

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций
Отчет по лабораторной работе № 2.13
«Модули и пакеты»
по дисциплине «Основы программной инженерии»**

Выполнил студент группы

ПИЖ-б-о-21-1

Зиберов Александр

« » декабря 2022 г.

Подпись студента _____

Работа защищена

« » _____ 20__ г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь, 2022

Цель работы:

Приобретение навыков по работе с модулями и пакетами языка программирования Python версии 3.x..

Выполнение работы:

Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT, рисунок 1.

Ссылка: <https://github.com/afk552/lab2.13>

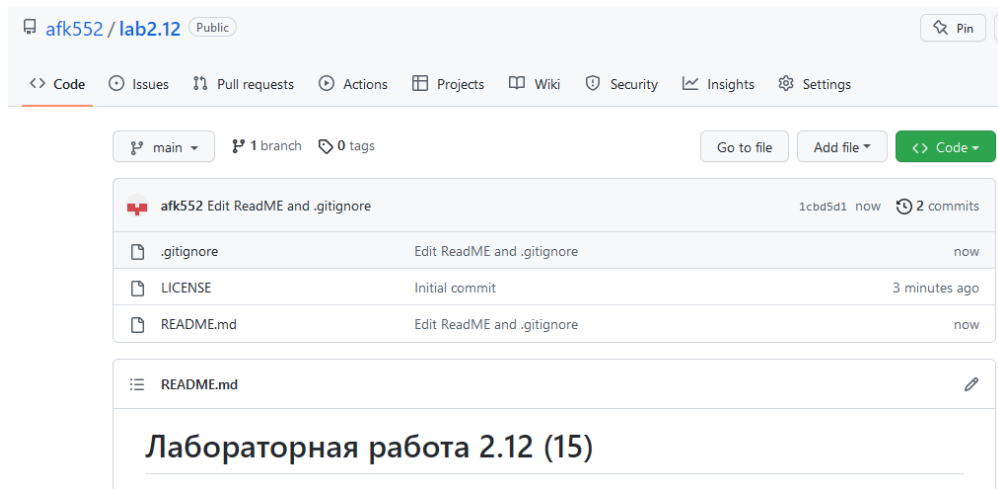


Рисунок 1 – Удаленный репозиторий на GitHub

Дополните файл `.gitignore` необходимыми правилами для работы с IDE PyCharm, рисунок 2.

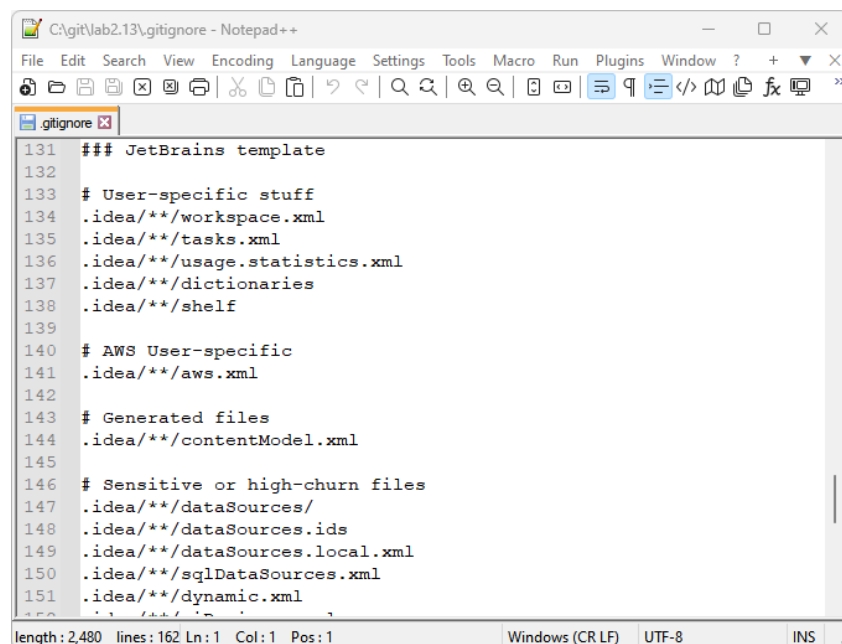


Рисунок 2 – Окно блокнота

Организируйте свой репозиторий в соответствии с моделью ветвления git-flow, рисунок 3.



```
C:\Windows\System32\cmd.exe

C:\git\lab2.13>git checkout -b develop
Switched to a new branch 'develop'

C:\git\lab2.13>git branch
* develop
  main

C:\git\lab2.13>git push origin develop
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/afk552/lab2.13/pull/new/develop
remote:
To https://github.com/afk552/lab2.13
 * [new branch]      develop -> develop

C:\git\lab2.13>
```

Рисунок 3 – Окно командной строки

Создайте проект PyCharm в папке репозитория, рисунок 4.

