

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

Кафедра инфокоммуникаций

Отчет по лабораторной работе №2.15

Работа с файлами в языке Python

по дисциплине «Технологии программирования и алгоритмизации»

Выполнил студент группы ИВТ-б-о-20-1

Хашиев Х.М. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____

(подпись)

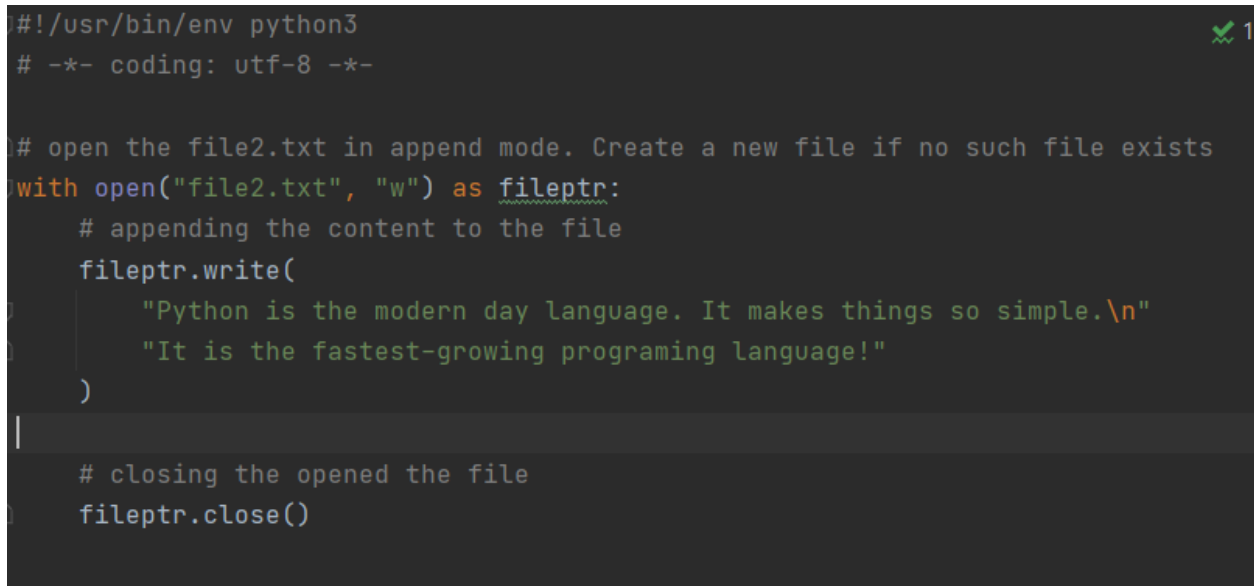
Ставрополь 2021

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.

Ход работы: Примеры

<https://github.com/Mirror-Shard/L2.15>

1. Изучил теоретический материал и приступил к выполнению примеров:

A screenshot of a code editor with a dark background. The code is a Python script that opens a file named 'file2.txt' in write mode ('w'). It then writes two lines of text to the file: 'Python is the modern day language. It makes things so simple.\n' and 'It is the fastest-growing programming language!'. The file is then closed. The code is as follows:

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

# open the file2.txt in append mode. Create a new file if no such file exists
with open("file2.txt", "w") as fileptr:
    # appending the content to the file
    fileptr.write(
        "Python is the modern day language. It makes things so simple.\n"
        "It is the fastest-growing programming language!"
    )

# closing the opened the file
fileptr.close()
```

Рисунок 1 – Запись файла

```

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

# open the file2.txt in write mode.
with open("file2.txt", "a") as fileptr:
    # overwriting the content of the file
    fileptr.write(" Python has an easy syntax and user-friendly interaction.")

