

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Отчет по лабораторной работе № 2.15

Декораторы функций в языке Python

Выполнил студент группы ИВТ-б-о-20-1

Ищенко М.А.

Работа защищена « » _____ 20__ г.

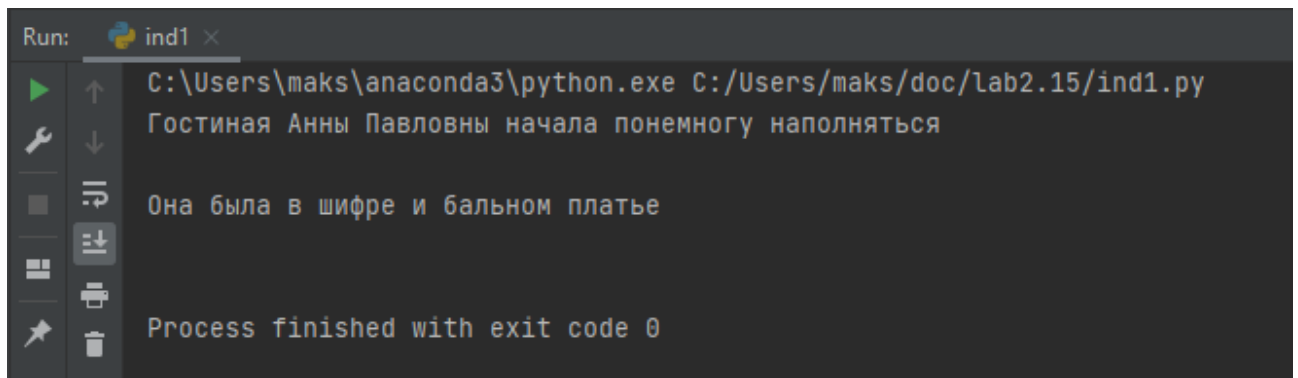
Проверил(а) _____

Ставрополь 2021

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.

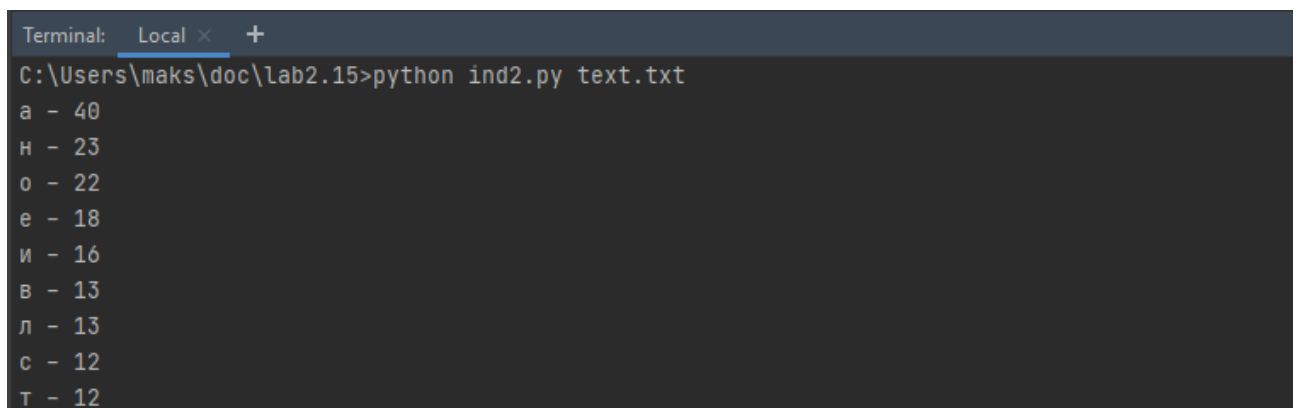
Создан общедоступный репозиторий на GitHub. Дополнен файл .gitignore необходимыми правилами для работы с IDE PyCharm.

