

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»
Вариант 29

Выполнил:
Саенко Андрей Максимович
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, 2023 г.

Тема: Работа с файлами в языке Python

Цель: 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.

Порядок выполнения работы:

Задание 1. Проработаны примеры лабораторной работы

Пример 1. Запись файла

Код программы:

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

if __name__ == '__main__':
    # open the file2.txt in append mode. Create a new file if no such file exists.
    fileptr = open("C:/Users/HAIER/Desktop/Задания/Анализ данных/Data_lr_1/code/file2.txt",
"w")
    # 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 programing language"
    )
    # closing the opened the file
    fileptr.close()
```

Результат работы программы:

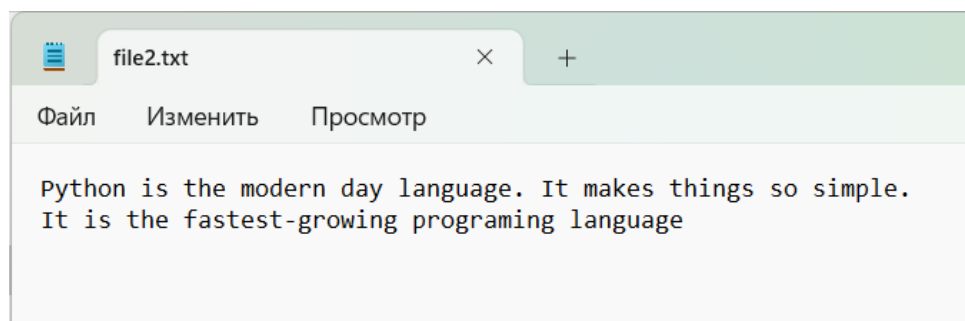


Рисунок 1 – Результат работы программы

Второй способ:

Код программы:

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

if __name__ == '__main__':
    # open the file2.txt in write mode.
```

```
with open("C:/Users/HAIER/Desktop/Задания/Анализ данных/Data_lr_1/code/file2.txt",
"a") as fileptr:
    # overwriting the content of the file
    fileptr.write(" Python has an easy syntax and user-friendly interaction.")
```

Результат работы программы:

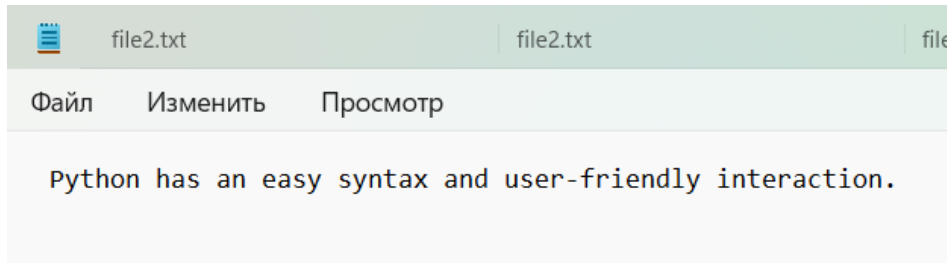


Рисунок 2 – Результат работы программы

Пример 2. Чтение из файла

Код программы:

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

if __name__ == '__main__':
    # open the file2.txt in read mode. causes error if no such file exists.
    fileptr = open("C:/Users/HAIER/Desktop/Задания/Анализ данных/Data_lr_1/code/file2.txt",
    "r")
    # 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)
    # closes the opened file
    fileptr.close()
```

Результат работы программы:

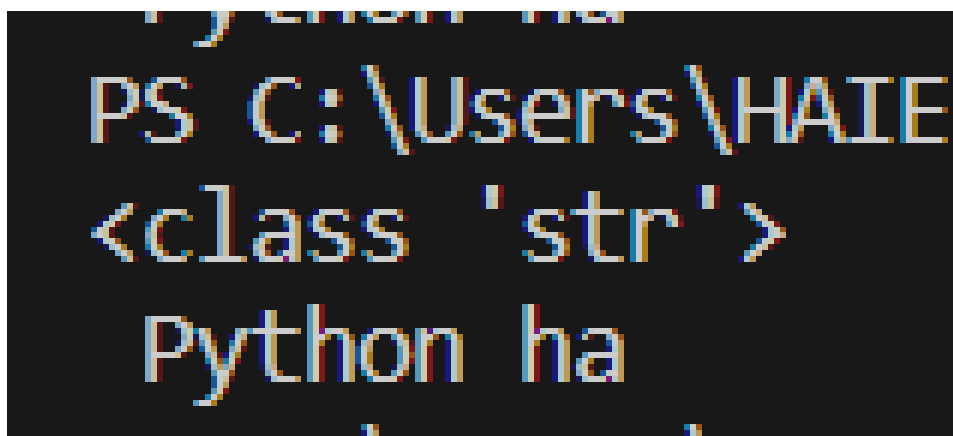


Рисунок 3 – Результат работы программы

Второй способ:

Код программы:

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

if __name__ == '__main__':
    # open the file2.txt in read mode. causes error if no such file exists.
    with open("C:/Users/HAIER/Desktop/Задания/Анализ данных/Data_lr_1/code/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)
```

