

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное автономное образовательное

учреждение высшего образования

**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ»**

Институт цифрового

развития Кафедра информационных систем и

технологий

Отчет по лабораторной работе №18.

Дисциплина: «Основы программной инженерии»

Выполнил:

Студент группы

ПИЖ- б-о-22-1,

направление

подготовки:

09.03.04

«Программная инженерия»

ФИО: Гуртовой

Ярослав Дмитриевич

Проверил:

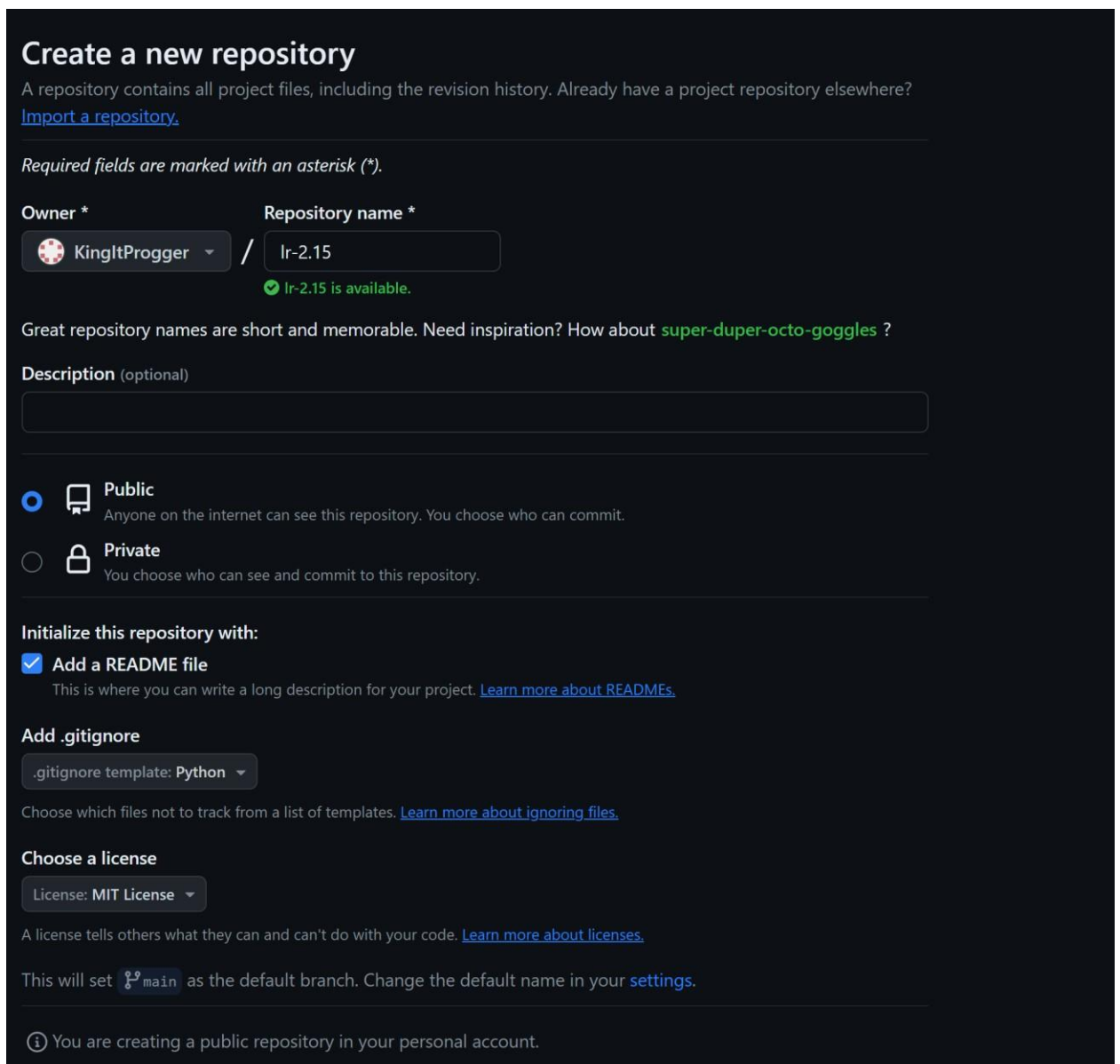
Богданов С.С

Ставрополь 2024

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.

Ссылка на репозиторий GitHub: <https://github.com/KingItProgger/lr-2.15>

1. Изучил теоритически й материал работы
2. Создал репозиторий





Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).


