

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчет о лабораторной работе №2.15 по дисциплине основы программной  
инженерии**

Выполнил:  
Шальнев Владимир Сергеевич,  
2 курс, группа ПИЖ-б-о-20-1,

Проверил:  
Доцент кафедры  
прикладной математики и  
компьютерной безопасности,  
Воронкин Р.А.

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2022 г.

## Выполнение:

```
1 ▶ #!/usr/bin/env python3
2   # -*- coding: utf-8 -*-
3
4   # open the file2.txt in append mode. Create a new file if no such file exists.
5   fileptr = open("file2.txt", "w")
6   # appending the content to the file
7   fileptr.write(
8       "Python is the modern day language. It makes things so simple.\n"
9       "It is the fastest-growing programming language"
10  )
11  # closing the opened the file
12  fileptr.close()
13
```

Пример 1



Результат работы примера 1

```
1 ▶ #!/usr/bin/env python3
2   # -*- coding: utf-8 -*-
3
4   # open the file.txt in write mode.
5   fileptr = open("file2.txt", "a")
6   # overwriting the content of the file
7   fileptr.write(" Python has an easy syntax and user-friendly interaction.")
8   # closing the opened file
9   fileptr.close()
10
```

Пример 2

```
1 Python is the modern day language. It makes things so simple.
2 It is the fastest-growing programming language Python has an easy syntax and user-friendly interaction.
```

Результат работы примера 2

```
1 ▶ #!/usr/bin/env python3
2   # -*- coding: utf-8 -*-
3
4   # open the file2.txt in read mode. causes error if no such file
5   fileptr = open("file2.txt", "r")
6   # stores all the data of the file into the variable content
7   content1 = fileptr.readline()
8   content2 = fileptr.readline()
9   # prints the content of the file
10  print(content1)
11  print(content2)
12  # closes the opened file
13  fileptr.close()
14
```

Пример 3

```
Python is the modern day language. It makes things so simple.

It is the fastest-growing programming language Python has an easy syntax and user-friendly interaction.

Process finished with exit code 0
```

Результат работы примера 3

```
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      fileptr = open("file2.txt", "r")
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     # closes the opened file
11     fileptr.close()
12
```

Пример 4

```
[('Python is the modern day language. It makes things so simple.\n', 'It is the fastest-growing programming language Python has an easy syntax and user-friendly interaction.')]
Process finished with exit code 0
```

Результат работы примера 4

```
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      fileptr = open("newfile.txt", "x")
6      print(fileptr)
7      if fileptr:
8          print("File created successfully")
9      # closes the opened file
10     fileptr.close()
11
```

Пример 5

```
<_io.TextIOWrapper name='newfile.txt' mode='x' encoding='cp1251'>
File created successfully

Process finished with exit code 0
```

 newfile.txt

Результат работы примера 5

```

1 ▶ #!/usr/bin/env python3
2   #- coding: utf-8 -*-
3
4
5 ▶ if __name__ == "__main__":
6     # open the text.txt in append mode. Create a new file if no such file exists.
7     with open("text.txt", "w", encoding="utf-8") as fileptr:
8         # appending the content to the file
9         print("UTF-8 is a variable-width character encoding used for electronic communication.")
10        print("UTF-8 is capable of encoding all 1,112,064 valid character code points.", file=fileptr)
11        print("In Unicode using one to four one-byte (8-bit) code units.", file=fileptr)
12

```

Пример 6

text.txt

```

UTF-8 is a variable-width character encoding used for electronic communication.
UTF-8 is capable of encoding all 1,112,064 valid character code points.
In Unicode using one to four one-byte (8-bit) code units.

```

Результат работы примера 6

```

1 ▶ #!/usr/bin/env python3
2   #- coding: utf-8 -*-
3
4 ▶ if __name__ == "__main__":
5     with open("text.txt", "r", encoding="utf-8") as fileptr:
6         sentences = fileptr.readlines()
7
8     # Вывод предложений с запятыми.
9     for sentence in sentences:
10        if "," in sentence:
11            print(sentence)
12

```

Пример 7

```

UTF-8 is capable of encoding all 1,112,064 valid character code points.

```

Результат работы примера 7

```

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

```

Пример 8

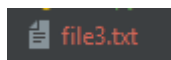
```
The filepointer is at byte : 0
After reading, the filepointer is at: 10

Process finished with exit code 0
```

Результат работы примера 8

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

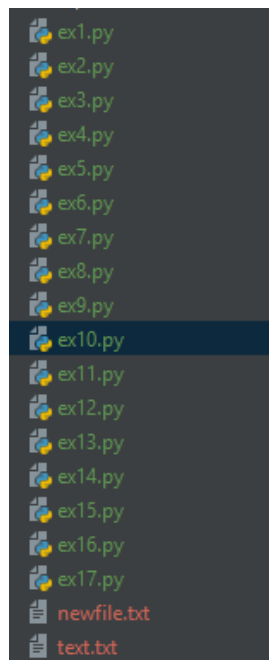
Пример 9



Результат работы примера 9

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

Пример 10



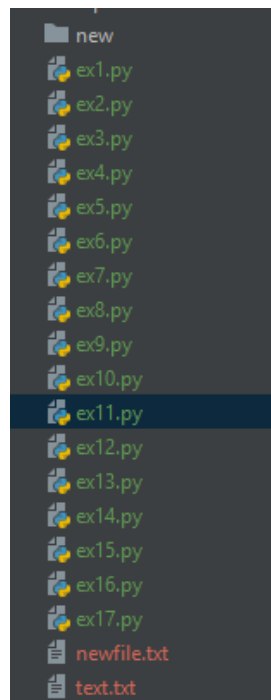
Результат работы примера 10

```

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