Результат работы программы:

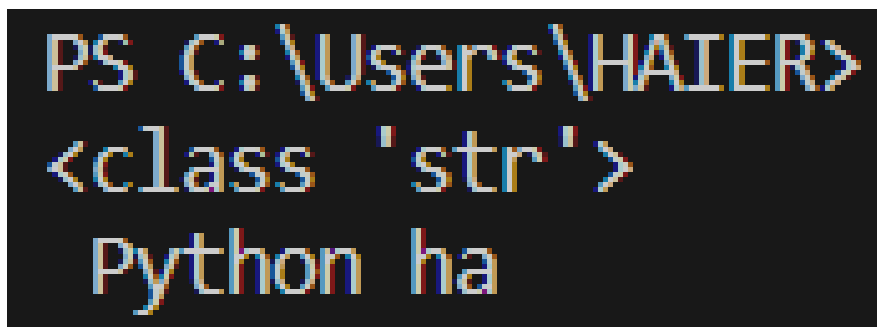


Рисунок 4 – Результат работы программы

Пример 3. Чтение строк с помощью метода readline()

Код программы:

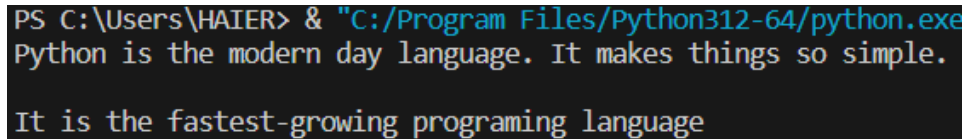
```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':
    # open the file2.txt in read mode. causes error if no such file exists.
    fileptr = open("file2.txt", "r")
    # stores all the data of the file into the variable content
    content1 = fileptr.readline()
    content2 = fileptr.readline()

    # prints the content of the file
    print(content1)
    print(content2)

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

Результат работы программы:



```
PS C:\Users\HAIER> & "C:/Program Files/Python312-64/python.exe"  
Python is the modern day language. It makes things so simple.  
It is the fastest-growing programing language
```

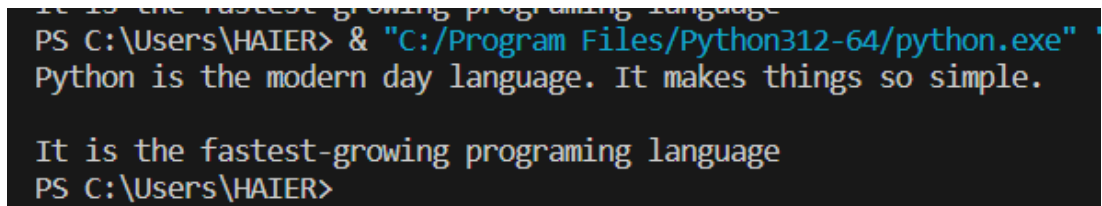
Рисунок 5 – Результат работы программы

Второй способ:

Код программы:

```
#!/usr/bin/env python3  
# -*- coding: utf-8 -*-  
  
if __name__ == '__main__':  
    # 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  
        content1 = fileptr.readline()  
        content2 = fileptr.readline()  
  
    # prints the content of the file  
    print(content1)  
    print(content2)
```

Результат работы программы:



```
PS C:\Users\HAIER> & "C:/Program Files/Python312-64/python.exe"  
Python is the modern day language. It makes things so simple.  
It is the fastest-growing programing language  
PS C:\Users\HAIER>
```

Рисунок 6 – Результат работы программы

Пример 4. Чтение строк с помощью функции readlines()

Код программы:

```
#!/usr/bin/env python3  
# -*- coding: utf-8 -*-  
  
if __name__ == '__main__':  
    # open the fil2.txt in read mode. causes error if no such file exists.  
    fileptr = open("file2.txt", "r")  
    # stores all the data of the file into the variable content  
    content = fileptr.readlines()  
    # prints the content of the file  
    print(content)  
    # closes the opened file  
    fileptr.close()
```

Результат работы программы:

```
PS C:\Users\HAIER> & "C:/Program Files/Python312-64/python.exe" "c:/Users/HAIER/Desktop/Задания/Анализ данных/Data_lr_1.py"
['Python is the modern day language. It makes things so simple.\n', 'It is the fastest-growing programming language']
PS C:\Users\HAIER>
```

Рисунок 7 – Результат работы программы

Второй способ:

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

if __name__ == '__main__':
    # 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.readlines()
        # prints the content of the file
        print(content)
```

Результат работы программы:

```
PS C:\Users\HAIER> & "C:/Program Files/Python312-64/python.exe" "c:/Users/HAIER/Desktop/Задания/Анализ данных/Data_lr_1.py"
['Python is the modern day language. It makes things so simple.\n', 'It is the fastest-growing programming language']
PS C:\Users\HAIER>
```

Рисунок 8 – Результат работы программы

Пример 5. Создание нового файла

Код программы:

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

if __name__ == '__main__':
    # open the newfile.txt in read mode. causes error if no such file exists.
    fileptr = open("newfile.txt", "x")
    print(fileptr)

    if fileptr:
        print("File created successfully")

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

Результат работы программы:

```
PS C:\Users\HAIER> & "C:/Program Files/Python312-64/python.exe" "c:/Users/HAIER/Desktop/Задания/Анализ данных/Data_lr_1.py"
<io.TextIOWrapper name='newfile.txt' mode='x' encoding='cp1251'>
File created successfully
PS C:\Users\HAIER>
```

Рисунок 9 – Результат работы программы

Второй способ:

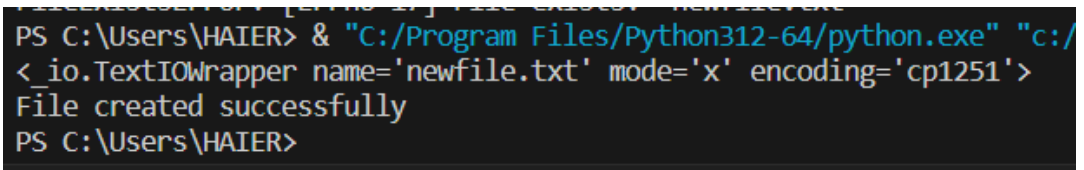
Код программы:

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

if __name__ == '__main__':
    # open the newfile.txt in read mode. causes error if no such file exists.
    with open("newfile.txt", "x") as fileptr:
        print(fileptr)

    if fileptr:
        print("File created successfully")
```

Результат работы программы:



```
PS C:\Users\HAIER> & "C:/Program Files/Python312-64/python.exe" "c:/
<_io.TextIOWrapper name='newfile.txt' mode='x' encoding='cp1251'>
File created successfully
PS C:\Users\HAIER>
```

Рисунок 10 – Результат работы программы

Пример 6. Изменение кодировки файла

Код программы:

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

if __name__ == "__main__":
    # open the text.txt in append mode. Create a new file if no such file exists.
    with open("text.txt", "w", encoding="utf-8") as fileptr:
        # appending the content to the file
        print(
            "UTF-8 is a variable-width character encoding used for electronic communication.",
            file=fileptr
        )
        print(
            "UTF-8 is capable of encoding all 1,112,064 valid character code points.",
            file=fileptr
        )
        print(
            "In Unicode using one to four one-byte (8-bit) code units.",
            file=fileptr
        )
```

Результат работы программы:

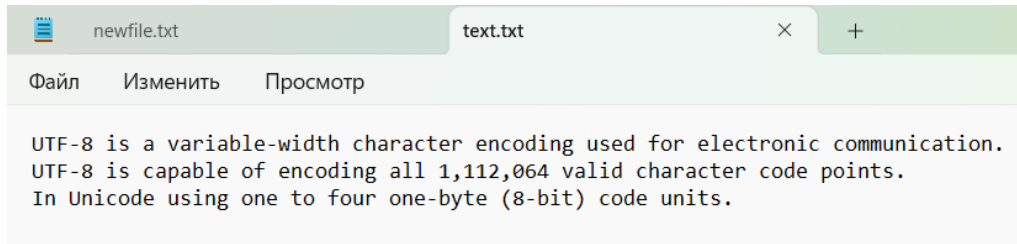


Рисунок 11 – Результат работы программы

Пример 7. Написать программу, которая считывает текст из файла и выводит на экран только предложения, содержащие запятые. Каждое предложение в файле записано на отдельной строке.