Owner * / Repository name *


 KingItProgger / lr-2.15

 lr-2.15 is available.

Great repository names are short and memorable. Need inspiration? How about [super-duper-octo-goggles](#) ?

Description (optional)


☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:


☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore


 .gitignore template: Python

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

 License: MIT License

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set  main as the default branch. Change the default name in your [settings](#).


 You are creating a public repository in your personal account.

Рисунок 1. Создание репозитория

3. Клонировал репозиторий

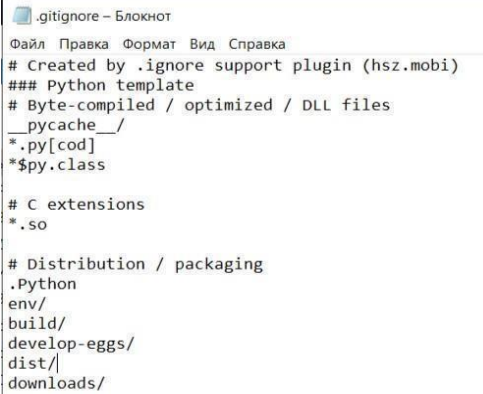
```
C:\Users\User\Desktop\учеба\4_сем\python> git clone https://github.com/KingItProgger/lr-2.15.git
Cloning into 'lr-2.15'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.

C:\Users\User\Desktop\учеба\4_сем\python>
```

Рисунок 1. Клонирование репозитория

4. Дополнил необходимыми правилами файл .gitignore

5.



```
.gitignore - Блокнот
Файл Правка Формат Вид Справка
# Created by .ignore support plugin (hsz.mobi)
### Python template
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*$py.class

# C extensions
*.so

# Distribution / packaging
.Python
env/
build/
develop-eggs/
dist/
downloads/
```

Рисунок 3 – .gitignore для IDE PyCharm

6. Организовал репозиторий в соответствии с моделью ветвления git-flow

```
C:\Users\User\Desktop\учеба\4_сем\python\lr-2.15> git checkout -b develop
Switched to a new branch 'develop'

C:\Users\User\Desktop\учеба\4_сем\python\lr-2.15>
```

Рисунок 4 – создание ветки develop

7. Проработал примеры лабораторной работы

```
# open the file2.txt in append mode. Create a new file if no such file exists.
fileptr = open("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 programming language"
)
# closing the opened the file
fileptr.close()
```

Рисунок 5 – пример 1

```
# open the file.txt in write mode.
fileptr = open("file2.txt", "a")

# overwriting the content of the file
fileptr.write(" Python has an easy syntax and user-friendly interaction.")

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

Рисунок 6 – пример 2

```
# 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() |
```

Рисунок 7 – пример 3

```
C:\Users\User\AppData\Local\Programs\Python\Python311\python.exe "C:/Users/User/Desktop/ОПИ/1r 2.15/ex3.py"
Python is the modern day language. It makes things so simple.

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

Process finished with exit code 0
```

Рисунок 8 – выполнение примера 3

```
# 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()
```

Рисунок 9 – пример 4

```
C:\Users\User\AppData\Local\Programs\Python\Python311\python.exe "C:/Users/User/Desktop/ОПМ/лр 2.15/ex4.py"
['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
```

Рисунок 10 – выполнение примера 4

```
# 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()
```

Рисунок 11 – пример 5

```
C:\Users\User\AppData\Local\Programs\Python\Python311\python.exe "C:/Users/User/Desktop/ОПМ/лр 2.15/ex5.py"
<_io.TextIOWrapper name='newfile.txt' mode='x' encoding='cp1251'>
File created successfully
Process finished with exit code 0
```

Рисунок 12 – выполнение примера 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 13 – пример 6

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ == "__main__":
    with open("text.txt", "r", encoding="utf-8") as fileptr:
        sentences = f.readlines()

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

Рисунок 14 – пример 7

```
C:\Users\User\AppData\Local\Programs\Python\Python311\python.exe "C:/Users/User/Desktop/ОПИ/lr 2.15/ex7.py"
UTF-8 is capable of encoding all 1,112,064 valid character code points.