# open the file2.txt in read mode. causes error if no such file exists.
with open("file2.txt", "r") as fileptr:
    # stores all the data of the file into the variable content
    content = fileptr.read(10)
    # prints the type of the data stored in the file
    print(type(content))
    # prints the content of the file
    print(content)

# open the file2.txt in read mode. causes error if no such file exists.
with open("file2.txt", "r") as fileptr:
    # running a for loop
    for i in fileptr:
        print(i) # i contains each line of the file

```

Рисунок 2 – Чтение файла

```

1 ▶ 1 #!/usr/bin/env python3
2   2 # -*- coding: utf-8 -*-
3   3
4   4 # open the file2.txt in read mode. causes error if no such file exists.
5   5 with open("file2.txt", "r") as fileptr:
6   6     # stores all the data of the file into the variable content
7   7     content1 = fileptr.readline()
8   8     content2 = fileptr.readline()
9   9     # prints the content of the file
10  10     print(content1)
11  11     print(content2)
12  12

```

Рисунок 3 – Чтение строк при помощи readline()

```

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)
10

```

Рисунок 4 – Чтение строк с помощью readlines()

```

1 ▶ #!/usr/bin/env python3
2   # -*- coding: utf-8 -*-
3
4   # open the newfile.txt in read mode. causes error if no such file exists.
5   with open("newfile.txt", "x") as fileptr:
6       print(fileptr)
7       if fileptr:
8           print("File created successfully")
9

```

Рисунок 5 – Создание нового файла

```

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("UTF-8 is a variable-width character encoding used for electronic "
9               "communication.", file=fileptr)
10        print("UTF-8 is capable of encoding all 1,112,064 valid character"
11              " code points.", file=fileptr)
12        print("In Unicode using one to four one-byte (8-bit) code units.", file=fileptr)
13

```

Рисунок 6 – Изменение кодировки файла

```

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         # Вывод предложений с запятыми.
8         for sentence in sentences:
9             if "," in sentence:
10                 print(sentence)
11

```

Рисунок 7 – Вывод предложений, содержащих запятые

```

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
9     # changing the file pointer location to 10.
10    fileptr.seek(10)
11
12    # tell() returns the location of the fileptr.
13    print("After reading, the filepointer is at:", fileptr.tell())
14

```

Рисунок 8 – Позиция указателя файла

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import os
5
6
7 # rename file2.txt to file3.txt
8 os.rename("file2.txt", "file3.txt")
9
```

Рисунок 9 – Переименование файла

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import os
5
6
7 # deleting the file named file3.txt
8 os.remove("file3.txt")
9
```

Рисунок 10 – Удаление файла

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import os
5
6
7 # creating a new directory with the name new
8 os.mkdir("new")
9
```

Рисунок 11 – Создание новой директории

```

1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import os
5
6
7 path = os.getcwd()
8 print(path)
9

```

Рисунок 12 – Получение пути текущего рабочего каталога

```

1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import os
5
6
7 # Changing current directory with the new directory
8 os.chdir("C:\\Windows")
9 # It will display the current working directory
10 print(os.getcwd())
11

```

Рисунок 13 – Изменение текущего рабочего каталога

```

1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import os
5
6
7 # removing the new directory
8 os.rmdir("new")
9

```

Рисунок 14 – Удаление каталога

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

Рисунок 15 – Подсчёт аргументов командной строки

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 import sys
4
5 ▶ if __name__ == "__main__":
6     for idx, arg in enumerate(sys.argv):
7         print(f"Argument #{idx} is {arg}")
8     print("No. of arguments passed is ", len(sys.argv))
9
```

Рисунок 16 – Иной способ подсчёта аргументов командной строки


```

1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 import ...
6
7
8
9
10 ▶ if __name__ == "__main__":
11     if len(sys.argv) != 2:
12         print("The password length is not given!", file=sys.stderr)
13         sys.exit(1)
14
15     chars = string.ascii_letters + string.punctuation + string.digits
16     length_pwd = int(sys.argv[1])
17
18     result = []
19     for _ in range(length_pwd):
20         idx = secrets.SystemRandom().randrange(len(chars))
21         result.append(chars[idx])
22
23     print(f"Secret Password: {''.join(result)}")
24

```

Рисунок 17 – Программа по созданию пароля

Задание 1

Создайте простой файловый менеджер, который может создавать, удалять, перемещать, переименовывать и совершать другие операции с файлами и папками.

1. Написал код для файлового менеджера, который использует модуль `os` для совершения всех операций с файлами и изменением текущего рабочего каталога на указанный:

