**2023/11/14 00:00|Лекция. Оператор "with"**

## Оператор "with"

Оператор "with" пришел на замену «try finally» для того, чтобы облегчить работу с контекстными менеджерами. Его удобно применять там, где в процессе выполнения задачи, нужно явно выделить начало и конец кода, после которого данный объект будет освобожден. В случае с файлом это отлично подходит, потому что работа с файлами делится на **3 этапа.**

1.**Открыть файл**

2.**Выполнить с ним действие**

3.**Закрыть**

Используя оператор "with", явно можно выделить блоки, где происходит работа с файлами и где его закрытие.

Мы можем весь код, который у нас есть убрать, воспользоваться оператором "with".Сначала идет выражение, затем добавляется «**as**», какой-то объект, куда это все сохраняется и действие, которое им выполняется. Формула выглядит следующим образом. (Рис. 1)



Рис. 1

В нашем случае на место выражения поставим функцию «**open**», откроем файл «**name**», сохраним его как файл, т.е. будем обращаться к открытому файлу по имени, которое укажем после «as». (Рис. 2)

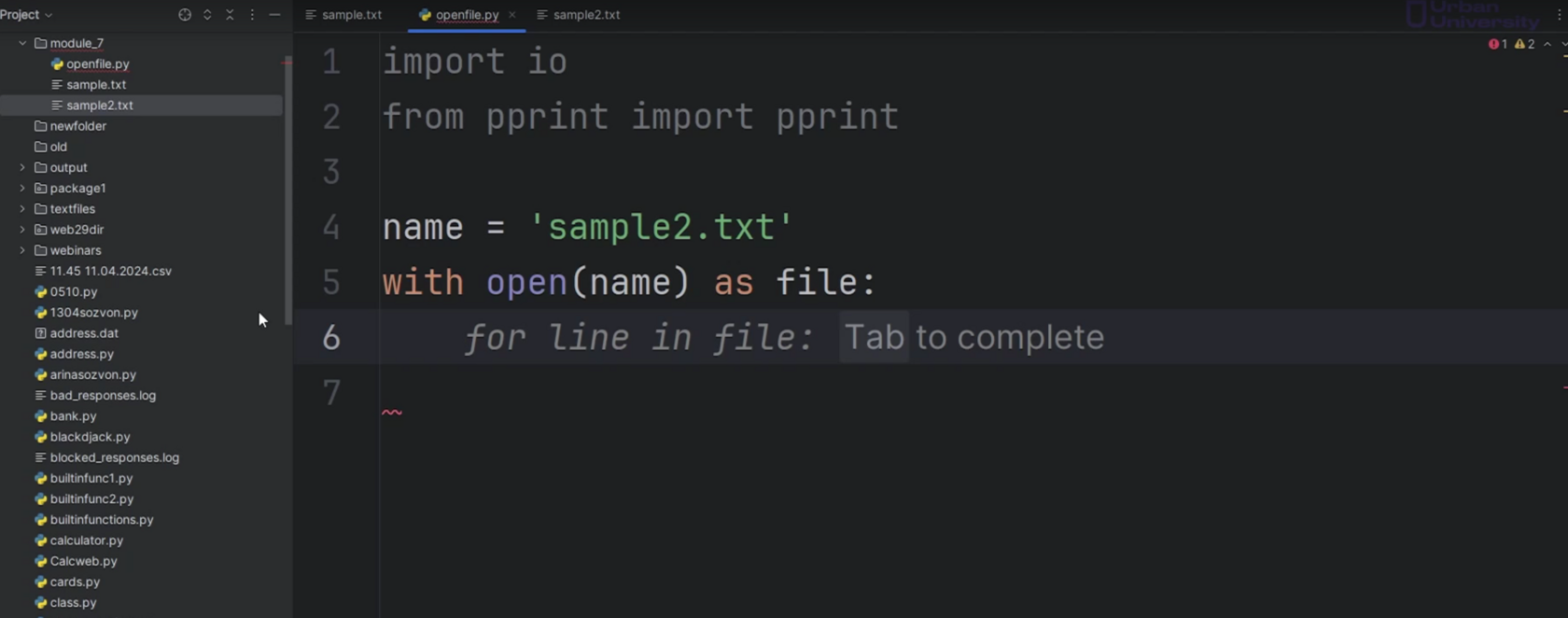


Рис. 2

Дальше хотим выполнить какие-то действия.