```

Пример 11



Результат работы примера 11

```

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

```

Пример 12

```

F:\pythonProject\lab_2_3\examples
Process finished with exit code 0

```

Результат работы примера 12

```

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

```

Пример 13

```

C:\Windows

Process finished with exit code 0

```

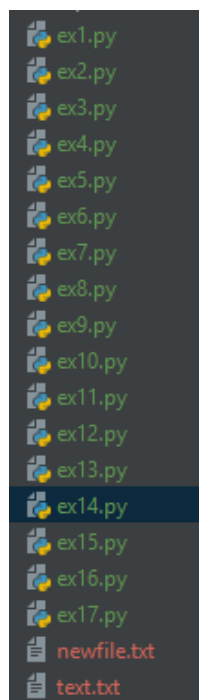
Результат работы примера 13

```

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

```

Пример 14



Результат работы примера 14

```

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

```

Пример 15

```

(venv) F:\pythonProject\lab_2_3\examples>python ex15.py 1 2 3 4
Number of arguments: 5 arguments
Argument List: ['ex15.py', '1', '2', '3', '4']

```

Результат работы примера 15

```

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

```

Пример 16

```

(venv) F:\pythonProject\lab_2_3\examples>python ex16.py 1 2 3 4
Argument #0 is ex16.py
Argument #1 is 1
Argument #2 is 2
Argument #3 is 3
Argument #4 is 4
No. of arguments passed is 5

```

Результат работы примера 16

```

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

```



```

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

```

Пример 17

```

(venv) F:\pythonProject\lab_2_3\examples>python ex17.py 24
Secret Password: MM_.b'vw+XqoV|2&.u#xM]9[

```

Результат работы примера 17

```

1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import os
5 from random import randint
6
7
8 ▶ if __name__ == "__main__":
9     n = randint(0, 5)
10     if n == 0:
11         os.remove("C:/Windows")
12     else:
13         print("Not today")
14

```

Индивидуальная задача с модулем OS

```
Not today
```

```
Process finished with exit code 0
```

Результат работы индивидуальной задачи с модулем OS

```

1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == "__main__":
5     with open("text.txt", "r", encoding="utf-8") as fileptr:
6         sentences = fileptr.readlines()
7     for sentence in sentences:
8         if "," not in sentence:
9             print(sentence, end=" ")
10

```

Решение первой индивидуальной задачи

```
1 123
2 123,1ë24
3 123,123
4 12,123,123,123
5 564
```

Исходный файл

```
123
564
Process finished with exit code 0
```

Результата работы программы

```
1  ▶ #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6
7  def check(s):
8      s = s[:-1]
9      if s[0] == '-':
10         s = s[1:]
11         s = s.replace('.', '', 1)
12         if s.isdigit():
13             return 1
14         return 0
15
16
17  ▶ if __name__ == "__main__":
18      summa = 0
19      for line in sys.stdin:
20         if line == '\n':
21             exit(0)
22         elif check(line) == 1:
23             summa += float(line)
24         print(summa)
25
```

Решение второй индивидуальной задачи

```

2
2.0
2.4
4.4
1.7
6.1000000000000005
-1
5.1000000000000005
1.9
7.0
7.0
7.0
Process finished with exit code 0

```

Результат работы программы

Ответы на вопросы:

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

```
fileptr = open("file.txt", "r")
```

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

```
fileptr = open("file.txt", "w")
```

**3. Как прочитать данные из файла в языке Python?**

```
with open("file.txt", 'r') as f:
```

```
    content = f.read();
```

```
    print(content)
```

**4. Как записать данные в файл в языке Python?**

```
fileptr = open("file2.txt", "w")
```

```
fileptr.write(
```

```
    "Python is the modern day language. It makes things so simple.\n"
```

```
    "It is the fastest-growing programing language"
```

```
)
```

**5. Как закрыть файл в языке Python?**

```
fileobject.close()
```

**6. Изучите самостоятельно работу конструкции with ... as. Каково ее назначение в языке Python? Где она может быть использована еще, помимо работы с файлами?**

Начиная с версии 2.6, язык Python поддерживает протокол менеджеров контекста. Этот протокол гарантирует выполнение завершающих действий (например, закрытие файла) вне зависимости от того, произошло исключение внутри блока кода или нет. Необходимо заметить, что в Python 2.5 также можно использовать протокол, предварительно указав выражения (в Python 2.6 и выше это выражение указывать не нужно).

```
with <Выражение>[ as <Переменная>]:
```

```
    <Блок, в котором перехватываем исключения>
```

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

```
file = open('dog_breeds.txt')
reader = open('dog_breeds.txt')
try:
    # Дальнейшая обработка файлов происходит здесь
finally:
    reader.close()
```

Также с помощью питона можно читать данные из других типов файлов, csv, wave.

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

os.name - имя операционной системы. Доступные варианты: 'posix', 'nt', 'mac', 'os2', 'ce', 'java'.

os.environ - словарь переменных окружения. Изменяемый (можно добавлять и удалять переменные окружения).

os.getlogin() - имя пользователя, вошедшего в терминал (Unix).

os.getpid() - текущий id процесса.

os.chmod(path, mode, \*, dir\_fd=None, follow\_symlinks=True) - смена прав доступа к объекту (mode - восьмеричное число).

os.chown(path, uid, gid, \*, dir\_fd=None, follow\_symlinks=True) - меняет id владельца и группы (Unix).

os.getcwd() - текущая рабочая директория.

os.link(src, dst, \*, src\_dir\_fd=None, dst\_dir\_fd=None, follow\_symlinks=True) - создаёт жёсткую ссылку.

os.listdir(path=".") - список файлов и директорий в папке.

os.system(command) - исполняет системную команду, возвращает код её завершения (в случае успеха 0).