

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ  
УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**«Модули и пакеты в Python»**

**Отчет по лабораторной работе № 2.13  
по дисциплине «Программирование на Python»**

Выполнил студент группы ИВТ-б-о-21-1

Сумин Никита Сергеевич.

« » \_\_\_\_\_ 2022г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверил Воронкин Р.А. \_\_\_\_\_  
(подпись)

Ставрополь 2022

**Цель работы:** приобретение навыков по работе с модулями и пакетами при написании программ с помощью языка программирования Python версии 3.x.


**Порядок выполнения работы:**

1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python.

Create a new repository

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

Owner \*

 BrandooDi ▾

Repository name \*

/ mywork2.13 ✓

Great repository names are short and memorable. Need inspiration? How about [bug-free-journey?](#)

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 ▾

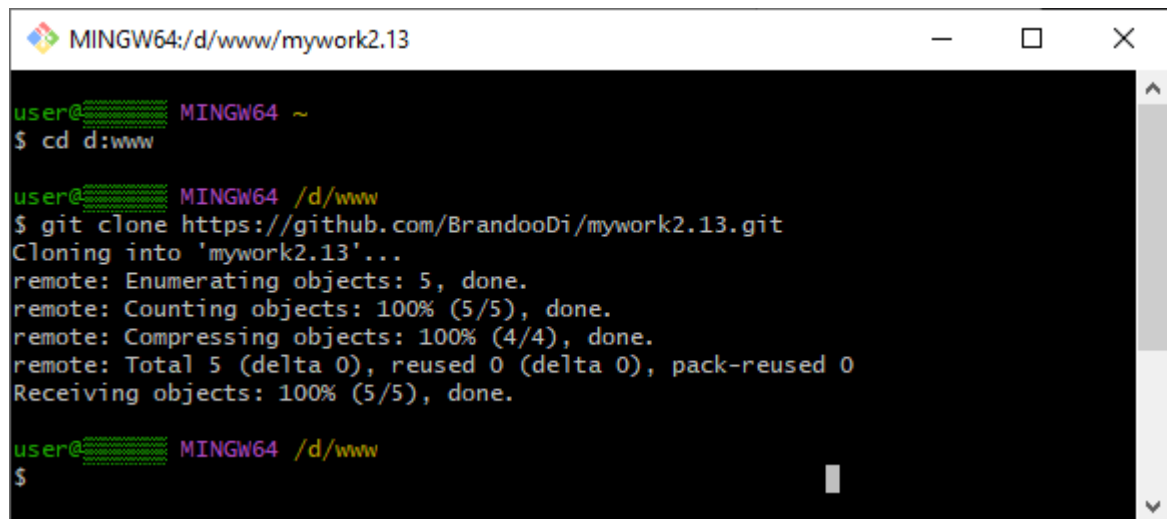
This will set  main as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

Create repository

Рисунок 1 - Создание репозитория

2. Выполните клонирование созданного репозитория.



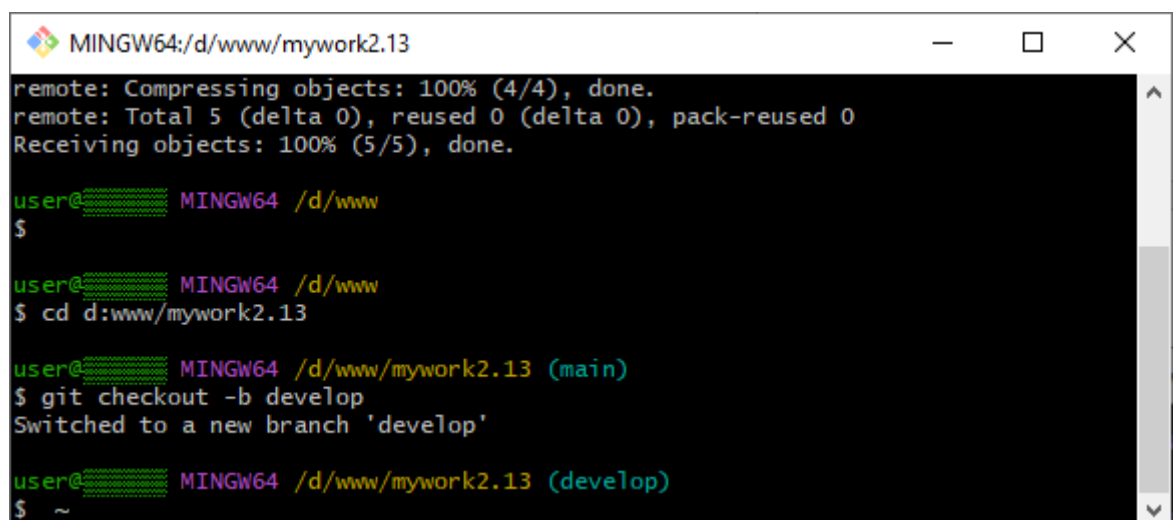
```
MINGW64:/d/www/mywork2.13
user@MINGW64 ~
$ cd d:/www

user@MINGW64 /d/www
$ git clone https://github.com/BrandooDi/mywork2.13.git
Cloning into 'mywork2.13'...
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.

user@MINGW64 /d/www
$
```