Мы можем воспользоваться циклом «**for**» для перебора содержимого нашего файла.

Воспользуемся циклом «for», переберем файл и будем выводить то, что будет находиться в этой переменной. Запустим. (Рис. 3)

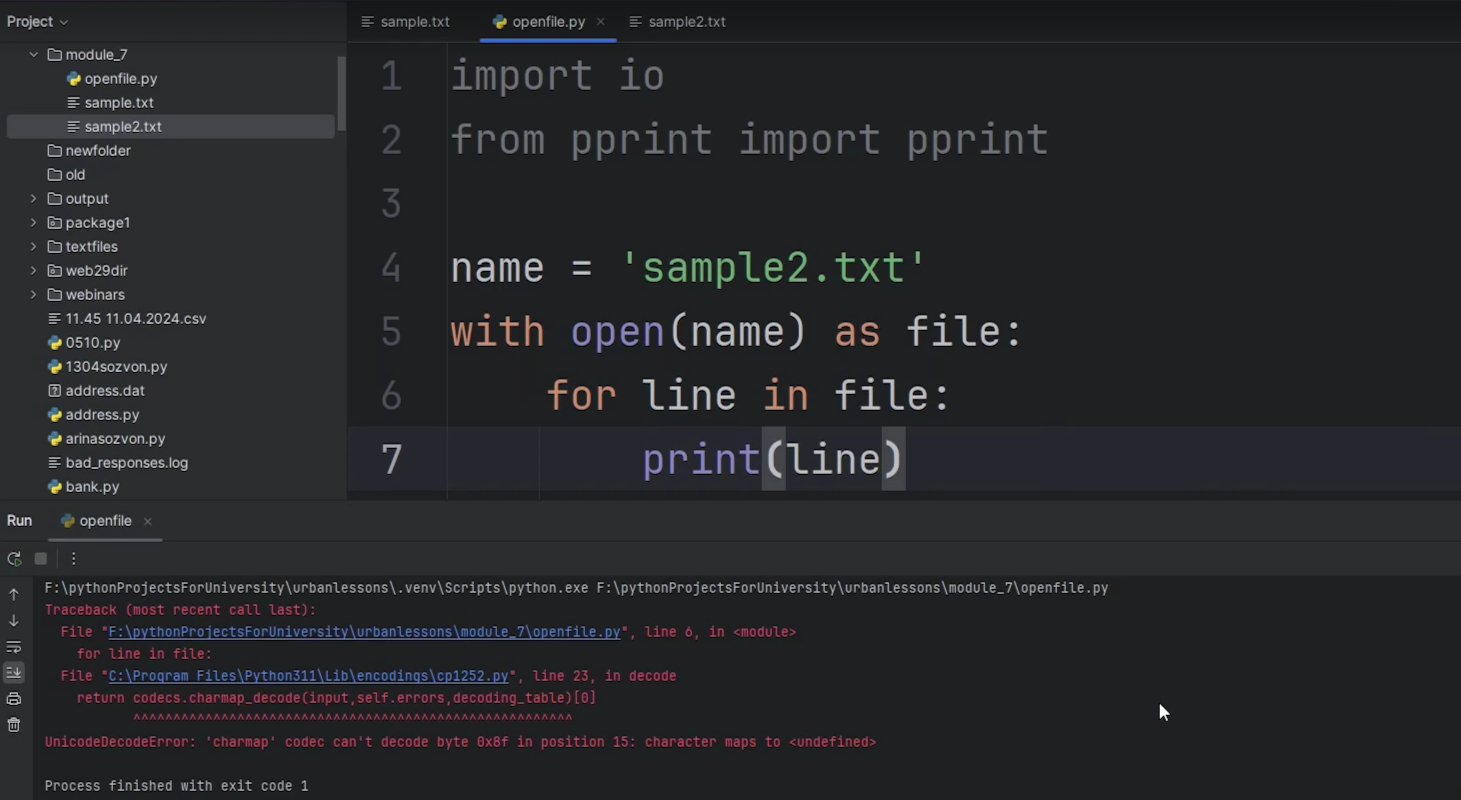


Рис. 3

Получаем ошибку «**DecodeError**». Добавим параметр «**encoding**» и зададим значение «**utf-8**». Запускаем. (рис. 4)

