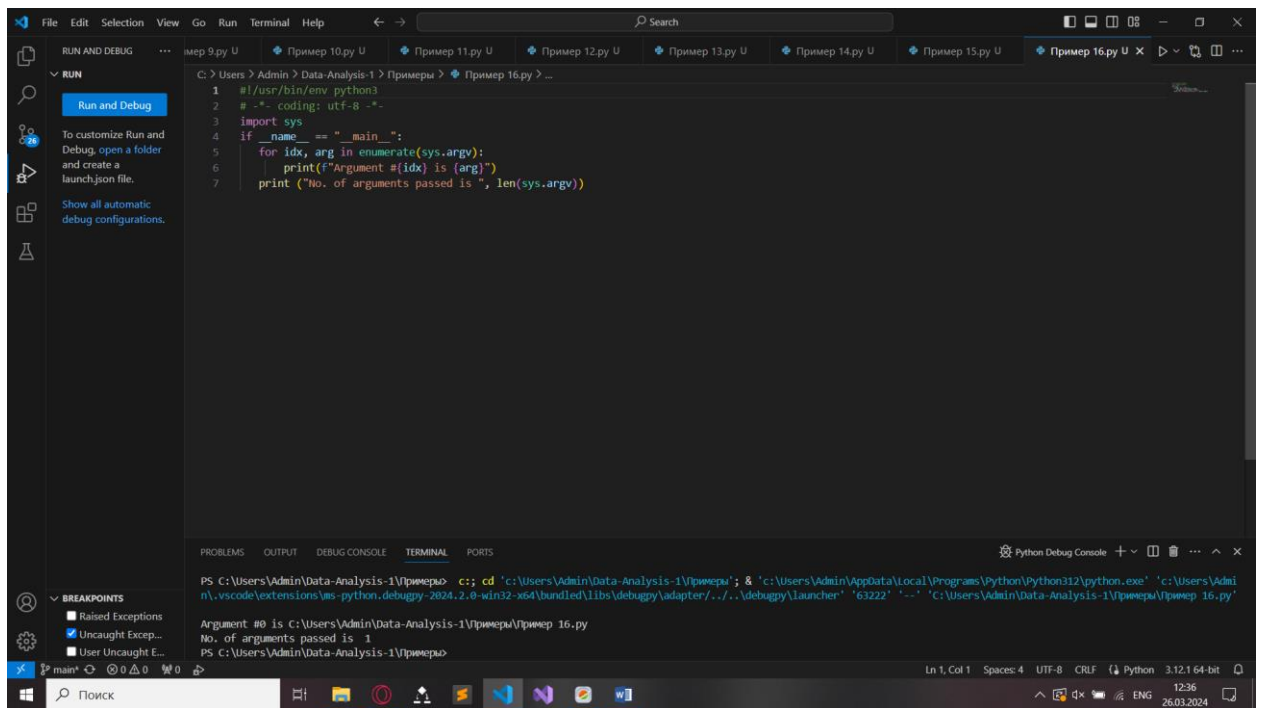


Рисунок 16. Результат работы из примера 16

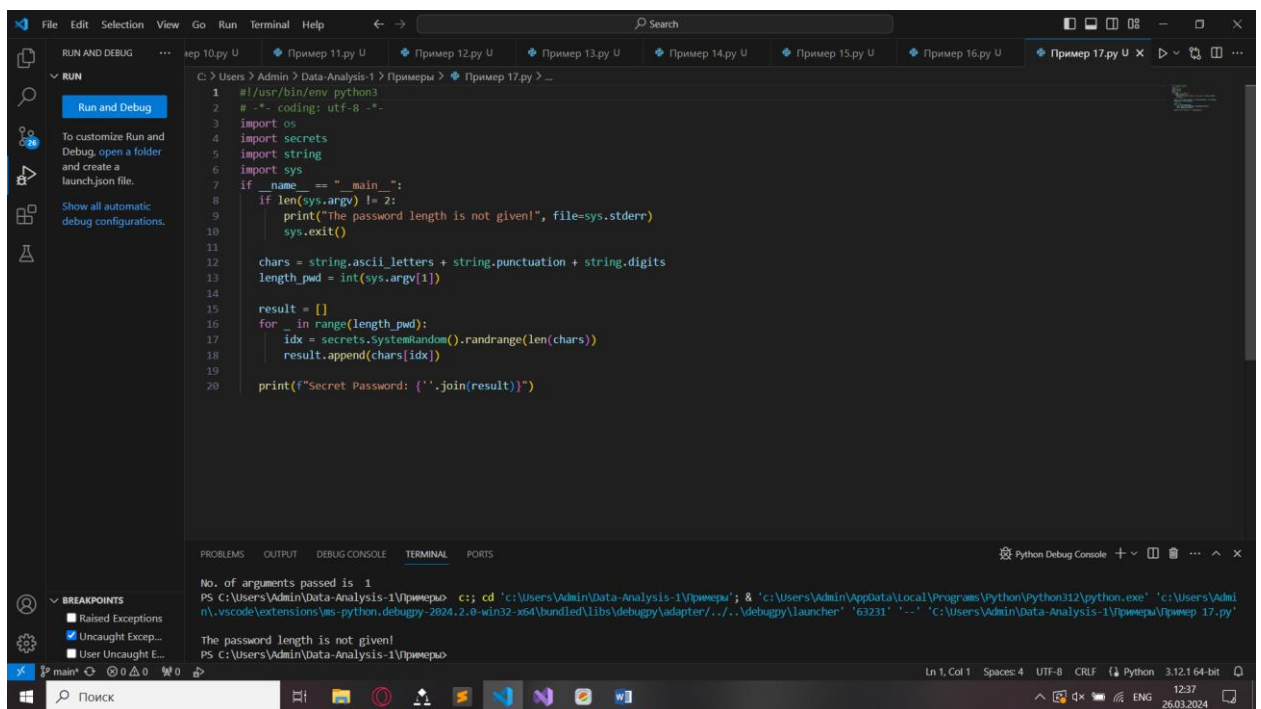


Рисунок 17. Результат работы из примера 17

5. Выполнил индивидуальные задания, согласно варианту 8. Привёл в отчёте скриншоты работы программ.

Задание 1. Написать программу, которая считывает текст из файла и выводит на экран только предложения, не содержащие запятых.

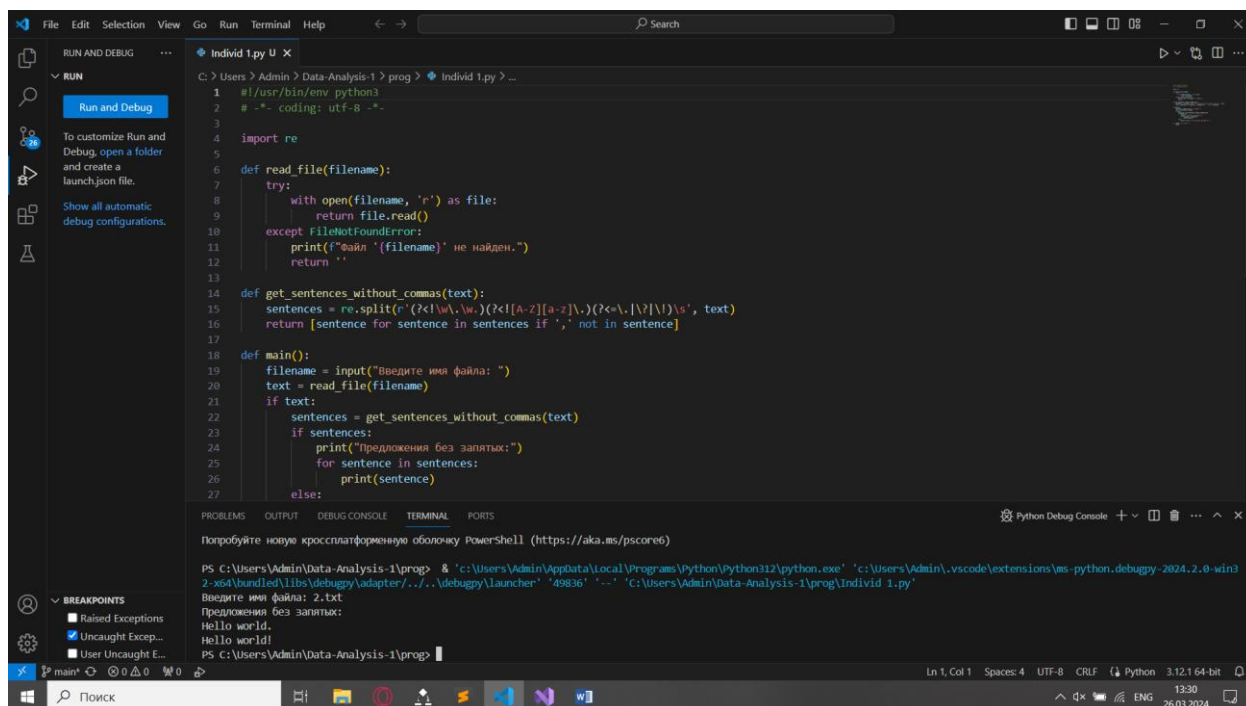


Рисунок 18. Результат работы программы из индивидуального задания 1

Задание 2. Одна из техник декодирования простейших алгоритмов шифрования заключается в применении частотного анализа. Иными словами, вы просто анализируете зашифрованный текст, подсчитывая частоту употребления всех букв. Затем можно использовать операции подстановки для замены наиболее популярных символов на часто используемые в языке буквы (в английском это, например, буквы Е и Т). Напишите программу, которая будет способствовать дешифрации текста путем вывода на экран частоты появления разных букв. При этом пробелы, знаки препинания и цифры должны быть проигнорированы. Также не должен учитываться регистр, то есть символы а и А должны восприниматься как одна буква. Имя файла для анализа пользователь должен передавать программе посредством аргумента командной строки. Если программе не удастся открыть файл для анализа или аргументов командной строки будет слишком много, на экране должно быть отображено соответствующее сообщение об ошибке.

