

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

Кафедра инфокоммуникаций

Отчет по лабораторной работе №10

Работа с файлами в языке Python

По дисциплине «Технологии программирования и алгоритмизация»

Выполнил студент группы ИВТ-б-о-20-1

Галяс Д. И. « » _____ 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:

Ссылка на репозиторий: <https://github.com/DIMITRY-GALYAS1/Laba-2.15.git>

1. Создал новый репозиторий на github, после клонировал его и создал в папке репозитория новый проект PyCharm.
2. Проработал все примеры.

```
#!/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.
    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"
        )
```

Рисунок 1. Пример 1

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

if __name__ == '__main__':
    # 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.")
```

Рисунок 2. Пример 2

```
#!/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)
```

Рисунок 3. Пример 3

```
#!/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)
```

Рисунок 4. Пример 4

```
#!/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")
```

Рисунок 5. Пример 5

```
#!/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
        )
```

Рисунок 6. Пример 6

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

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

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

Рисунок 7. Пример 7

```
#!/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())
```

Рисунок 8. Пример 8

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

import os

if __name__ == "__main__":
    # rename file2.txt to file3.txt
    os.rename("file2.txt", "file3.txt")
```

Рисунок 9. Пример 9

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

import os

if __name__ == "__main__":
    # deleting the file named file3.txt
    os.remove("file3.txt")
```

Рисунок 10. Пример 10

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

import os

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

Рисунок 11. Пример 11

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

import os

|

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

Рисунок 12. Пример 12

```
#!/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())
```

Рисунок 13. Пример 13

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

import os

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

Рисунок 14. Пример 14

```
#!/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))
```

Рисунок 15. Пример 15

```
#!/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))
```

Рисунок 16. Пример 16

```
#!/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)}")
```

Рисунок 17. Пример 17

3. Выполнил первое индивидуальное задание.

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

"""
Написать программу, которая считывает текст из файла и
выводит на экран только предложения, содержащие введенное
с клавиатуры слово.
"""

if __name__ == '__main__':
    with open('text1.txt', 'r', encoding="utf-8") as z:
        sentences = z.readlines()

        word = input("Введите слово: ")

        for sentence in sentences:
            if word in sentence:
                print(sentence)
```

Рисунок 18. Индивидуальное задание 1

```
C:\Users\lizeq\anaconda3\envs\pythonProject10\python.exe C:/Users/lizeq/PycharmProjects/pythonProject10/individual_1.py
Введите слово: принц
Выйдет доверчивый маленький принц.
Process finished with exit code 0
```

Рисунок 19. Результат работы кода

4. Выполнил второе индивидуальное задание.


```

import sys

if __name__ == '__main__':
    # Задаем количество строк
    number_of_rows = 10

    # Если передано нужное количество аргументов
    if len(sys.argv) == 2:
        # Открываем файла
        with open(sys.argv[1], "r", encoding="utf-8") as f:
            z = f.readlines()

        # Создаем список для хранения строк
        last_lines = []
        # Просматриваем каждую строку
        for line in z:
            # Добавляем прочитанную строку в список
            last_lines.append(line)
            # Если строк больше, чем надо, то удаляем самую старую
            if len(last_lines) > number_of_rows:
                last_lines.pop(0)

        # Печатаем последние строки из файла
        for line in last_lines:
            print(line, end="")
    else:
        print("Ошибка, проверьте правильность ввода!")

```

Рисунок 20. Код второго индивидуального задания

```

PS C:\Users\lizeq\PycharmProjects\pythonProject10> py -3.8 individual_2.py text2.txt
И сколько нежности для сердца твоего!
Рождается дитя и сразу - же любимо,
Растет , и матери не чают в нем души.
За руки водят , с ним неразделимы,
Родителям любовь даруют малыши.
Проходит время , замуж вышла иль женился,
Своих детей родили , в школу повели...
С любовью в сердце мать за детей гордится,
И внуков любит , дети счастье родили...
Жить не любя...
PS C:\Users\lizeq\PycharmProjects\pythonProject10>

```

Рисунок 21. Работа кода

5. Выполнил третье индивидуальное задание.

```

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

"""
Написать программу, с помощью которой можно получить доступ
к названиям и путям всех подпапок и файлов, относящихся к
заданному каталогу.
"""

import os

if __name__ == "__main__":
    for root, directories, files in os.walk(r"C:\Users\lizeq\Laba-2.12\tasks"):
        print(root)
        for directory in directories:
            print(directory)
        for file in files:
            print(file)

```

Рисунок 22. Код третьего индивидуального задания

```

C:\Users\lizeq\anaconda3\envs\pythonProject10\python.exe C:/Users/lizeq/PycharmProjects/pythonProject10/individual_3.py
C:\Users\lizeq\Laba-2.12\tasks
individual_1.py
primer_1.py
primer_2.py

Process finished with exit code 0

```

Рисунок 23. Работа кода

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 выполнения лабораторной работы приобрел навыки по работе с текстовыми файлами при написании программ с помощью языка программирования Python версии 3.x, изучил основные методы модуля os для работы с файловой системой и получения аргументов командной строки.