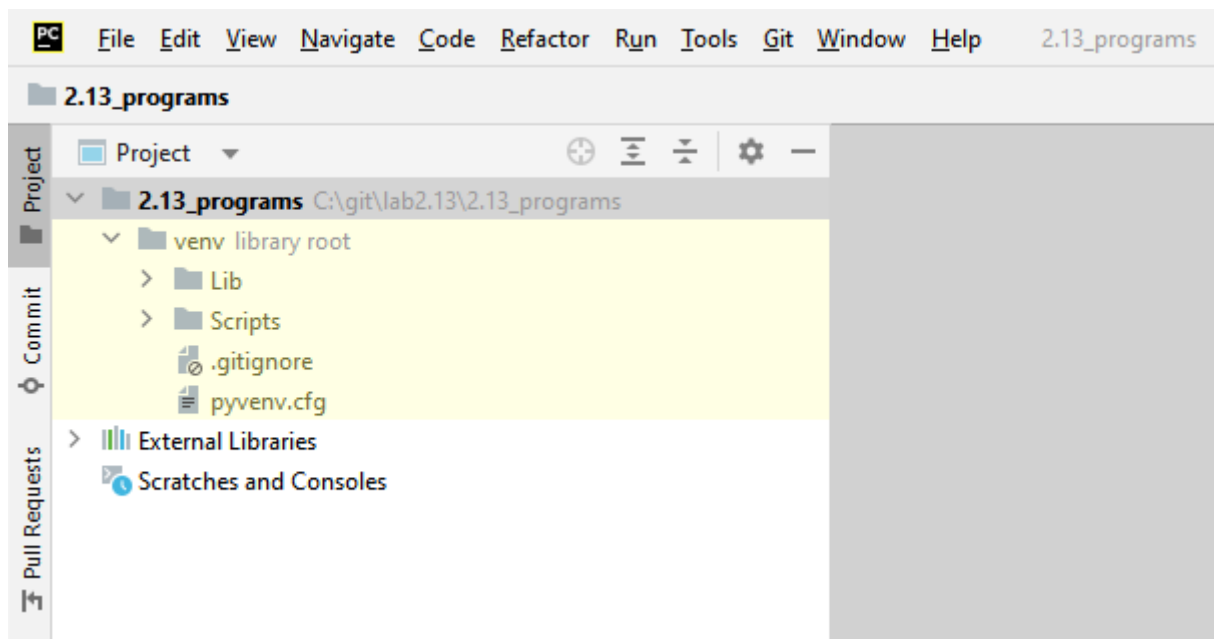
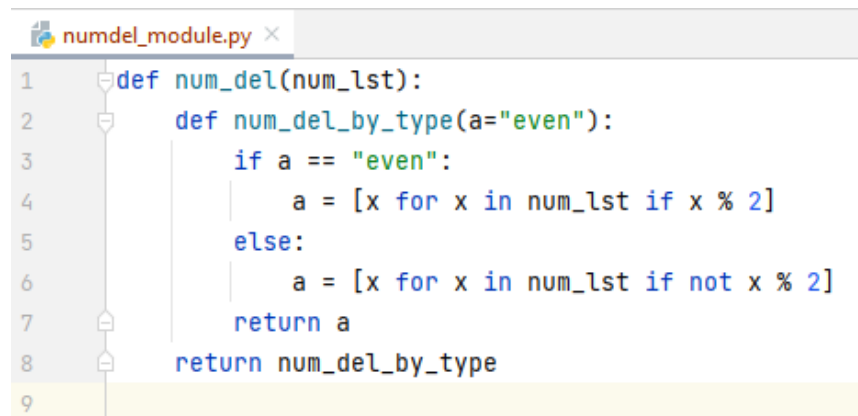


