

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования «СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций
Институт цифрового развития

ОТЧЁТ
по лабораторной работе №2.15

Дисциплина: «Программирование на Python»

Тема: «Работа с файлами в языке Python»

Вариант 2

Выполнил: студент 2 курса,
группы ИВТ-б-о-21-1
Богдан Александр Анатольевич

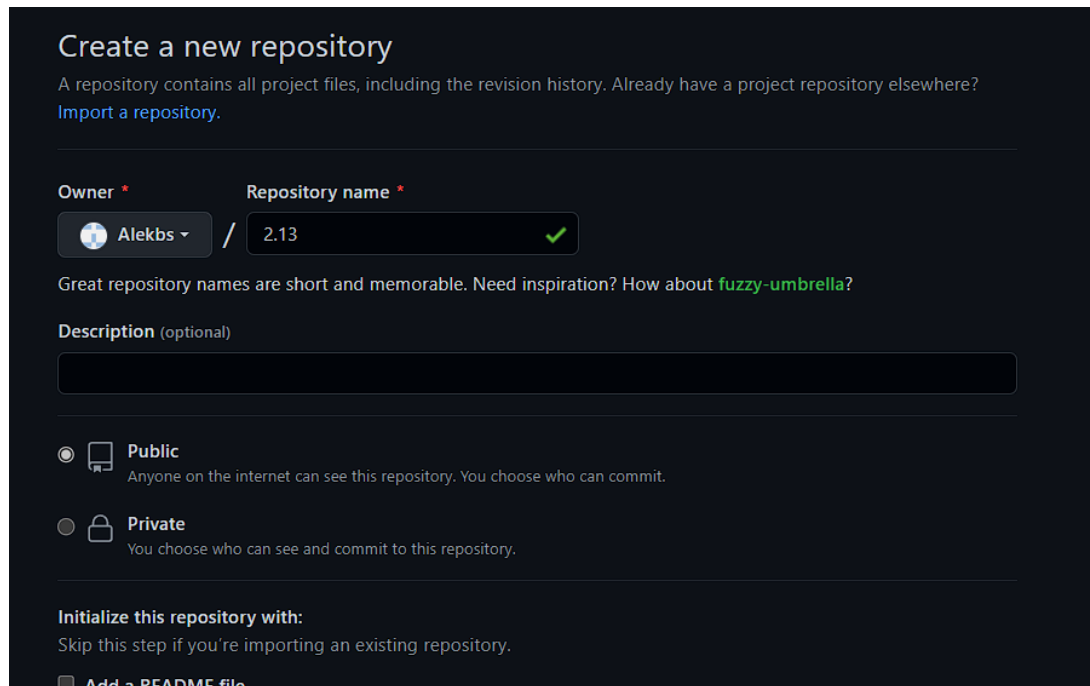
Ставрополь 2022

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.

Практическая часть:

1. Создал репозиторий, клонировал его, изменил файл .gitignore.



Организовал репозиторий в соответствии с моделью ветвления git-flow



Create a new repository


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


Owner * Repository name *

 Alekb's / 2.13 

Great repository names are short and memorable. Need inspiration? How about [fuzzy-umbrella?](#)

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:

Skip this step if you're importing an existing repository.

☐ Add a README file

Рисунок 1. Создание репозитория

```
# Created by https://www.toptal.com/developers/gitignore/api/pycharm,python
# Edit at https://www.toptal.com/developers/gitignore?templates=pycharm,python

### PyCharm ###
# Covers JetBrains IDEs: IntelliJ, RubyMine, PhpStorm, AppCode, PyCharm, CLion, Android Studio, WebStorm and Rider
# Reference: https://intellij-support.jetbrains.com/hc/en-us/articles/206544839

# User-specific stuff
.idea/**/workspace.xml
.idea/**/tasks.xml
.idea/**/usage.statistics.xml
.idea/**/dictionaries
.idea/**/shelf
.idea/
.idea

# AWS User-specific
.idea/**/aws.xml

# Generated files
.idea/**/contentModel.xml

# Sensitive or high-churn files
.idea/**/dataSources/
.idea/**/dataSources.ids
.idea/**/dataSources.local.xml
.idea/**/sqlDataSources.xml
.idea/**/dynamic.xml
.idea/**/uiDesigner.xml
.idea/**/dbnavigator.xml

# Gradle
.idea/**/gradle.xml
.idea/**/libraries

# Gradle and Maven with auto-import
```