Код программы:

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

if __name__ == "__main__":
    with open("text.txt", "r", encoding="utf-8") as fileptr:
        sentences = fileptr.readlines()

    # Вывод предложений с запятыми.
    for sentence in sentences:
        if "," in sentence:
            print(sentence)
```

Результат работы программы:

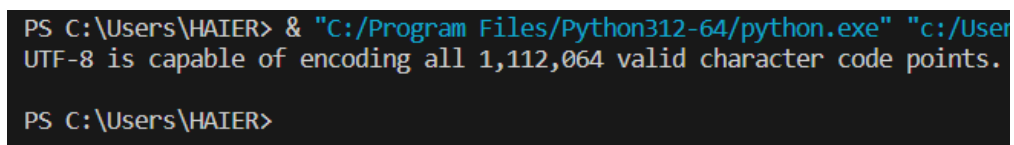


Рисунок 12 – Результат работы программы

Пример 8. Позиция указателя файла

Код программы:

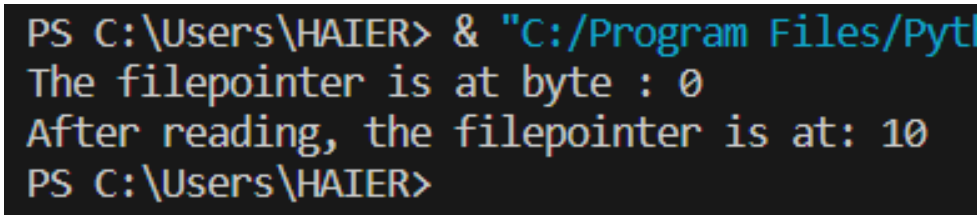
```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == "__main__":
    # open the file file2.txt in read mode
    with open("file2.txt", "r") as fileptr:
        #initially the filepointer is at 0
        print("The filepointer is at byte :", fileptr.tell())
        #changing the file pointer location to 10.
        fileptr.seek(10);
```



```
#tell() returns the location of the fileptr.  
print("After reading, the filepointer is at:",fileptr.tell())
```

Результат работы программы:



```
PS C:\Users\HAIER> & "C:/Program Files/Python/Python38-32/python.exe" fileptr.py  
The filepointer is at byte : 0  
After reading, the filepointer is at: 10  
PS C:\Users\HAIER>
```

Рисунок 13 – Результат работы программы

Пример 9. Переименование файла

Код программы:

```
#!/usr/bin/env python3  
# -*- coding: utf-8 -*-  
  
import os  
  
if __name__ == "__main__":  
    # rename file2.txt to file3.txt  
    os.rename("file2.txt", "file3.txt")
```

Результат работы программы:

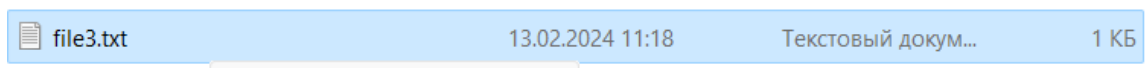


Рисунок 14 – Результат работы программы

Пример 10. Удаление файла

Код программы:

```
#!/usr/bin/env python3  
# -*- coding: utf-8 -*-  
  
import os  
  
if __name__ == "__main__":  
    # deleting the file named file3.txt  
    os.remove("file3.txt")
```

Результат работы программы:

Файл file3.txt был удалён

Пример 11. Создание нового каталога

Код программы:

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

import os

if __name__ == "__main__":
    #creating a new directory with the name new
    os.mkdir("new")
```

Результат работы программы:

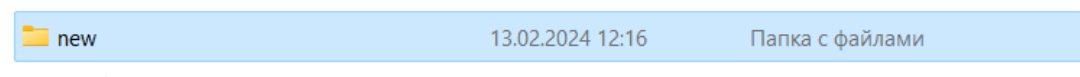


Рисунок 15 – Результат работы программы

Пример 12. Получение текущего каталога

Код программы:

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

import os

if __name__ == "__main__":
    path = os.getcwd()
    print(path)
```

Результат работы программы:

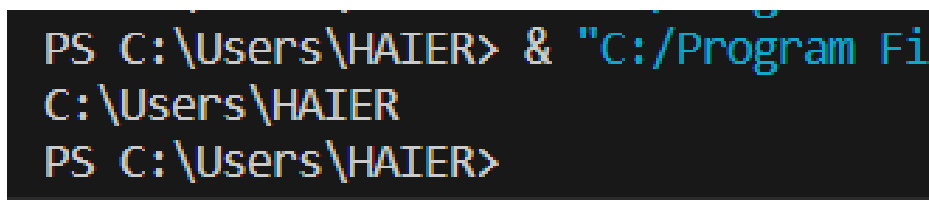


Рисунок 16 – Результат работы программы

Пример 13. Изменение текущего рабочего каталога

Код программы:

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

import os

if __name__ == "__main__":
    # Changing current directory with the new directory
    os.chdir("C://Windows")
    # It will display the current working directory
    print(os.getcwd())
```

Результат работы программы:

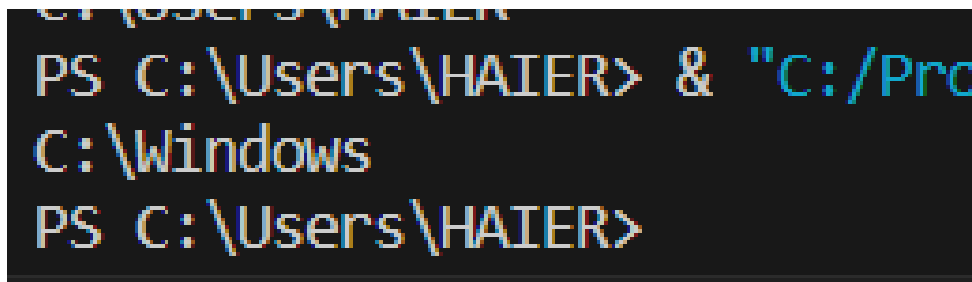


Рисунок 17 – Результат работы программы

Пример 14. Удаление каталога

Код программы:

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

import os

if __name__ == "__main__":
    # removing the new directory
    os.rmdir("new")
```

Результат работы программы:

Каталог «new» был удалён

Пример 15. Доступ к элементам командной строки в языке программирования Python.

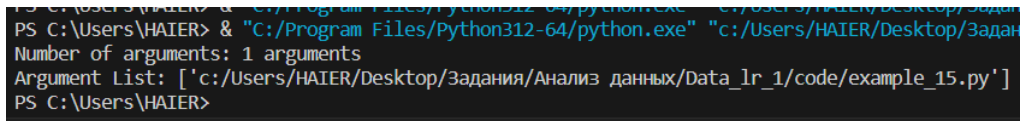
Код программы:

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

import sys

if __name__ == "__main__":
    print("Number of arguments:", len(sys.argv), "arguments")
    print("Argument List:", str(sys.argv))
```

Результат работы программы:



```
PS C:\Users\HAIER> & "C:/Program Files/Python312-64/python.exe" "c:/Users/HAIER/Desktop/Задан
Number of arguments: 1 arguments
Argument List: ['c:/Users/HAIER/Desktop/Задания/Анализ данных/Data_lr_1/code/example_15.py']
PS C:\Users\HAIER>
```

Рисунок 18 – Результат работы программы

Пример 16.