Process finished with exit code 0
```

Рисунок 15 – выполнение примера 7

```
# 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 16 – пример 8

```
C:\Users\User\AppData\Local\Programs\Python\Python311\python.exe "C:/Users/User/Desktop/ОПИ/lr 2.15/ex8.py"
The filepointer is at byte : 0
After reading, the filepointer is at: 10

Process finished with exit code 0
```

Рисунок 17 – выполнение примера 8

```
import os  
  
.  
.  
.  
# deleting the file named file3.txt  
os.remove("file3.txt")
```

Рисунок 18 – пример 9

```
import os  
  
# creating a new directory with the name new  
os.mkdir("new")
```

Рисунок 19 – пример 11

```
import os  
  
path = os.getcwd()  
print(path)
```

Рисунок 20 – пример 12

```
C:\Users\User\AppData\Local\Programs\Python\Python38-32\Scripts>python lr 2.15  
C:\Users\User\Desktop\ОПИ\lr 2.15  
  
Process finished with exit code 0
```

Рисунок 21 – выполнение примера 12


```
import os
. . .
# Changing current directory with the new directory
os.chdir("C:\\Windows")
#It will display the current working directory
print(os.getcwd())
```

Рисунок 22 – пример 13

```
C:\Users\User\AppData\Local\Programs\Python
C:\Windows

Process finished with exit code 0
```

Рисунок 23 – выполнение примера 13

```
import os
|
# removing the new directory
os.rmdir("new")
```

Рисунок 24 – 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 25 – пример 15


```
C:\Users\User\AppData\Local\Programs\Python\Python311\python.exe
Number of arguments: 1 arguments
Argument List: ['C:/Users/User/Desktop/ОПИ/1r 2.15/ex15.py']

Process finished with exit code 0
```

Рисунок 26 – выполнение примера 15

```
C:\Users\User\AppData\Local\Programs\Python\Python311\python.exe
Argument #0 is C:/Users/User/Desktop/ОПИ/1r 2.15/ex16.py
No. of arguments passed is 1

Process finished with exit code 0
```

Рисунок 27 – пример 16

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

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 28 – пример 17

8. Выполнил индивидуальное задание

```
with open('asd.txt','r') as f:
    spis=f.readlines()

res=' '.join(spis)
res=res.split(' ')
for i in range(0,len(res)-1,2):
    c=res[i]
    res[i]=res[i+1]
    res[i+1]=c
print(" ".join(res))
```

Рисунок 29 – Задание 1.

```
C:\Users\User\AppData\Local\Programs\Python\Python311\python.exe "C:/Users/User/Desktop/000/Tr 2.15/task1.py"
s As writer professional extensive with I experience, been have with tasked a crafting text voluminous of consisting English 100 me words.
Allow present to with you following the excerpt: well-crafted enchanting "The painted sunset sky the hues with gold of crimson, and a casting light mesmerizing the over coastline. tranquil gentle
The whispered breeze the through palm swaying creating trees, soothing a that melody the embraced
Waves senses. upon crashed shore, the a creating of symphony power nature's beauty. and gracefully
Seagulls overhead, glided distant their merging cries with harmoniously rhythmic the of sound ocean. the sands,
The beneath warm feet, my tales whispered countless of days sun-soaked laughter. and this
In escape, momentary seemed time stand to as still, world the in reveled captivating the of serenity embrace." nature's

Process finished with exit code 0
```

Рисунок 30 – Выполнение Задания 1.

```

with open('asd.txt', 'r') as f:
    a=f.read().split()

    a.sort(key=len, reverse=True)
    max_ = len(a[0])
    print(max_)
    for i in range(len(a)):
        if len(a[i]) == max_:
            print(a[i])

```

Рисунок 31 – Задание 2

```

C:\Users\User\AppData\Local\Programs\Python\Python311\
12
professional
well-crafted
harmoniously

Process finished with exit code 0

```

Рисунок 32 – Выполнение Задания 2.

9. Подберал задачу для работы с изученными функциями модуля os .

Программа 1м аргументом принимает имя , 2м- 'datetime'/'date', Приветствует человека и выводит текущую дату\врем

```

import sys
import datetime

if __name__ == '__main__':

    #enter name and word 'date'/'time'/'datetime' .. you will get greeting and date/time/datetime

    if len(sys.argv)!=3:
        print('invalid count of arguments',file=sys.stderr)
        sys.exit(1)
    else:
        print(f"hello,{sys.argv[1]}")
        if sys.argv[2]=='date':
            print(datetime.date.today())

        elif sys.argv[2]=='datetime':
            print(datetime.datetime.now())
        else:
            print('invalid 2nd arg',file=sys.stderr)
            sys.exit(1)

```

Рисунок 33 – Задания 3.

```

C:\Users\User\Desktop\ОПИ\lr 2.15>python extra.py yaroslav date
hello,yaroslav
2024-02-25

```

Рисунок 34– Выполнение Задания 3.