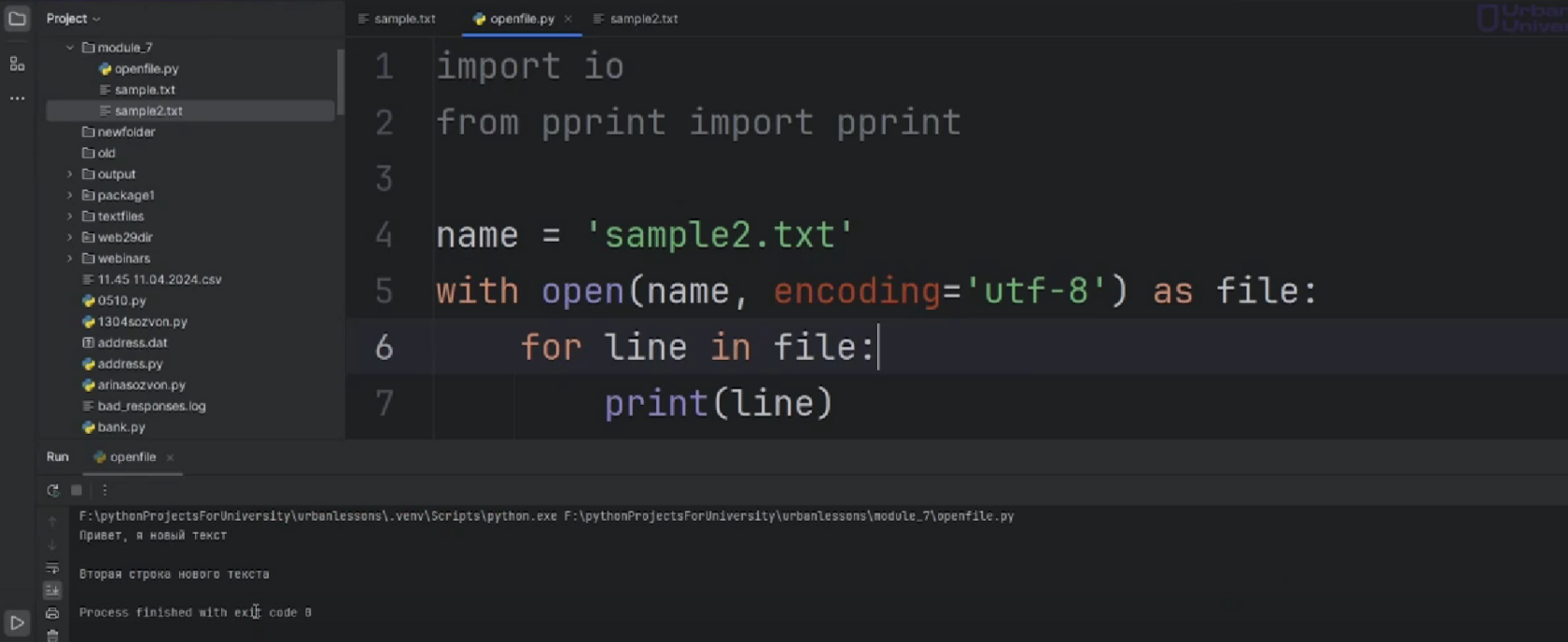


Рис. 4

Видим **расстояние между файлами.** Дело в том, что в файле в конце каждой строки есть символ переноса на новую строку и «print» переносит запись на следующую строку. Чтобы не было «слипания» можно добавить параметр «**end**» и сказать, что в качестве разделителя, т.е. в конце принта не будет стоять символ переноса на другую строку. (Рис. 5)