Рисунок 3. Изменение файла .gitignore

```

Which branch should be used for bringing forth production releases?
  - develop
  - main
Branch name for production releases: [main] main

Which branch should be used for integration of the "next release"?
  - develop
Branch name for "next release" development: [develop] rel
Fatal: Local branch 'rel' does not exist.

D:\WAR\gi\2.13>git flow init

Which branch should be used for integration of the "next release"?
  - develop
Branch name for "next release" development: [develop] develop

How to name your supporting branch prefixes?
Feature branches? [feature/] fea
Bugfix branches? [bugfix/] bug
Release branches? [release/] rel
Hotfix branches? [hotfix/] hot
Support branches? [support/] sup
Version tag prefix? [] pre
Hooks and filters directory? [D:/WAR/gi/2.13/.git/hooks] hook

```

Рисунок 4. Организация репозитория в соответствии с git-flow

2. Проработал примеры:

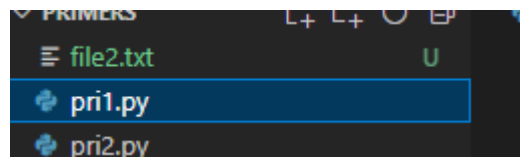


Рисунок 5. Результат работы примера 1

```

file2.txt
1 Python is the modern day language. It makes things so simple.
2 It is the fastest-growing programming language Python has an easy syn

```

Рисунок 6. Результат работы примера 2

```

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 i
nteraction.
PS C:\Users\super\Desktop\Cfi\Lab-2.15\Primers> 

```

Рисунок 7. Результат работы примера 3

```

../../debugpy\launcher' '51795' '--' 'c:\Users\super\Desktop\Cfi\Lab-2.15\Primers\pri4.py'
['Python is the modern day language. It makes things so simple.\n', 'It is the fastest-grow
ing programming language Python has an easy syntax and user-friendly interaction.']
PS C:\Users\super\Desktop\Cfi\Lab-2.15\Primers> 

```

Рисунок 8. Результат работы примера 4

```

../..\\debugpy\\launcher 51845 -- c:\\Users\\super\\Desktop\\Cfi\\Lab-2.15\\Prim
<_io.TextIOWrapper name='newfile.txt' mode='x' encoding='cp1251'>
File created successfully
PS C:\\Users\\super\\Desktop\\Cfi\\Lab-2.15\\Primers>

```

Рисунок 9. Результат работы примера 5

```

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

```

Рисунок 10. Результат работы примера 6

```

UTF-8 is capable of encoding all 1,112,064 valid character code points.
PS C:\\Users\\super\\Desktop\\Cfi\\Lab-2.15\\Primers>

```

Рисунок 11. Результат работы примера 7

```

The filepointer is at byte : 0
After reading, the filepointer is at: 10
PS C:\\Users\\super\\Desktop\\Cfi\\Lab-2.15\\Primers>

```

Рисунок 12. Результат работы примера 8

```

file3.txt      U
newfile.txt    U
pri1.py

```

Рисунок 13. Результат работы примера 9

```

newfile.txt    U
pri1.py

```

Рисунок 14. Результат работы примера 10

```

> new
newfile.txt    U
pri1.py

```

Рисунок 15. Результат работы примера 11

```

C:\\Users\\super\\Desktop\\Cfi\\Lab-2.15\\Primers
PS C:\\Users\\super\\Desktop\\Cfi\\Lab-2.15\\Primers>

```

Рисунок 16. Результат работы примера 12

```
C:\Windows
PS C:\Users\super\Desktop\Cfi\Lab-2.15\Primers> 
```

Рисунок 17. Результат работы примера 13

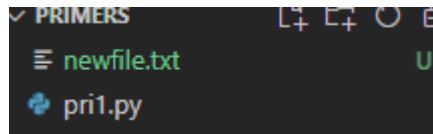


Рисунок 18. Результат работы примера 14

```
Number of arguments: 1 arguments
Argument List: ['c:\\Users\\super\\Desktop\\Cfi\\Lab-2.15\\Primers\\pri15.py']
PS C:\Users\super\Desktop\Cfi\Lab-2.15\Primers> 
```