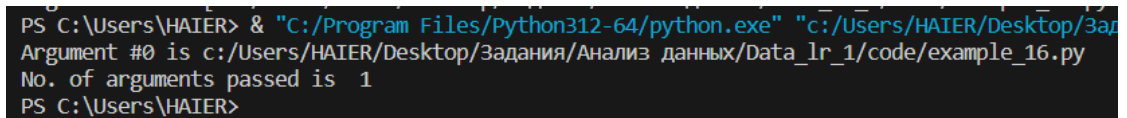
Код программы:

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

import sys

if __name__ == "__main__":
    for idx, arg in enumerate(sys.argv):
        print(f"Argument #{idx} is {arg}")
    print ("No. of arguments passed is ", len(sys.argv))
```

Результат работы программы:



```
PS C:\Users\HAIER> & "C:/Program Files/Python312-64/python.exe" "c:/Users/HAIER/Desktop/Зад
Argument #0 is c:/Users/HAIER/Desktop/Задания/Анализ данных/Data_lr_1/code/example_16.py
No. of arguments passed is 1
PS C:\Users\HAIER>
```

Рисунок 19 – Результат работы программы

Пример 17. Написать программу для генерации пароля заданной длины. Длина пароля должна передаваться как аргумент командной строки сценария.

Код программы:

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

import os
import secrets
import string
import sys

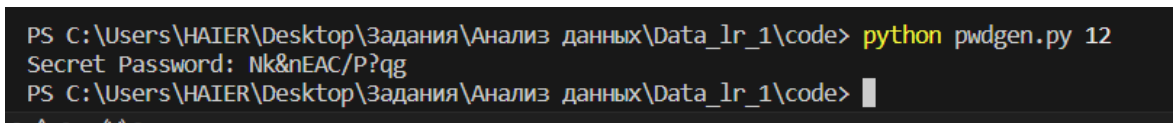
if __name__ == "__main__":
    if len(sys.argv) != 2:
        print("The password length is not given!", file=sys.stderr)
        sys.exit(1)

    chars = string.ascii_letters + string.punctuation + string.digits
    length_pwd = int(sys.argv[1])

    result = []
    for _ in range(length_pwd):
        idx = secrets.SystemRandom().randrange(len(chars))
        result.append(chars[idx])

    print(f"Secret Password: {"".join(result)}")
```

Результат работы программы:



```
PS C:\Users\HAIER\Desktop\Задания\Анализ данных\Data_lr_1\code> python pwdgen.py 12
Secret Password: Nk&nEAC/P?qg
PS C:\Users\HAIER\Desktop\Задания\Анализ данных\Data_lr_1\code> █
```

Рисунок 20 – Результат работы программы

Задание 2. Выполнены индивидуальные задания

Индивидуальное задание 1.

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

Код программы:

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

import re

if __name__ == "__main__":
    address = ("C:/Users/HAIER/Desktop/Задания/Анализ данных/" +
               "Data_lr_1/code/task1.txt")
    with open(address, "r", encoding="utf-8") as fileptr:
        sentences = fileptr.readlines()

    for sentence in sentences:
        result = (re.sub('\D', ' ', sentence).strip()).split()
        variable = True

        for i in result:
            if int(i) < 100 and int(i) >= 10:
                variable = False
                break