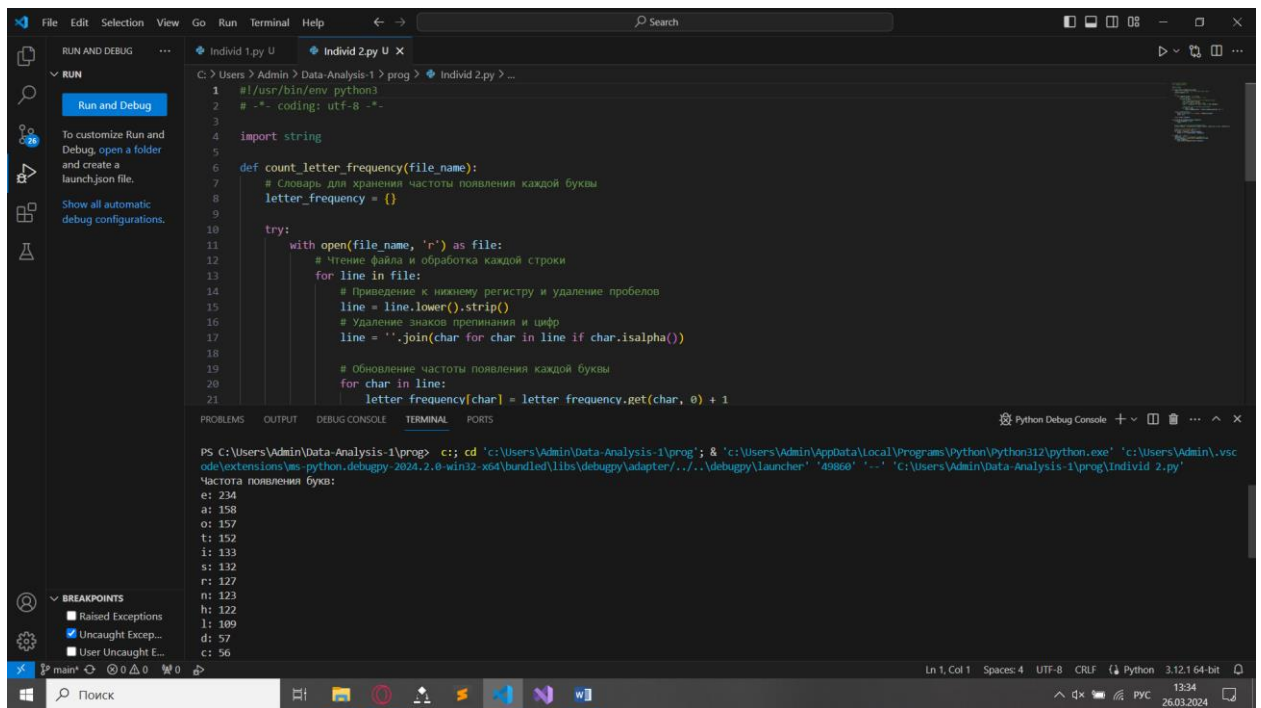


Рисунок 19. Результат работы программы из индивидуального задания 2

### Контрольные вопросы

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

Для открытия файла только для чтения в Python используется функция `open()` с параметром `'r'`. Например: `file = open('file.txt', 'r')`

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

Для открытия файла только для записи в Python используется функция `open()` с параметром `'w'`. Например: `file = open('file.txt', 'w')`

3. Как прочитать данные из файла в языке Python?

Для чтения данных из файла в Python используется метод `read()` или метод `readline()` объекта файла. Например: `data = file.read()` # Чтение всего содержимого файла или `line = file.readline()` # Чтение одной строки из файла

4. Как записать данные в файл в языке Python?

Для записи данных в файл в Python используется метод `write()` объекта файла. Например: `file.write("Some text")` # Запись строки в файл

5. Как закрыть файл в языке Python?

Для закрытия файла в Python используется метод `close()` объекта файла. Например: `file.close()`

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 создания контекста, который автоматически управляет ресурсами. Например, при работе с файлами она автоматически закрывает файл после завершения блока кода. Кроме работы с файлами, она может быть использована для управления подключением к базе данных, сокетами и другими ресурсами.

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

Помимо методов `read()` и `write()`, существуют такие методы для работы с файлами, как `readline()`, `readlines()`, `writelines()` и другие.

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

Помимо рассмотренных функций модуля `os` для работы с файловой системой существует множество других функций. Например:

`os.rename()` - для переименования файла;

`os.remove()` - для удаления файла;

`os.makedirs()` - для создания директорий;

`os.listdir()` - для получения списка файлов в директории;

`os.path.join()` - для объединения путей к файлам и директориям;

и многие другие.

**Вывод:** в ходе данной работы были приобретены навыки по работе с текстовыми файлами при написании программ с помощью языка программирования Python версии 3.x, изучены основные методы модуля `os` для работы с файловой системой, получение аргументов командной строки.