Рисунок 4 – Окно проекта в PyCharm

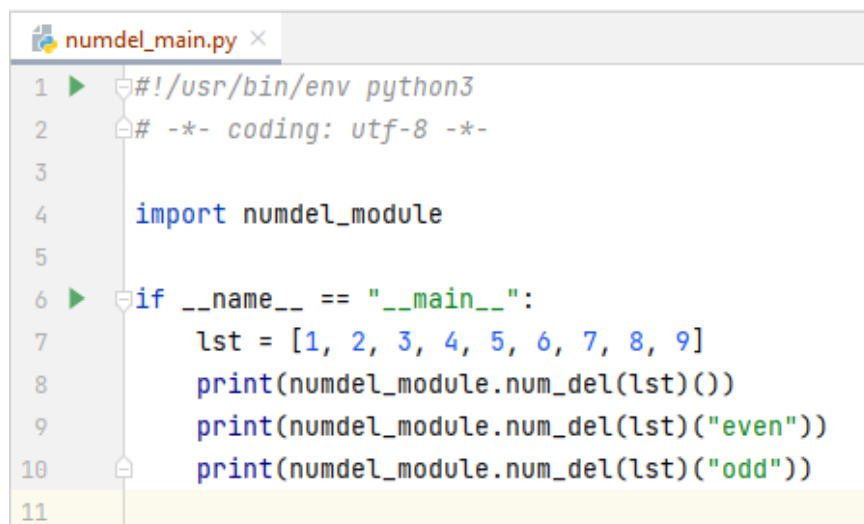
Индивидуальное задание.

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`. Номер варианта уточнить у преподавателя.



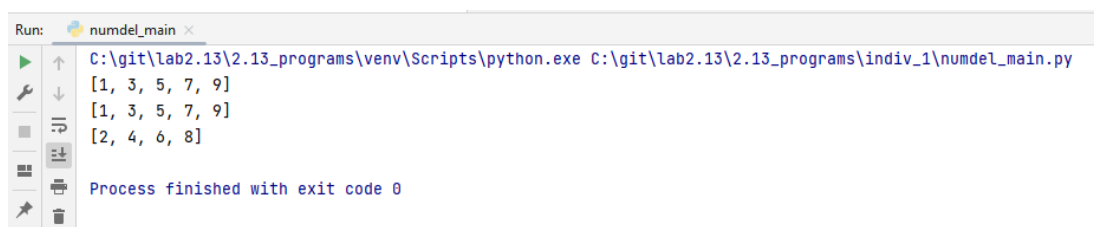
```
1 def num_del(num_lst):
2     def num_del_by_type(a="even"):
3         if a == "even":
4             a = [x for x in num_lst if x % 2]
5         else:
6             a = [x for x in num_lst if not x % 2]
7         return a
8     return num_del_by_type
9
```

Рисунок 5 – Код модуля `numdel_module.py` программы задания 1



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import numdel_module
5
6 if __name__ == "__main__":
7     lst = [1, 2, 3, 4, 5, 6, 7, 8, 9]
8     print(numdel_module.num_del(lst)())
9     print(numdel_module.num_del(lst)("even"))
10    print(numdel_module.num_del(lst)("odd"))
11
```

Рисунок 6 – Код программы с использованием модуля `numdel_module.py`