        if variable:
            print(sentence)
```

Содержимое файла:

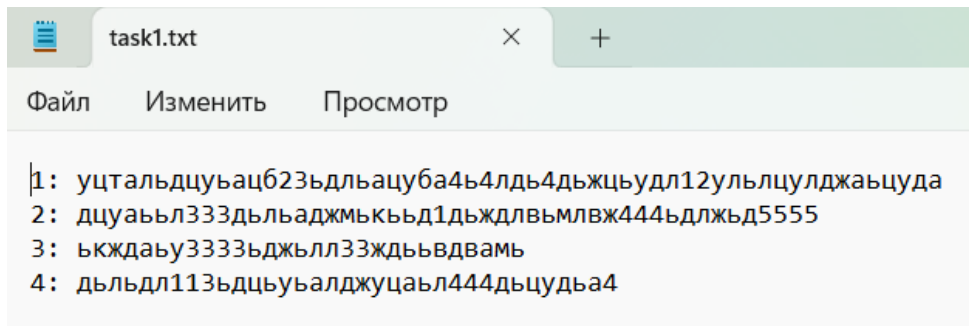


Рисунок 21 – Содержимое файла

Результат работы программы:

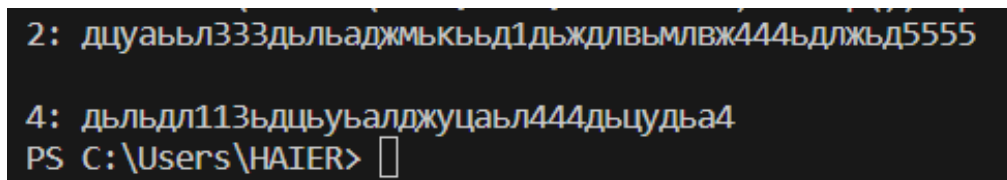


Рисунок 22 – Результат работы программы

Индивидуальное задание 2.

Перед публикацией текста или документа обычно принято удалять или изменять в них служебную информацию. В данном упражнении вам необходимо написать программу, которая будет заменять все служебные слова в тексте на символы звездочек (по количеству символов в словах). Вы должны осуществлять регистрозависимый поиск служебных слов в тексте, даже если эти слова входят в состав других слов. Список служебных слов должен храниться в отдельном файле. Сохраните отредактированную версию исходного файла в новом файле. Имена исходного файла, файла со служебными словами и нового файла должны быть введены пользователем. В качестве дополнительного задания расширьте свою программу таким образом, чтобы она выполняла замену служебных слов вне зависимости от того, какой регистр символов используется в тексте. Например, если в списке служебных слов будет присутствовать слово `exam`, то все следующие варианты слов должны быть заменены звездочками: `exam`, `Exam`, `ЕхaМ` и `EXAM`.

В каталоге, в котором находится файл с кодом, созданы файлы `text.txt` и `words.txt`.

Код программы:

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

import os
import re

if __name__ == "__main__":
    current_directory = os.path.dirname(os.path.abspath(__file__))

    words_file_name = input("Введите название файла со служебными словами: ")
    words_file_path = os.path.join(current_directory, words_file_name)

    with open(words_file_path, 'r') as fileptr:
        words = fileptr.readlines()

    words = [word.strip() for word in words]
    print(f"Список служебных слов из файла {words_file_name}:")
```

```
print(*words, sep="\n")

text_file_name = input("Введите название файла с текстом: ")
text_file_path = os.path.join(current_directory, text_file_name)

with open(text_file_path, "r") as fileptr:
    text = fileptr.readlines()

text = [string.strip() for string in text]
print(f"Список строк из файла {text_file_name}:")
print(*text, sep="\n")

"""for word in words:
    for i, string in enumerate(text):
        if string != string.replace(word, "*" * len(word)):
            text[i] = string.replace(word, "*" * len(word))"""
for word in words:
    for i, string in enumerate(text):
        result = re.sub(word, "*" * len(word), string)
        text[i] = result

new_text_file_name = input("Введите название нового файла с текстом: ")
new_text_file_path = os.path.join(current_directory, new_text_file_name)

with open(new_text_file_path, "w") as fileptr:
    for string in text:
        fileptr.write(string + "\n")

with open(new_text_file_path, "r") as fileptr:
    strings = fileptr.readlines()

print("Список строк из нового файла: ")
for string in strings:
    print(string.strip())
```


Результат работы программы:

```
Введите название файла со служебными словами: words.txt
Список служебных слов из файла words.txt:
Word1
Word3
Text
Введите название файла с текстом: text.txt
Список строк из файла text.txt:
Text
Text text text
text text
Word0Word1Word2
Введите название нового файла с текстом: new_text.txt
Список строк из нового файла:
****
**** text text
text text
Word0****Word2
```

Рисунок 23 – Результат работы программы

Программа была модифицирована для поиска служебных слов вне зависимости от используемого регистра.

Код программы:

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

import os

if __name__ == "__main__":
    current_directory = os.path.dirname(os.path.abspath(__file__))

    words_file_name = input("Введите название файла со служебными словами: ")
    words_file_path = os.path.join(current_directory, words_file_name)

    with open(words_file_path, 'r') as fileptr:
        words = fileptr.readlines()

    words = [word.strip() for word in words]
    print(f"Список служебных слов из файла {words_file_name}:")
    print(*words, sep="\n")

    text_file_name = input("Введите название файла с текстом: ")
    text_file_path = os.path.join(current_directory, text_file_name)

    with open(text_file_path, "r") as fileptr:
        text = fileptr.readlines()

    text = [string.strip() for string in text]
    print(f"Список строк из файла {text_file_name}:")
```

```

print(*text, sep="\n")

for word in words:
    for i, string in enumerate(text):
        s = string
        while (s.lower()).find(word.lower()) != -1:
            start_index = (s.lower()).find(word.lower())
            end_index = start_index + len(word)
            s = s[start_index] + "*" * len(word) + s[end_index:]
            text[i] = s

new_text_file_name = input("Введите название нового файла с текстом: ")
new_text_file_path = os.path.join(current_directory, new_text_file_name)