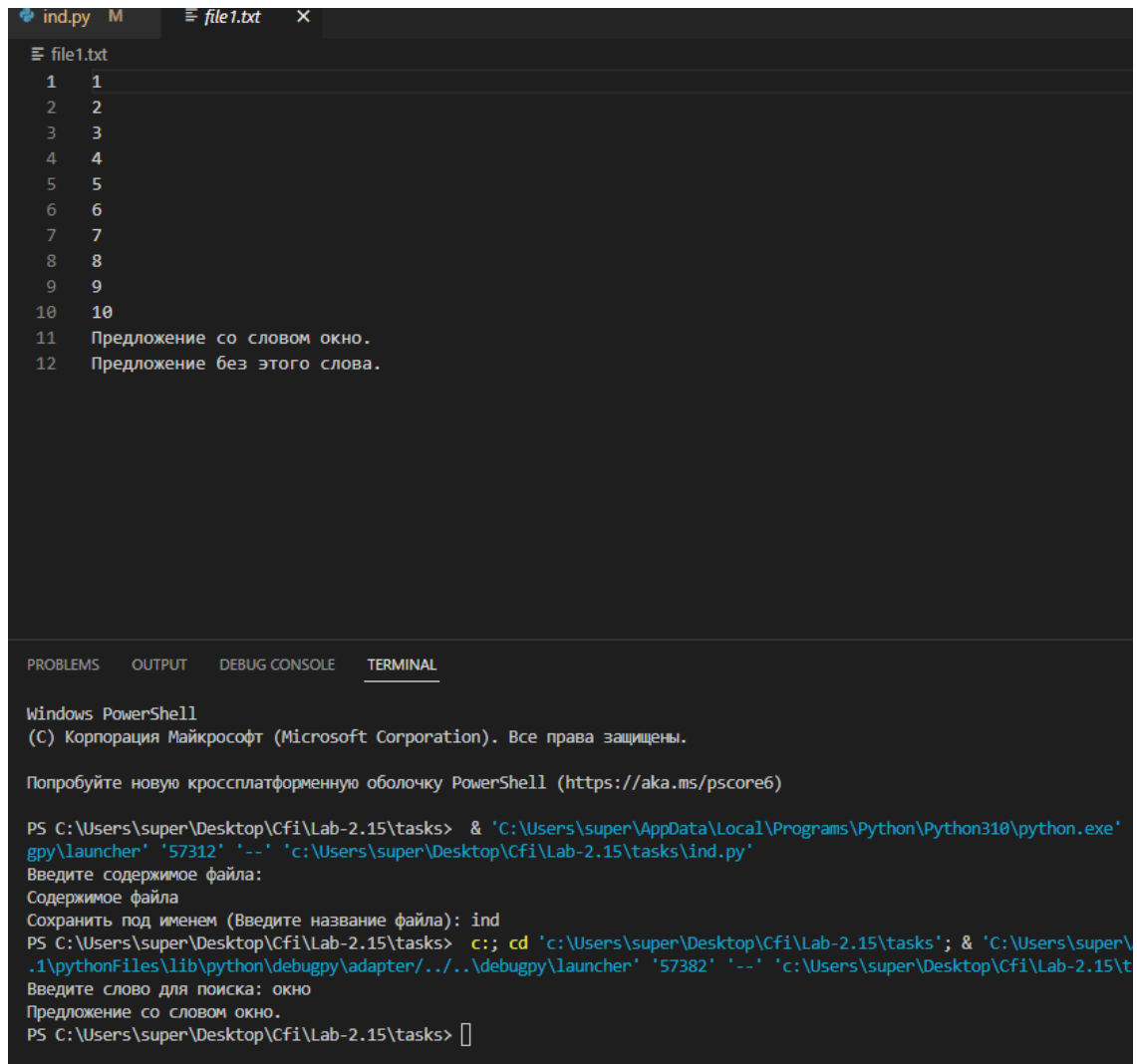
Рисунок 19. Результат работы примера 15

4. Выполнил индивидуальное задание:

Задание 1

Составить программу с использованием списков и словарей для решения задачи. Номер варианта определяется по согласованию с преподавателем. Исходный файл, из которого выполняется чтение, необходимо также добавить в репозиторий, каждое предложение в файле должно находиться на отдельной строке.

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



The image shows a code editor window with a file named `file1.txt` open. The file contains 12 lines of text: lines 1-10 are the numbers 1 through 10, line 11 is "Предложение со словом окно.", and line 12 is "Предложение без этого слова." Below the editor is a terminal window titled "TERMINAL" showing the execution of a Python script. The terminal output includes the Windows PowerShell prompt, the execution of `python.exe` with arguments for the script and file, and the script's execution flow: "Введите содержимое файла:", "Содержимое файла", "Сохранить под именем (Введите название файла): ind", and the execution of `cd` to change the directory. The terminal also shows the script's search logic: "Введите слово для поиска: окно", "Предложение со словом окно.", and the final PowerShell prompt.

```
ind.py M file1.txt X
file1.txt
1 1
2 2
3 3
4 4
5 5
6 6
7 7
8 8
9 9
10 10
11 Предложение со словом окно.
12 Предложение без этого слова.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/powershell)

PS C:\Users\super\Desktop\Cfi\Lab-2.15\tasks> & 'C:\Users\super\AppData\Local\Programs\Python\Python310\python.exe'
gpy\launcher' '57312' '--' 'c:\Users\super\Desktop\Cfi\Lab-2.15\tasks\ind.py'
Введите содержимое файла:
Содержимое файла
Сохранить под именем (Введите название файла): ind
PS C:\Users\super\Desktop\Cfi\Lab-2.15\tasks> cd 'c:\Users\super\Desktop\Cfi\Lab-2.15\tasks'; & 'C:\Users\super\
.1\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '57382' '--' 'c:\Users\super\Desktop\Cfi\Lab-2.15\t
Введите слово для поиска: окно
Предложение со словом окно.
PS C:\Users\super\Desktop\Cfi\Lab-2.15\tasks> 
```

Рисунок 20. Результат выполнения индивидуального задания 1

Задание 2

Составить программу с использованием текстовых файлов. Номер варианта необходимо получить у преподавателя.