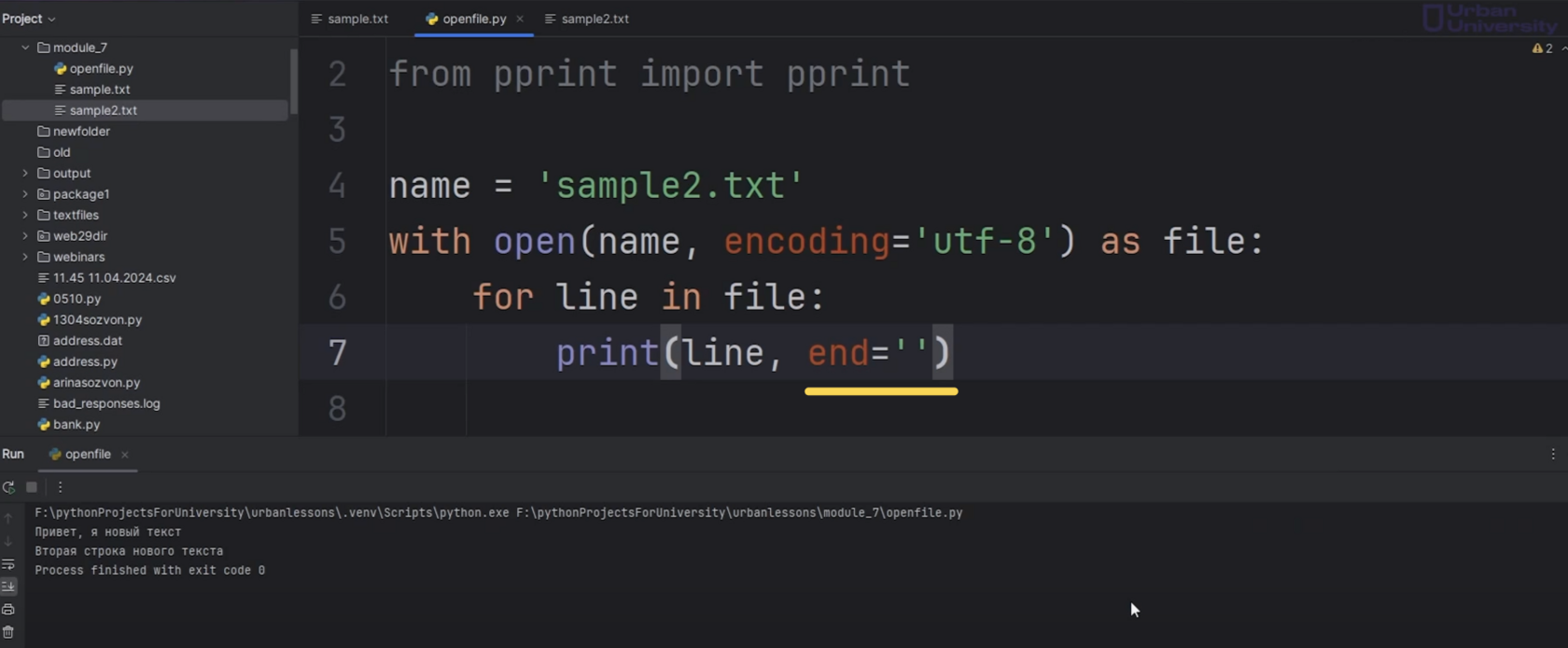


Рис. 5

Если хочется пробежаться по каждому символу в файле, можно добавить еще 1 цикл «for», в котором можно перебирать переменную. (Рис. 6)



Рис. 6

При запуске результат будет точно таким. Если убрать параметр «end», то получим по отдельности каждый символ. (Рис. 7)