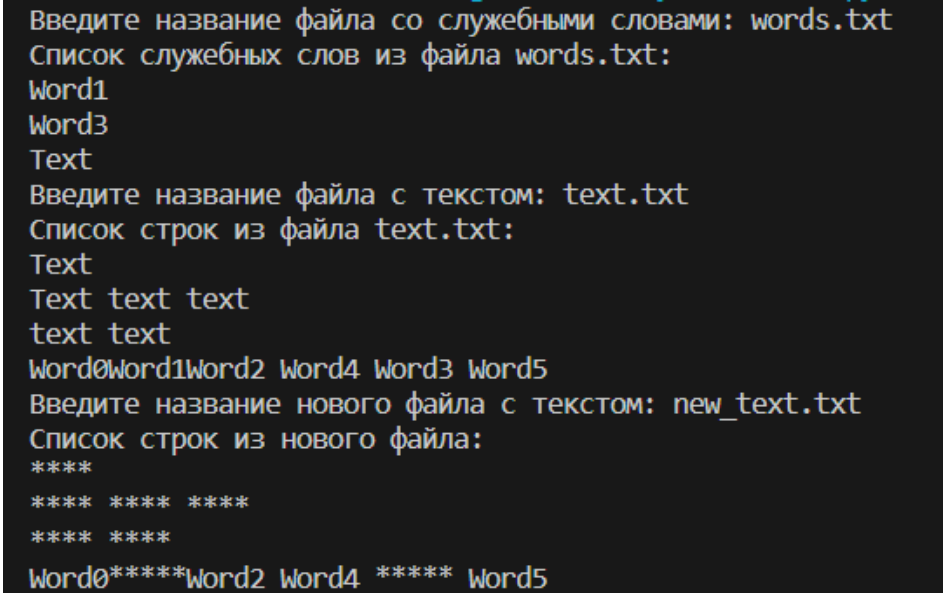
with open(new_text_file_path, "w") as fileptr:
    for string in text:
        fileptr.write(string + "\n")

with open(new_text_file_path, "r") as fileptr:
    strings = fileptr.readlines()

print("Список строк из нового файла: ")
for string in strings:
    print(string.strip())

```

Результат работы программы:



```

Введите название файла со служебными словами: words.txt
Список служебных слов из файла words.txt:
Word1
Word3
Text
Введите название файла с текстом: text.txt
Список строк из файла text.txt:
Text
Text text text
text text
Word0Word1Word2 Word4 Word3 Word5
Введите название нового файла с текстом: new_text.txt
Список строк из нового файла:
****
**** ****
****
Word0*****Word2 Word4 ***** Word5

```

Рисунок 24 – Результат работы программы

Ответы на контрольные вопросы:

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

Чтобы открыть файл в языке Python только для чтения, нужно использовать параметр «r» методе `open()`.

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

Чтобы открыть файл в языке Python только для записи, нужно использовать параметр «w» методе `open`.

3. Как прочитать данные из файла в языке Python?

Чтобы прочитать данные из файла в языке Python, нужно открыть файл в режиме, в котором возможно чтение файла, после чего использовать метод `readlines()` для считывания всех строк, содержащихся в файле.

4. Как записать данные в файл в языке Python?

Чтобы записать данные в файл в языке Python, нужно открыть файл в режиме, в котором возможна запись данных в файл, после чего использовать метод `write()` для записи строк в файл.

5. Как закрыть файл в языке Python?

Для закрытия файла используется метод `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 оборачивания выполнения блока инструкций менеджером контекста для гарантии того, что критические функции выполняются в любом случае.

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

Существуют следующие методы записи/чтения информации из файла:

`truncate()` (обрезает переданную строку до указанного количества символов).

`writelines()` (принимает список строк и записывает их в файл).

`readinto()` считывает байты в заранее выделенный, доступный для записи байтоподобный объект `b` и возвращает количество прочитанных байтов.

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

Существуют следующие функции модуля `os`:

`os.unlink(path)`

`os.symlink(src, dst)`

`os.link(src, dst)`

`os.chmod(path, mode)`

`os.utime(path, times)`

Вывод

В ходе выполнения лабораторной работы были получены навыки работы с файлами с помощью средств языка программирования Python.