

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 8

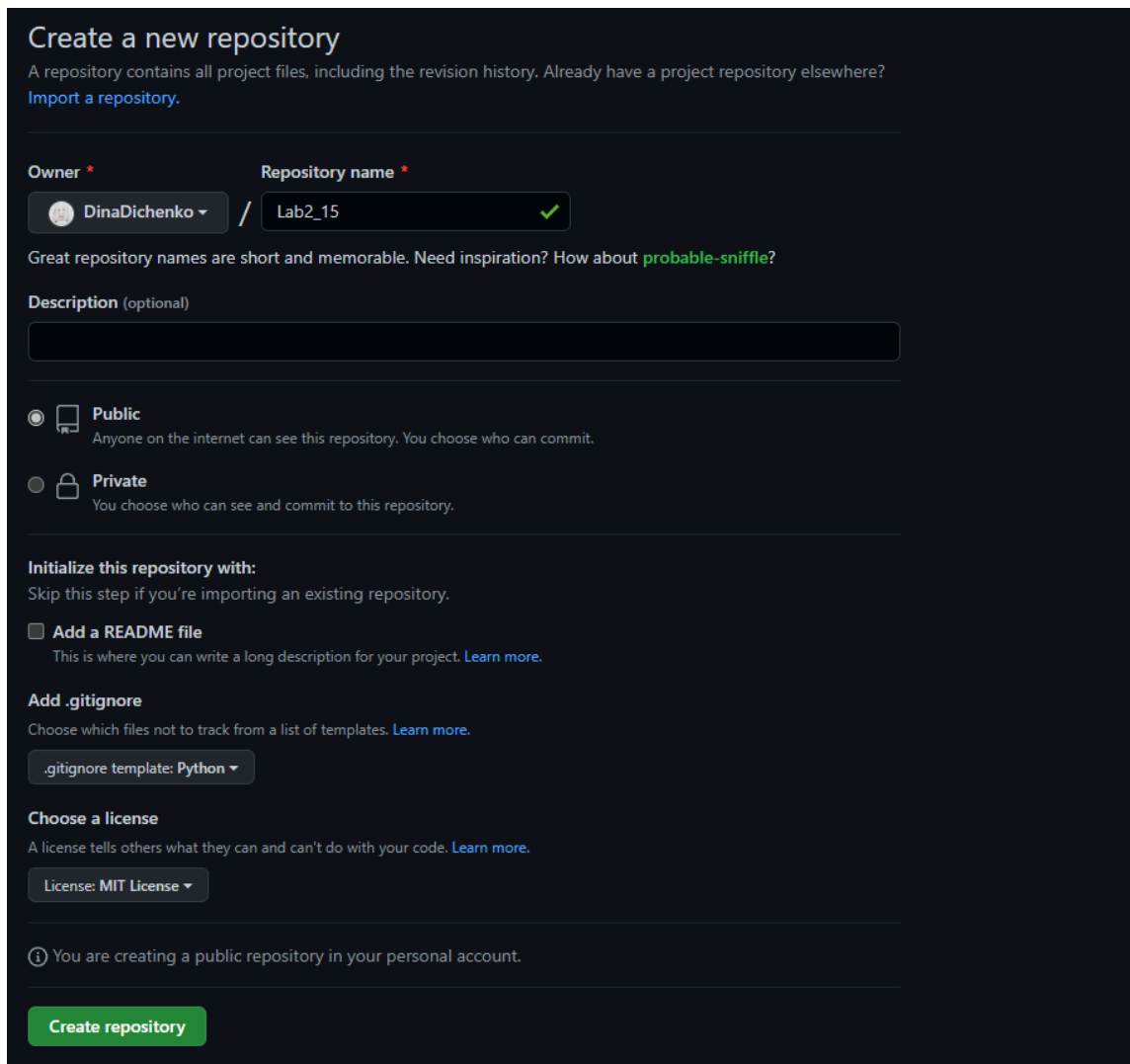
Выполнил: студент 2 курса,  
группы ИВТ-б-о-21-1  
Диченко Дина Алексеевна

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. Создала репозиторий.



Create a new repository

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

Owner \* DinaDichenko / Repository name \* Lab2\_15 ✓

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

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  
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore  
Choose which files not to track from a list of templates. [Learn more.](#)  
.gitignore template: Python

Choose a license  
A license tells others what they can and can't do with your code. [Learn more.](#)  
License: MIT License

(i) You are creating a public repository in your personal account.