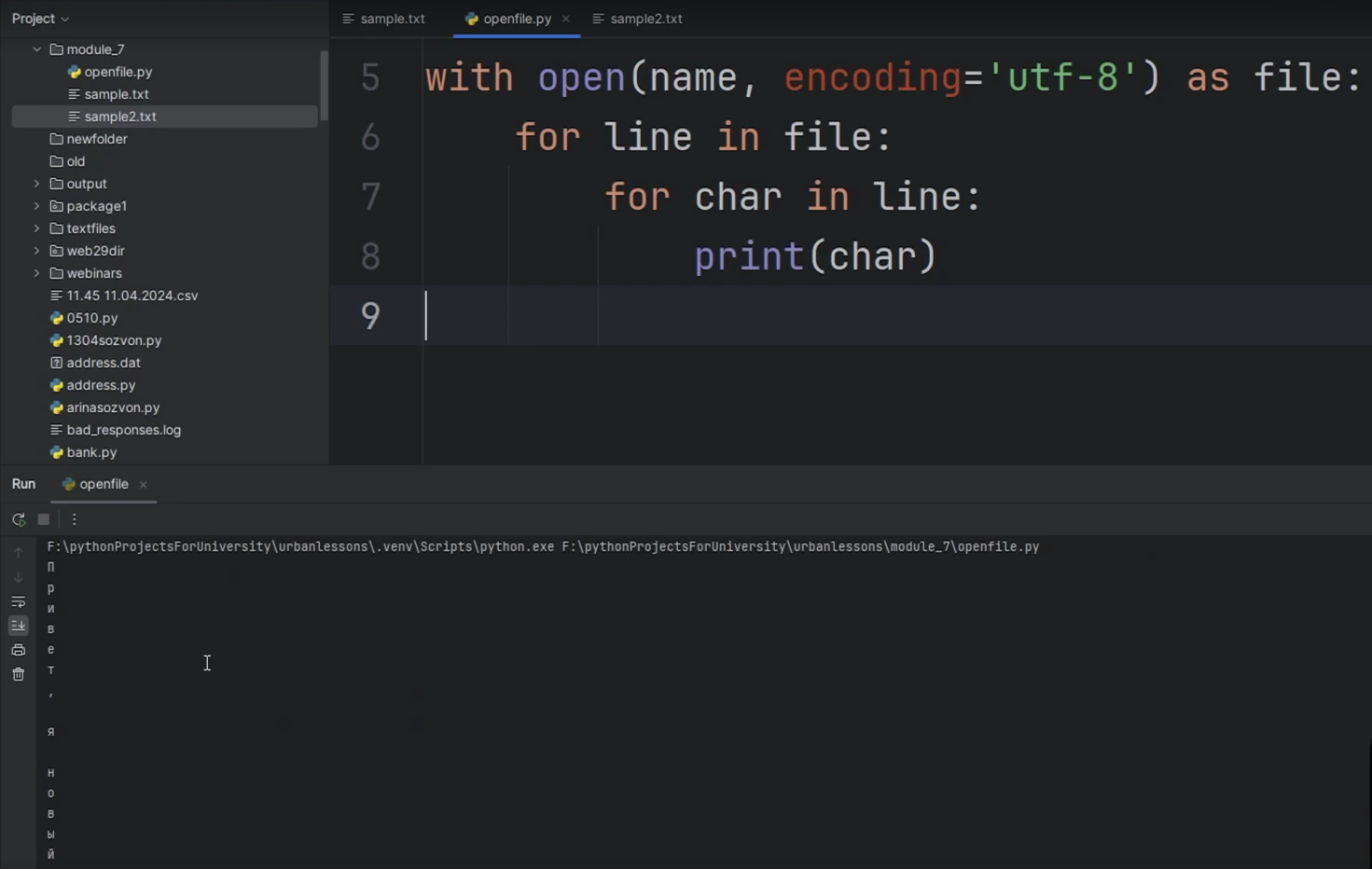


Рис. 7

Таким образом мы считываем содержимое файла. Если где-то нужно посмотреть, например, воспользоваться методом «tell» чтобы посмотреть на курсор, то выходит 91, потому что мы пробежались по файлу и каждому символу. (Рис. 8)