10. Зафиксировал изменения в репозитории

```

C:\Users\User\Desktop\учеба\4_сэм\python\lr-2.15> git add .
warning: in the working copy of 'pycharm/.idea/inspectionProfiles/profiles_settings.xml', LF will be replaced by CRLF the next time Git touches it

C:\Users\User\Desktop\учеба\4_сэм\python\lr-2.15> git status
On branch develop
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   jnxtn.docx
    new file:   pycharm/.idea/.gitignore
    new file:   pycharm/.idea/.name
    new file:   pycharm/.idea/inspectionProfiles/profiles_settings.xml
    new file:   pycharm/.idea/lr 2.15.iml
    new file:   pycharm/.idea/misc.xml
    new file:   pycharm/.idea/modules.xml
    new file:   pycharm/.idea/pycharm.iml
    new file:   pycharm/.idea/vcs.xml
    new file:   pycharm/asd.txt
    new file:   pycharm/ex1
    new file:   pycharm/ex1.py
    new file:   pycharm/ex10.py
    new file:   pycharm/ex11.py
    new file:   pycharm/ex12.py
    new file:   pycharm/ex13.py
    new file:   pycharm/ex14.py
    new file:   pycharm/ex15.py
    new file:   pycharm/ex16.py
    new file:   pycharm/ex17.py
    new file:   pycharm/ex18.py
    new file:   pycharm/ex2.py
    new file:   pycharm/ex3.py
    new file:   pycharm/ex4.py
    new file:   pycharm/ex5.py
    new file:   pycharm/ex6.py
    new file:   pycharm/ex7.py
    new file:   pycharm/ex8.py
    new file:   pycharm/ex9.py
    new file:   pycharm/extra.py
    new file:   pycharm/file2.txt
    new file:   pycharm/newfile.txt
    new file:   pycharm/task1.py
    new file:   pycharm/task2.py
    new file:   pycharm/text.txt
    new file:   ~$jnxtn.docx

C:\Users\User\Desktop\учеба\4_сэм\python\lr-2.15> git commit -m "laba"
[develop a807cde] laba
36 files changed, 243 insertions(+)
create mode 100644 jnxtn.docx
create mode 100644 pycharm/.idea/.gitignore
create mode 100644 pycharm/.idea/.name
create mode 100644 pycharm/.idea/inspectionProfiles/profiles_settings.xml
create mode 100644 pycharm/.idea/lr 2.15.iml
create mode 100644 pycharm/.idea/misc.xml
create mode 100644 pycharm/.idea/modules.xml
create mode 100644 pycharm/.idea/pycharm.iml

```

Рисунок 36-фиксация изменений

11. Слил ветки

```

C:\Users\User\Desktop\учеба\4_сем\python\lr-2.15> git checkout main
Already on 'main'
Your branch is up to date with 'origin/main'.

C:\Users\User\Desktop\учеба\4_сем\python\lr-2.15> git merge develop
Updating e325911..a807cde
Fast-forward
 jnxtn.docx | Bin 0 -> 1117967 bytes
 pycharm/.idea/.gitignore | 3 +++
 pycharm/.idea/.name | 1 +
 .../.idea/inspectionProfiles/profiles_settings.xml | 6 ++++++
 pycharm/.idea/lr 2.15.iml | 8 ++++++++
 pycharm/.idea/misc.xml | 4 +++++
 pycharm/.idea/modules.xml | 8 ++++++++
 pycharm/.idea/pycharm.iml | 8 ++++++++
 pycharm/.idea/vcs.xml | 6 ++++++
 pycharm/asd.txt | 7 ++++++
 pycharm/ex1 | 0
 pycharm/ex1.py | 9 ++++++++
 pycharm/ex10.py | 0
 pycharm/ex11.py | 4 +++++
 pycharm/ex12.py | 4 +++++
 pycharm/ex13.py | 6 ++++++
 pycharm/ex14.py | 4 +++++
 pycharm/ex15.py | 6 ++++++
 pycharm/ex16.py | 7 ++++++
 pycharm/ex17.py | 21 ++++++++
 pycharm/ex18.py | 0
 pycharm/ex2.py | 8 ++++++
 pycharm/ex3.py | 10 ++++++++
 pycharm/ex4.py | 10 ++++++++
 pycharm/ex5.py | 8 ++++++
 pycharm/ex6.py | 18 ++++++++
 pycharm/ex7.py | 10 ++++++++
 pycharm/ex8.py | 8 ++++++
 pycharm/ex9.py | 4 +++++
 pycharm/extra.py | 21 ++++++++
 pycharm/file2.txt | 2 ++
 pycharm/newfile.txt | 0
 pycharm/task1.py | 15 ++++++++
 pycharm/task2.py | 14 ++++++++
 pycharm/text.txt | 3 +++
 ~$jnxtn.docx | Bin 0 -> 162 bytes
36 files changed, 243 insertions(+)