[Create repository](#)

Рисунок 1. Создание репозитория

#### 2. Клонировала репозиторий.

```
C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python>git clone https://github.com/DinaDichenko/Lab2_15.git
Cloning into 'Lab2_15'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python>_
```

Рисунок 2. Клонирование репозитория

### 3. Изменила файл .gitignore.

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

### Python ###
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*$py.class
.idea/
.idea

# C extensions
*.so
.idea/
# Distribution / packaging
.Python
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/
var/
wheels/
```

Рисунок 3. Изменение файла .gitignore

### 4. Организовала репозиторий в соответствии с моделью ветвления git-flow.

```
C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python\Lab2_15>git branch develop
C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python\Lab2_15>git checkout develop
Switched to branch 'develop'

C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python\Lab2_15>git flow init

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] develop

How to name your supporting branch prefixes?
Feature branches? [feature/] fea
Bugfix branches? [bugfix/] bug
Release branches? [release/] branch
Hotfix branches? [hotfix/] hot
Support branches? [support/] sup
Version tag prefix? [] pre
Hooks and filters directory? [C:/Users/super/Desktop/Dina/ВУЗ/Программирование на python/Lab2_15/.git/hooks] hoo
```

Рисунок 4. Организация репозитория в соответствии с git-flow

### 5. Проработала примеры.

file2.txt – Блокнот

Файл Правка Формат Вид Справка

Python is the modern day language. It makes things so simple.  
It is the fastest-growing programming language

Рисунок 5. Результат работы примера 1

file2.txt – Блокнот

Файл Правка Формат Вид Справка

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

Рисунок 6. Результат работы примера 2

```
<class 'str'>  
Python is  
PS C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python\Lab2_15\prog>
```

Рисунок 7. Результат работы примера 3

```
Python is the modern day language. It makes things so simple.  
  
It is the fastest-growing programming language  
PS C:\Users\super>
```

Рисунок 8. Результат работы примера 4

```
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.  
PS C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python\Lab2_15\prog>
```

Рисунок 9. Результат работы примера 5

```
['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.']  
PS C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python\Lab2_15\prog>
```

Рисунок 9. Результат работы примера 6

```
primer7.py  
<_io.TextIOWrapper name='newfile.txt' mode='x' encoding='cp1251'>  
File created successfully  
PS C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python\Lab2_15\prog>
```

Рисунок 10. Результат работы примера 7

```
UTF-8 is capable of encoding all 1,112,064 valid character code points.
PS C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python\Lab2_15\prog>
```

Рисунок 11. Результат работы примера 8

```
The filepointer is at byte : 0
After reading, the filepointer is at: 10
PS C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python\Lab2_15\prog>
```

Рисунок 12. Результат работы примера 9



Рисунок 13. Результат работы примера 11

Имя	Дата изменения	Тип
individual1.py	29.11.2022 22:42	И
individual2.py	29.11.2022 22:42	И
newfile.txt	29.11.2022 22:43	Т
primer1.py	29.11.2022 22:29	И
primer2.py	29.11.2022 22:32	И

Рисунок 14. Результат работы примера 12

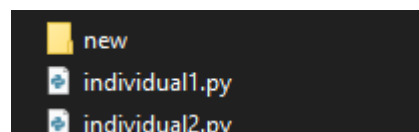


Рисунок 15. Результат работы примера 13

```
C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python\Lab2_15\prog
PS C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python\Lab2_15\prog>
```

Рисунок 16. Результат работы примера 14

```
C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python\Lab2_15\prog>python primer16.py Knowledge Hut 21
Number of arguments: 4 arguments
Argument List: ['primer16.py', 'Knowledge', 'Hut', '21']
```

Рисунок 17. Результат работы примера 16

6. Выполнила индивидуальные задания.

Задание 1. Написать программу, которая считывает текст из файла и выводит на экран только цитаты, то есть предложения, заключенные в кавычки.

Рисунок 18. Индивидуальное задание 1

Задание 2. Как вы знаете, в языке Python для создания комментариев в коде используется символ #. Комментарий начинается с этого символа и продолжается до конца строки – без возможности остановить его раньше. В данном упражнении вам предстоит написать программу, которая будет удалять все комментарии из исходного файла с кодом на языке Python. Пройдите по всем строкам в файле на предмет поиска символа #. Обнаружив его, программа должна удалить все содержимое, начиная с этого символа и до конца строки. Для простоты не будем рассматривать ситуации, когда знак решетки встречается в середине строки. Сохраните новое содержимое в созданном файле. Имена файла источника и файла назначения должны быть запрошены у пользователя. Удостоверьтесь в том, что программа корректно обрабатывает возможные ошибки при работе с обоими файлами.

#### Рисунок 19. Индивидуальное задание 2

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

8.

#### **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.