

**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 № 8

Дисциплина: «Программирование на Python»

Тема: «Модули и пакеты»

Выполнил: студент 2 курса

группы ИВТ-б-о-21-1

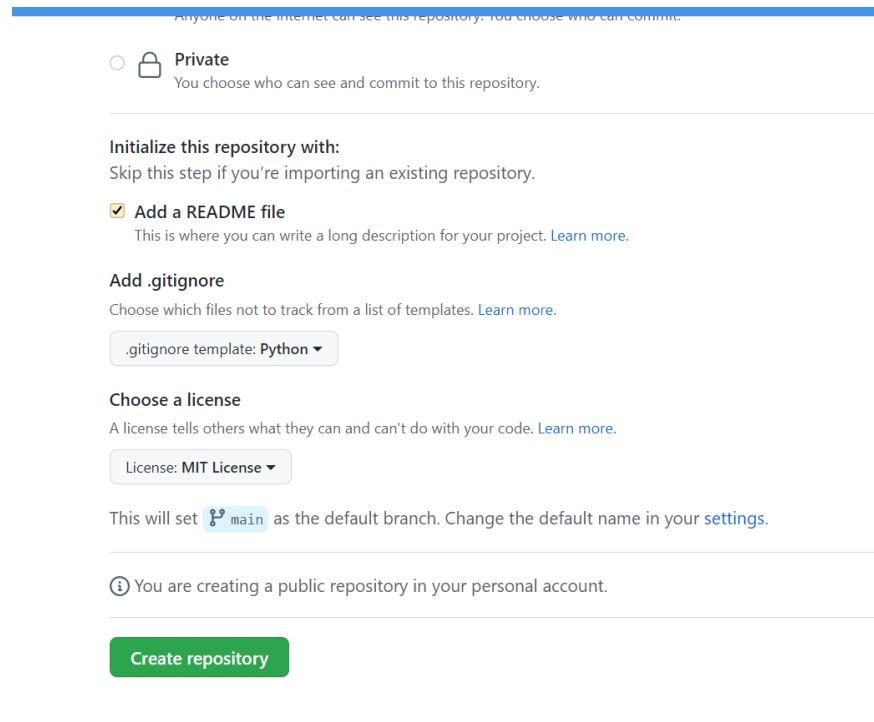
Уланбекова Айканыш Уланбековна

Ставрополь 2022

Цель работы: приобретение навыков по работе с модулями и пакетами языка программирования Python версии 3.x.

Порядок выполнения работы:

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



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.

```
C:\Users\User>cd C:\Users\User\Desktop\2 кypc Python\lab 16
C:\Users\User\Desktop\2 кypc Python\lab 16>git clone https://github.com/aikanyshkauanbekova/2.13.git
Cloning into '2.13'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (8/8), 4.49 KiB | 574.00 KiB/s, done.
Resolving deltas: 100% (1/1), done.
```

Рисунок 2. Клонирование репозитория

3. Дополните файл .gitignore необходимыми правилами для работы с IDE PyCharm.

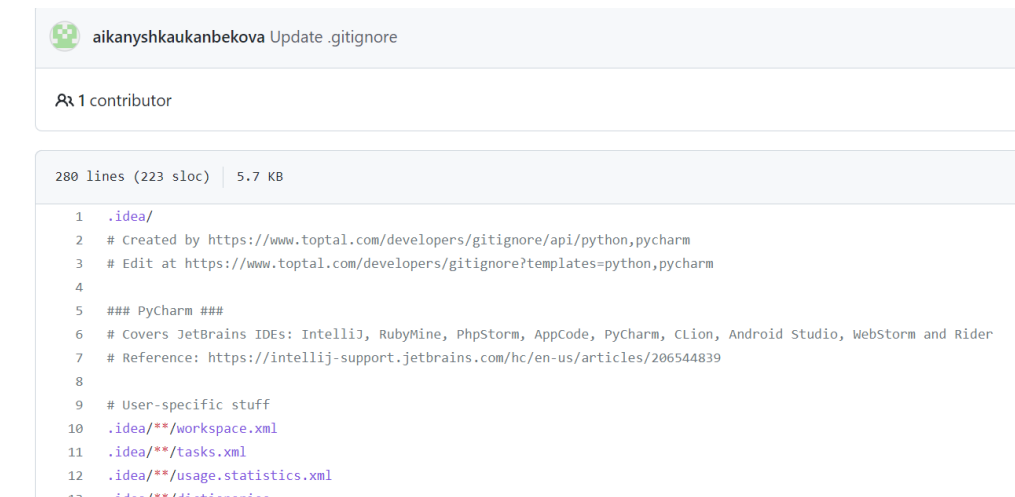


Рисунок 3. Дополнение файла .gitignore

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

```

C:\Users\User\Desktop\2 кypc Python\lab 10\lab-10>git flow init
which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/User/Desktop/2 кypc Python/lab 10/lab-10/.git/hooks]
C:\Users\User\Desktop\2 кypc Python\lab 10\lab-10>

```

Рисунок 4. Организован модель ветвления git flow

5. Индивидуальное задание 1

Вариант 9.

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`. Номер варианта уточнить у преподавателя.

```

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

from fun1 import fun1

a = [1, 2, 34, 54, 36, 7, 8]

```

```

max_fun = fun1()
min_fun = fun1('min')

if __name__ == '__main__':
    print(max_fun(a))
    print(min_fun(a))
print(webpage)

```

Рисунок 5. Выполненное индивидуальное задание 1

6. Индивидуальное задание 2

Вариант 9.

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. Номер варианта уточнить у преподавателя.

```

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

import sys
from ROUTE import get_route, display_route, select_route

if __name__ == '__main__':
    routes = []

    while True:
        command = input(">>> ").lower()
        if command == 'exit':
            break

        elif command == 'add':
            get_route(routes)

        elif command == 'list':
            display_route(routes)

        elif command.startswith('select'):
            parts = command.split(' ', maxsplit=1)
            period = int(parts[1])

            selected = select_route(routes, period)
            display_route(selected)

        elif command == 'help':
            print("Список команд:\n")
            print("add - добавить маршрут;")
            print("list - вывести список маршрутов;")
            print("select <номер маршрута> - запросить данные о маршруте;")
            print("help - отобразить справку;")
            print("exit - завершить работу с программой.")

```

```
else:
    print(f"Неизвестная команда {command}", file=sys.stderr)
```

Рисунки 6. Выполненное индивидуальное задание

7. Сделала коммит, выполнил слияние с веткой main, и запустил изменения в уд. репозиторий.

```
c:\Users\User\Desktop\2 курс Python\lab 16\2.13>git add .
c:\Users\User\Desktop\2 курс Python\lab 16\2.13>git commit -m "new"
[main d1af21e] new
 2 files changed, 52 insertions(+)
 create mode 100644 indiv 1.py
 create mode 100644 indiv 2.py
c:\Users\User\Desktop\2 курс Python\lab 16\2.13>git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 1.04 KiB | 354.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/aikanyshkaukanbekova/2.13.git
 402eb15..d1af21e  main -> main
c:\Users\User\Desktop\2 курс Python\lab 16\2.13>
```

Рисунок 7. Сохранения

Ответы на 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?

После слова import, или через конструкцию from ... 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?

Применяется для полного подключения пакета к программе.

5. Каково назначение переменной __all__ файла __init__.py?

Она хранит список модулей, который импортируется при загрузке через конструкцию.