Продолжая тему предыдущего упражнения, в тех же операционных системах на базе Unix обычно есть и утилита с названием `tail`, которая отображает последние десять строк содержимого файла, имя которого передается в качестве аргумента командной строки. Реализуйте программу, которая будет делать то же самое. В случае отсутствия файла, указанного пользователем, или аргумента командной строки вам нужно вывести соответствующее сообщение.

```
передать имя файла в качестве аргумента командной строки. [ ind2.py , file , 1.txt ]  
PS C:\Users\super\Desktop\Cf1\Lab-2.15\tasks> python ind2.py file1.txt  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

Рисунок 21. Результат выполнения индивидуального задания 2

Задание 3

Самостоятельно подберите или придумайте задачу для работы с изученными функциями модуля os. Приведите решение этой задачи.

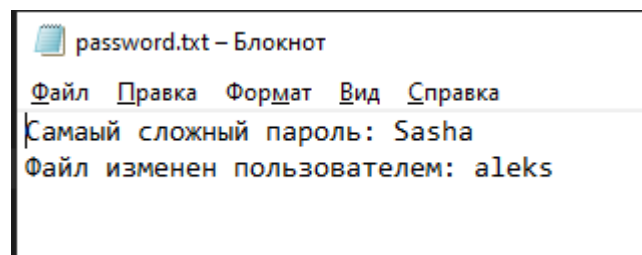


Рисунок 22. Результат выполнения индивидуального задания 3

Контрольные вопросы:

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

r – открывает файл в режиме только для чтения. Указатель файла существует в начале. Файл по умолчанию открывается в этом режиме, если не передан режим доступа.

Например: fileptr = open("file.txt","r").

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

w – только для записи. Он перезаписывает файл, если он существовал ранее, или создает новый, если файл с таким именем не существует. Указатель имеется в начале файла.

3. Как прочитать данные из файла в языке Python?

Чтобы прочитать файл с помощью сценария Python, Python предоставляет метод `read()`. Метод `read()` считывает строку из файла. Он может читать данные как в текстовом, так и в двоичном формате.

Синтаксис метода `read()`: `fileobj.read(<count>)`.

Мы можем прочитать файл построчно, используя цикл `for`.

Python упрощает чтение файла построчно с помощью метода `readline()`. Метод `readline()` читает строки файла с самого начала, т. е. если мы используем его два раза, мы можем получить первые две строки файла.

4. Как записать данные в файл в языке Python?

Чтобы записать текст в файл, нам нужно открыть файл с помощью метода `open` с одним из следующих режимов доступа.

'w': он перезапишет файл, если какой-либо файл существует. Указатель файла находится в начале файла.

'a': добавит существующий файл. Указатель файла находится в конце файла. Он создает новый файл, если файл не существует.

5. Как закрыть файл в языке Python?

После того, как все операции будут выполнены с файлом, мы должны закрыть его с помощью нашего скрипта Python, используя метод `close()`. Любая незаписанная информация уничтожается после вызова метода `close()` для файлового объекта.

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.

Синтаксис конструкции `with ... as`:

```
"with" expression ["as" target] ("," expression ["as" target])* ":"  
suite
```


Теперь по порядку о том, что происходит при выполнении данного блока:

- 1) Выполняется выражение в конструкции with ... as.
- 2) Загружается специальный метод `__exit__` для дальнейшего использования.
- 3) Выполняется метод `__enter__`. Если конструкция with включает в себя слово as, то возвращаемое методом `__enter__` значение записывается в переменную.
- 4) Выполняется suite.
- 5) Вызывается метод `__exit__`, причём неважно, выполнилось ли suite или произошло исключение. В этот метод передаются параметры исключения, если оно произошло, или во всех аргументах значение None, если исключения не было.

Если в конструкции with - as было несколько выражений, то это эквивалентно нескольким вложенным конструкциям.

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

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

Функции модуля os:

- Получение информации об ОС Чтобы узнать имя текущей ОС, достаточно воспользоваться методом `name`. ...
- Изменение рабочей директории ...
- Проверка существования пути ...
- Создание директорий ...
- Удаление файлов и директорий ...
- Запуск на исполнение ...
- Получение имени файла и директории ...

- Вычисление размера

Вывод: в результате выполнения работы были приобретены навыки 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.