```

Для работы с файлами вводите: *команда* *имя_файла*
Список команд – cd, ls, getcwd
mkdir, mkfile, remove, rmdir, rename
>>> mkdir folder
>>> ls
['folder', 'task 1.py']
>>> cd folder
C:\Users\1\Desktop\Алгоритмизация\Lab10\L2.15\tasks\folder
>>> mkfile text.txt
>>> ls
['text.txt']
>>> cd ..
C:\Users\1\Desktop\Алгоритмизация\Lab10\L2.15\tasks
>>> rename folder
Введите новое имя: new_folder

```

Рисунок 18 – Работа файлового менеджера

Индивидуальное задание 1

Вариант 10

Написать программу, которая считывает текст из файла и выводит на экран только строки не содержащие двухзначных чисел.

1. Создал файл text.txt вписал туда 3 строки, где только в 1-й находится двухзначное число:

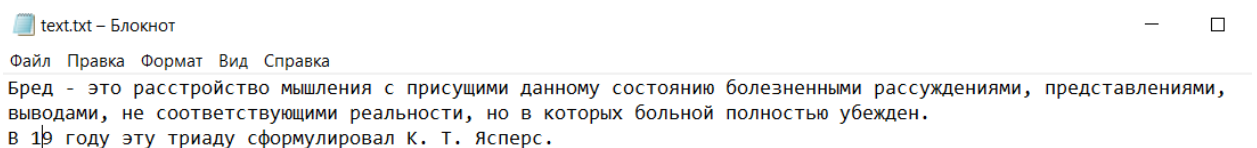


Рисунок 19 – Текстовый файл

2. Написал программу, которая считывает текст, делит его на строки и слова, и вычисляет присутствие двухзначных чисел:

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

if __name__ == "__main__":
    # Открытие файла
    with open("text.txt", 'r', encoding="utf-8") as file:
        content = file.readlines()

        # Проходит по строкам
        for line in content:
            words = line.split()
            # Проходит по словам
            for word in words:
                # Печатает строку, если в ней 2-х значное число
                if word.isdigit() and len(word) == 2:
                    print(line)

if __name__ == "__main__":
    ind 1 x
    C:\Users\1\AppData\Local\Programs\Python\Python39\python.exe "C:/Use
    В 19 году эту триаду сформулировал К. Т. Ясперс.
```

Рисунок 20 – Работа программы

Индивидуальное задание 2

Вариант 10

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.

1. Написал программу, которая запрашивает имя файла и создаёт новый файл в который переписываются строки первого файла которые не начинаются на #.

```
11 ► if __name__ == "__main__":
12
13     # Запрос имени файла
14     file_name = input("Введите имя файла, который нужно открыть: ")
15
16     # Разбивает на строки первый файл
17     with open(file_name, "r") as file:
18         content = file.readlines()
19
20     # Запрос имени для нового файла
21     new_file_name = input("Введите имя для нового файла: ")
22
23     # Создаёт новый файл
24     with open(new_file_name, "w") as new_file:
25         # Вписывает строки которые не начинаются с #
26         for line in content:
27             if not line.startswith('#'):
28                 new_file.write(line)
29
```

Рисунок 21 – Код программы

 code.py – Блокнот

Файл Правка Формат Вид Справка


```
import sys

# Ввести кортеж одной строкой
A = tuple(map(int, input().split()))
# Проверить количество элементов кортежа
if len(A) != 10:
    print("Неверный размер кортежа", file=sys.stderr)
    exit(1)

# Найти искомую сумму
s = sum(a for a in A if abs(a) < 5)

print(s)
```

Рисунок 22 – Первый файл

 new code.txt – Блокнот

Файл Правка Формат Вид Справка

```
import sys

A = tuple(map(int, input().split()))
if len(A) != 10:
    print("Неверный размер кортежа", file=sys.stderr)
    exit(1)

s = sum(a for a in A if abs(a) < 5)

print(s)
```

Рисунок 23 – Новый файл

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.

Вывод: в ходе работы приобрёл навыки по работе с текстовыми файлами при написании программ с помощью языка программирования Python версии 3.