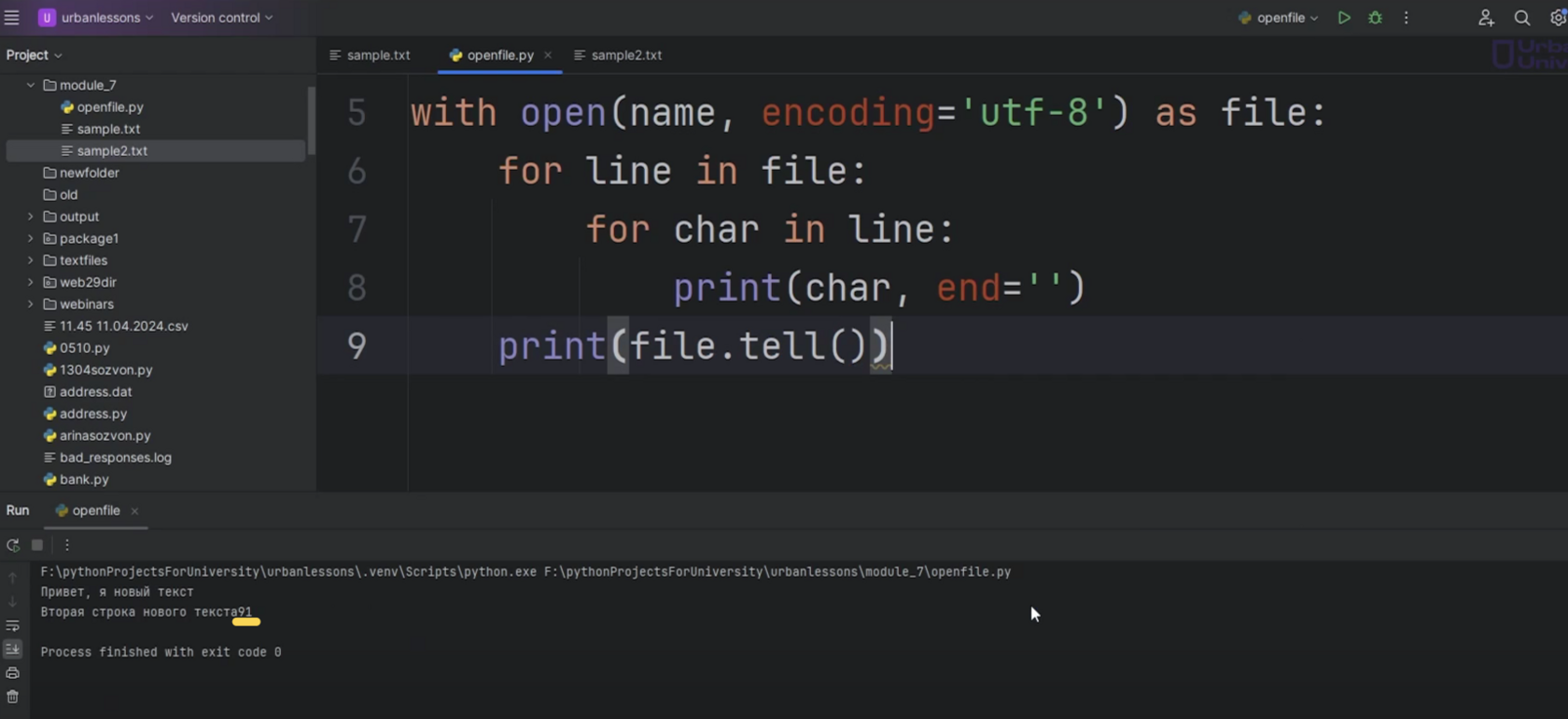


Рис. 8

Как только мы пробежались по каждому символу, курсор передвинулся в конец.