Рисунок 2 - Клонирование репозитория

3. Организуйте свой репозиторий в соответствии с моделью ветвления git-flow.



```
MINGW64:/d/www/mywork2.13
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.

user@MINGW64 /d/www
$

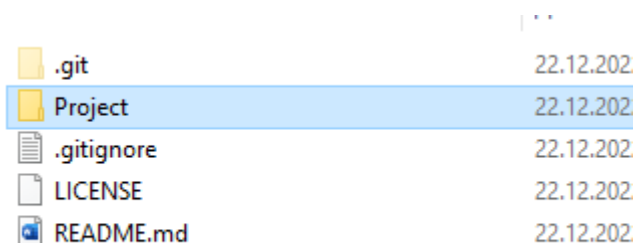
user@MINGW64 /d/www
$ cd d:/www/mywork2.13

user@MINGW64 /d/www/mywork2.13 (main)
$ git checkout -b develop
Switched to a new branch 'develop'

user@MINGW64 /d/www/mywork2.13 (develop)
$ ~
```

Рисунок 3 - Ветвление по модели git-flow 4.

Создайте проект PyCharm в папке репозитория.



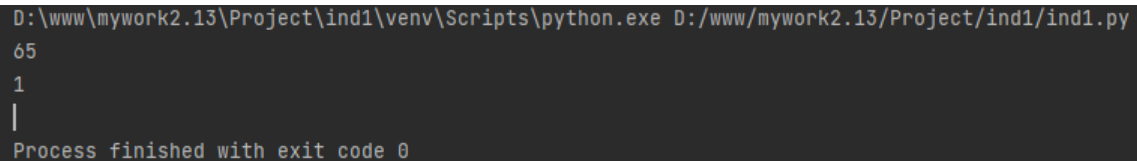
.git	22.12.202
Project	22.12.202
.gitignore	22.12.202
LICENSE	22.12.202
README.md	22.12.202

Рисунок 4 - Проект PyCharm

5. Выполнить индивидуальные задания.

Задание 1. Выполнить индивидуальное задание лабораторной работы 2.11, оформив все функции программы в виде отдельного модуля. Разработанный модуль должен быть подключен в основную программу с помощью одного из вариантов команды `import`.

Используя замыкания функций, объявите внутреннюю функцию, которая бы все повторяющиеся символы заменяла одним другим указанным символом. Какие повторяющиеся символы искать и на что заменять, определяются параметрами внешней функции. Внутренней функции передается только строка для преобразования. Преобразованная (сформированная) строка должна возвращаться внутренней функцией.



```
D:\www\mywork2.13\Project\ind1\venv\Scripts\python.exe D:/www/mywork2.13/Project/ind1/ind1.py
65
1
|
Process finished with exit code 0
```

Рисунок 5 - Результат выполнения индивидуального задания 1

Задание 2. Выполнить индивидуальное задание лабораторной работы 2.8, оформив все классы программы в виде отдельного пакета. Разработанный пакет должен быть подключен в основную программу с помощью одного из вариантов команды `import`. Настроить соответствующим образом переменную `__all__` в файле `__init__.py` пакета.

Использовать словарь, содержащий следующие ключи: название пункта назначения рейса; номер рейса; тип самолета. Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть размещены в алфавитном порядке по названиям пунктов назначения; вывод на экран пунктов назначения и номеров рейсов, обслуживаемых самолетом, тип которого введен с клавиатуры; если таких рейсов нет, выдать на дисплей соответствующее сообщение.

```
D:\www\mywork2.13\Project\ind2\venv\Scripts\python.exe D:/www/mywork2.13/Project/ind2/ind2.py
>>> add
Расчётный счёт платильщика: A
Расчётный счет получателя: K
Перечисляемая сумма в руб: 52235
>>> add
Расчётный счёт платильщика: B
Расчётный счет получателя: A
Перечисляемая сумма в руб: 52235
>>> list
+-----+-----+-----+-----+
| No | Расчётный счёт платильщика | Расчётный счет получателя | Перечисляемая сумма в руб |
+-----+-----+-----+-----+
| 1 | A | K | 52235 |
| 2 | B | A | 52235 |
+-----+-----+-----+-----+
>>> select 52235
+-----+-----+-----+-----+
| No | Расчётный счёт платильщика | Расчётный счет получателя | Перечисляемая сумма в руб |
+-----+-----+-----+-----+
| 1 | A | K | 52235 |
| 2 | B | A | 52235 |
+-----+-----+-----+-----+
>>> |
```

Рисунок 6 - Результат выполнения индивидуального задания 2

### Контрольные вопросы:

#### 1. Что является модулем языка Python?

Под модулем в Python понимается файл с расширением .py.

#### 2. Какие существуют способы подключения модулей в языке Python?

```
import имя_модуля
import имя_модуля as новое_имя from имя_модуля import
имя_объекта from имя_модуля import имя_объекта as
псевдоним_объекта from имя_модуля import *
```

#### 3. Что является пакетом языка Python?

Пакет в Python – это каталог, включающий в себя другие каталоги и модули, но при этом дополнительно содержащий файл `__init__.py`.

#### 4. Каково назначение файла `__init__.py` ?

Если файл с именем `__init__.py` присутствует в каталоге пакета, то он вызывается при импорте пакета или модуля в пакете. Это может быть использовано для выполнения кода инициализации пакета, например инициализации данных уровня пакета.

5. Каково назначение переменной `__all__` файла `__init__.py` ?

`__all__` хранит список модулей, который импортируется при загрузке через конструкцию `from имя_пакета import *`.

**Вывод:** были приобретены навыки по работе с модулями и пакетами при написании программ с помощью языка программирования Python версии 3.x.