Выполнено индивидуальные задания 1 и 2 варианта 6, рис. 1-2



```
Run: ind1 x
C:\Users\maks\anaconda3\python.exe C:/Users/maks/doc/lab2.15/ind1.py
Гостинная Анны Павловны начала понемногу наполняться
Она была в шифре и бальном платье
Process finished with exit code 0
```

Рисунок 1 – Индивидуальное задание 1



```
Terminal: Local x +
C:\Users\maks\doc\lab2.15>python ind2.py text.txt
а - 40
н - 23
о - 22
е - 18
и - 16
в - 13
л - 13
с - 12
т - 12
```

Рисунок 2 – Индивидуальное задание 2

Придумано и решено задание, рис. 3-5

Суть задания: “Напишите программу, использующую модуль os, которая будет подменять время изменения файла, переданного командной строке”

Данные файла до изменения представлены на рисунке 3

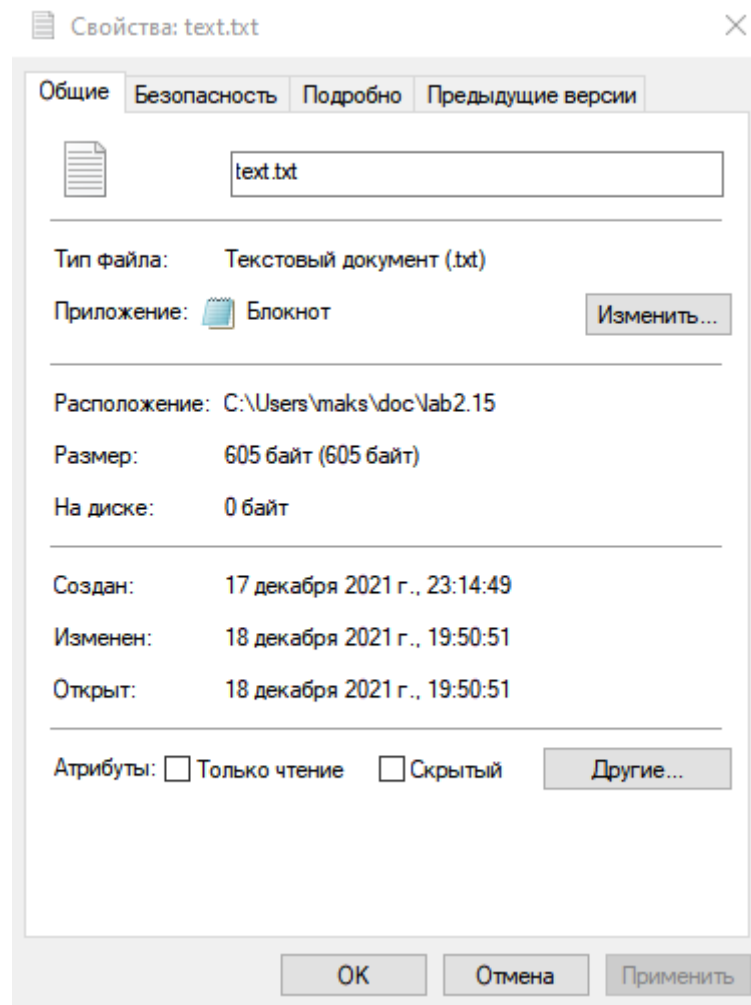


Рисунок 3 – Данные файла до изменения

Работа программы на рисунке 4

```
Terminal: Local x +
C:\Users\maks\doc\lab2.15>python ind3.py text.txt
Старое время: Sat Dec 18 19:51:25 2021 Sat Dec 18 19:50:51 2021
Введите разницу (в секундах): 3600
Новое время: Sat Dec 18 18:51:25 2021 Sat Dec 18 18:50:51 2021
C:\Users\maks\doc\lab2.15>
```