### Когда закрывать файл

Весь код, который будет идти без отступа, будет происходить после закрытия файла. "with" дает гарантию того, что все что выполняется с файлом, выполняется с отступом и принадлежит оператору "with". Как только действие с отступами заканчивается, файл автоматически закрывается. Если после действий над файлом, нужно взять и считать содержимое, то получим ошибку, т.к. это операция над закрытым файлом. (Рис. 9)

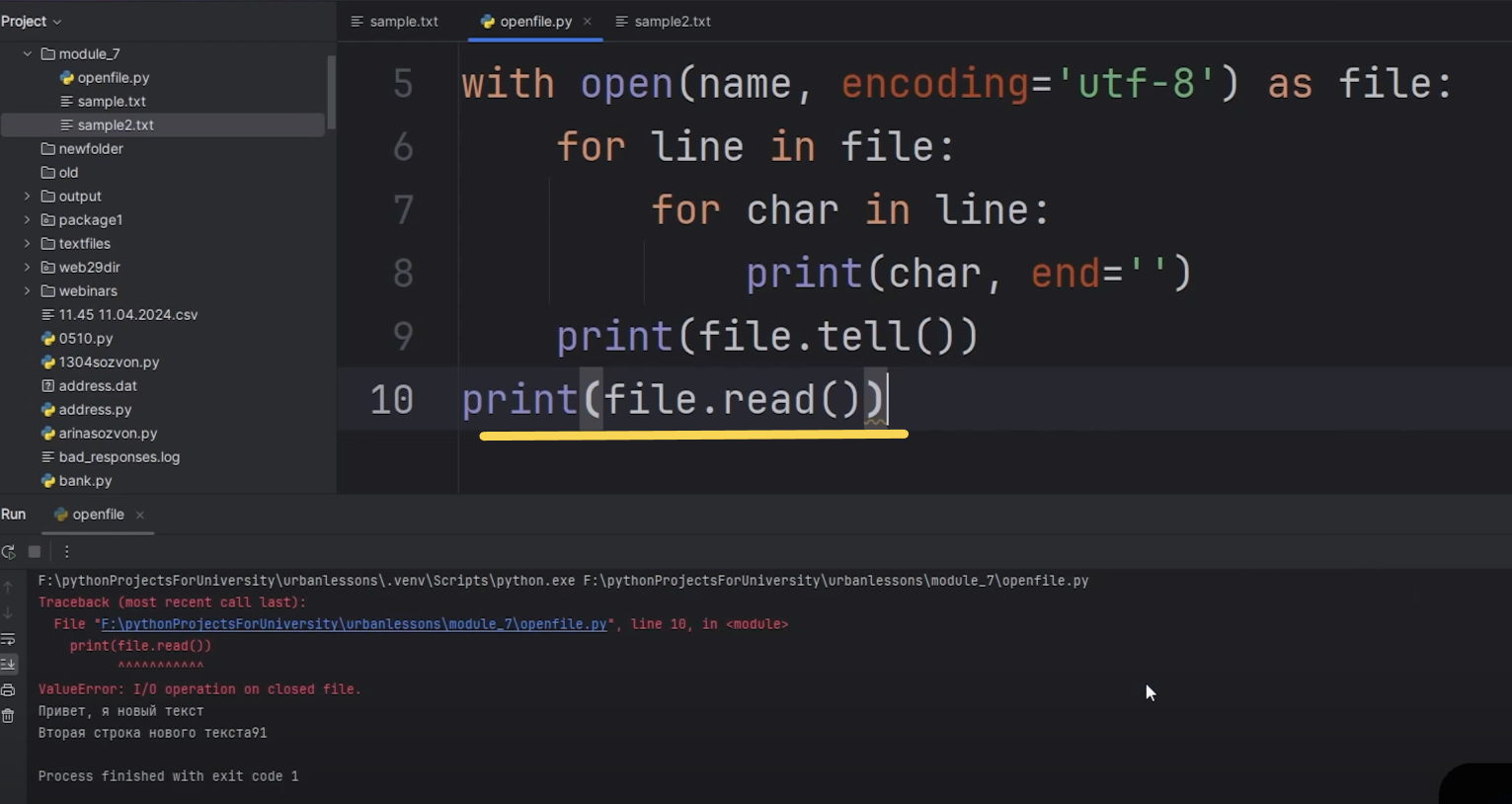


Рис. 9