```

Рисунок 37-Слияние веток

Вывод: приобретены навыки по работе с текстовыми файлами при написании программ с помощью языка программирования Python версии 3.x, изучены основные методы модуля os для работы с файловой системой, получения аргументов командной строки.

Контрольные вопросы:

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

Используйте функцию ``open()`,` чтобы открыть файл. Укажите путь к файлу в качестве первого аргумента и используйте режим чтения ``mode 'r'``.

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

Используйте функцию `open()`, чтобы открыть файл. Укажите путь к файлу в качестве первого аргумента и используйте режим чтения `'w'`

3. Как прочитать данные из файла в языке Python?

Чтение из файла Python можно реализовать с помощью функции `read()`, если он открыт в режиме чтения `r`. Синтаксис функции выглядит следующим образом: `file.read(size)`. В строке выше `file` обозначает объект, с которым работаем, а `size` — количество символов для чтения. Если ничего не указать в качестве аргумента функции `read()`, то получится прочитать сразу все содержимое документа.

4. Как записать данные в файл в языке Python?

Использование функции `write()`:

```
with open("examp.le", "w") as f:
```

```
    f.write(some_string_data)
```

Запись большого объёма данных в виде списка строк:

```
with open("examp.le", "w") as f:
```

```
    f.writelines(list_of_strings)
```

Использование функции `print()`:

```
with open("examp.le", "w") as f:
```

```
    print(some_data, file=f)
```

5. Как закрыть файл в языке Python?

```
file.close()
```

6. Изучите самостоятельно работу конструкции `with ... as`. Каково ее назначение в языке

Python? Где она может быть использована еще, помимо работы с файлами?

Преимущество заключается в том, что файл правильно закрывается после завершения его набора, даже если в какой-то момент возникает

исключение. Использование `with` также намного короче, чем написание эквивалентных блоков `try-finally`:

7. Изучите самостоятельно документацию Python по работе с файлами.

Какие помимо

рассмотренных существуют методы записи/чтения информации из файла?

Методы файла в Python

`file.close()` закрывает открытый файл

`file.fileno()` возвращает целочисленный дескриптор файла

`file.flush()` очищает внутренний буфер

`file.isatty()` возвращает `True`, если файл привязан к терминалу

`file.next()` возвращает следующую строку файла

`file.read(n)` чтение первых `n` символов файла

`file.readline()` читает одну строчку строки или файла

`file.readlines()` читает и возвращает список всех строк в файле

`file.seek(offset[,whence])` устанавливает текущую позицию в файле

`file.seekable()` проверяет, поддерживает ли файл случайный доступ.

Возвращает `True`, если да

`file.tell()` возвращает текущую позицию в файле

`file.truncate(n)` уменьшает размер файл. Если `n` указала, то файл обрезается до `n` байт, если нет — до текущей позиции

`file.write(str)` добавляет строку `str` в файл

`file.writelines(sequence)` добавляет последовательность строк в файл

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

системой?

`import os`

`os.chdir(r"D:\folder")`

`D:\folder`-изменение директории

`import os`

```
print(os.path.exists("D:/test.txt"))
```

True-проверка существования пути

```
import os
```

```
print(os.path.isfile("D:/test.txt"))
```

True-проверка объекта на явление файлом

```
import os
```

```
os.rmdir(r"D:\folder")-удаление директ
```

```
import os
```

```
os.startfile(r"D:\test.txt")-запуск
```

```
import os
```

```
print(os.path.basename("D:/test.txt"))
```

test.txt-получение имени файла

```
import os
```

```
print(os.path.getsize("D:\\test.txt"))
```

136226-получение размера

```
import os
```

```
print(os.listdir(r"D:\folder"))
```

['first', 'test.txt']-содержимое директорий

```
import os
```

```
print(os.path.split(r"D:\folder\test.txt"))
```

```
('D:\\folder', 'test.txt')
```

```
print(os.path.join(r"D:\folder", "test.txt"))
```

D:\folder\test.txt-обработка путей