Рисунок 4 – Работа программы

Данные файла после изменения представлены на рисунке 5

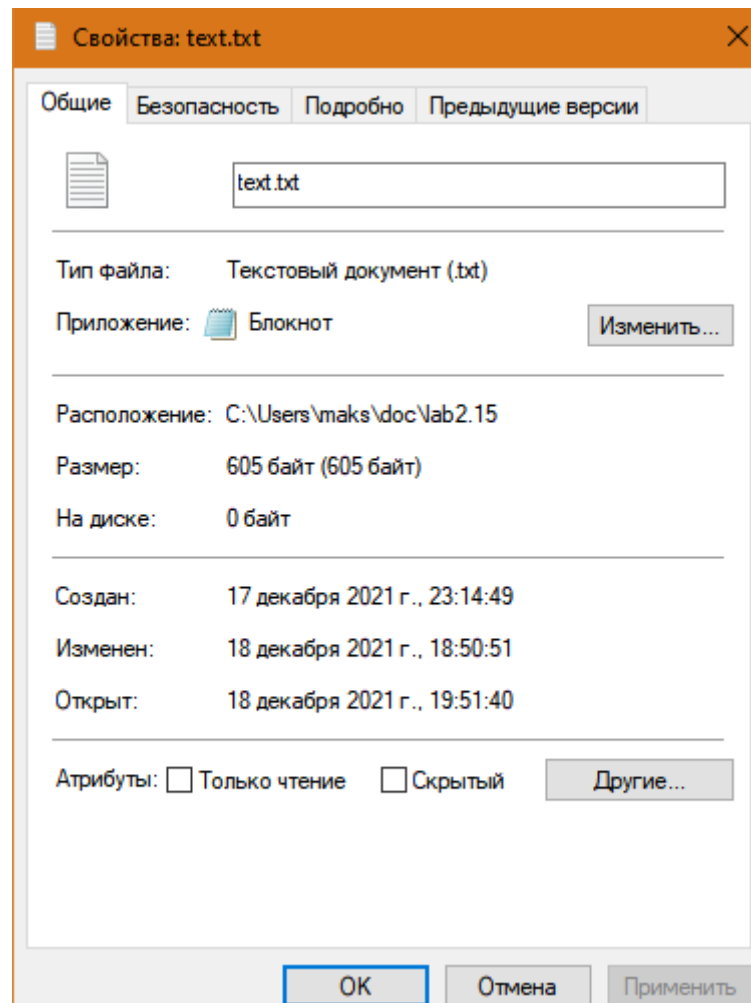


Рисунок 5 – Данные файла после изменения

Программы проверены на flake8, рис. 6

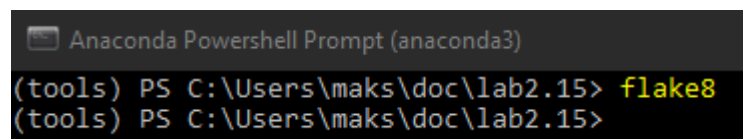


Рисунок 6 – Проверка заданий

Контрольные вопросы:

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

Используя функцию `open()`, после ввода имени файла через запятую указать режим “r”.

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

Используя функцию `open()`, после ввода имени файла через запятую указать режим “w”.

3. Как прочитать данные из файла в языке Python?

Сначала необходимо открыть файл, вызвав функцию `open()`, затем использовать метод `read()`.

4. Как записать данные в файл в языке Python?

Сначала необходимо открыть файл, вызвав функцию `open()`, затем использовать метод `write()`.

5. Как закрыть файл в языке Python?

Использовать метод `close()` или открывать файл при помощи оператора `with`, который закрывает файл, после окончания работы с ним

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` гарантирует, что критические функции выполнятся в любом случае. В основном она используется для работы с файлами разного типа, но также может использоваться для фиксации или отката транзакции базы данных, для перенаправления стандартного вывода однопоточных программ.

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

Метод `writelines()` – добавляет последовательность строк в файл.

Метод `tell()` - возвращает текущую позицию “условного курсора” в файле.

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

`os.name` - имя операционной системы

`os.environ` - словарь переменных окружения.

`os.getpid()` - текущий `id` процесса.

`os.uname()` - информация об ОС.

`os.access ()` - проверка доступа к объекту у текущего пользователя.

`os.chdir (path)` - смена текущей директории.

`os.chmod ()` - смена прав доступа к объекту.
`os.link ()` - создаёт жёсткую ссылку.

`os.listdir ()` - список файлов и директорий в папке.

`os.makedirs ()` - создаёт директорию, создавая при этом промежуточные директории.

`os.symlink ()` - создаёт символическую ссылку на объект.

`os.truncate ()` - обрезает файл до длины `length`.

`os.utime ()` - модификация времени последнего доступа и изменения файла.

`os.walk ()` - генерация имён файлов в дереве каталогов.

`os.system ()` - исполняет системную команду, возвращает код её завершения.

`os.urandom (n)` - `n` случайных байт.

`os.path` - модуль, реализующий некоторые полезные функции на работы с путями

Вывод: в ходе занятия 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.