```
Run: numdel_main x
C:\git\lab2.13\2.13_programs\venv\Scripts\python.exe C:\git\lab2.13\2.13_programs\indiv_1\numdel_main.py
[1, 3, 5, 7, 9]
[2, 4, 6, 8]
[1, 3, 5, 7, 9]
Process finished with exit code 0
```

Рисунок 7 – Результат выполнения программы с использованием модуля `numdel_module.py`

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. Номер варианта уточнить у преподавателя.

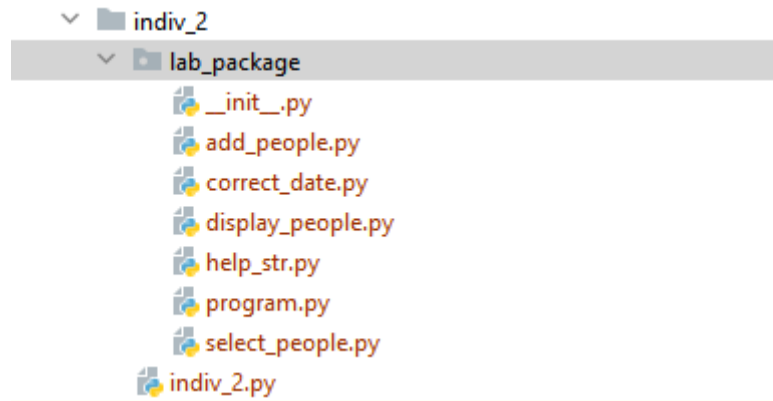


Рисунок 8 – Структура программы с пакетом `lab_package`



Рисунок 9 – Файл `__init__.py`

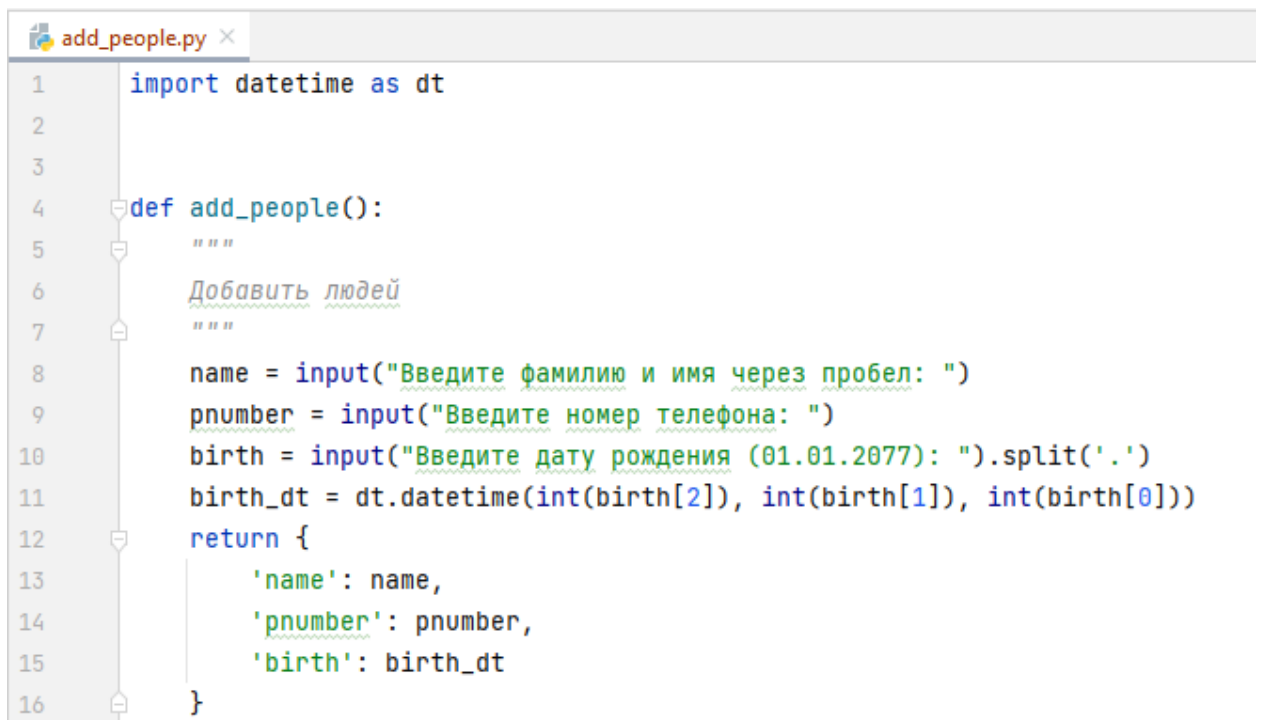


Рисунок 10 – Файл `add_people.py`

```

correct_date.py x
1 def correct_date(print_month):
2     """
3     Скорректировать номер месяца
4     """
5     month_by_text = {
6         "январь": "01",
7         "февраль": "02",
8         "март": "03",
9         "апрель": "04",
10        "май": "05",
11        "июнь": "06",
12        "июль": "07",
13        "август": "08",
14        "сентябрь": "09",
15        "октябрь": "10",
16        "ноябрь": "11",
17        "декабрь": "12",
18    }
19    if print_month.isalpha():
20        print_month.lower()
21        for key, value in month_by_text.items():
22            if key == print_month:
23                print_month = value
24    if len(print_month) == 1:
25        return "0" + print_month
26    else:
27        return print_month
28

```

Рисунок 11 – Файл correct_date.py

```

select_people.py x
1 def select_people(people_list, correct_printed_month):
2     """
3     Выбрать людей по заданному месяцу рождения
4     """
5     result = []
6     for person in people_list:
7         birth = person.get('birth')
8         if correct_printed_month == birth.strftime("%m"):
9             result.append(person)
10    return result

```

Рисунок 12 – Файл select_people.py

```
display_people.py x
1 def display_people(people_list):
2     """
3     Вывести людей из списка
4     """
5     if people_list:
6         line = '+-{}-+-{}-+-{}-+-{}-+'.format(
7             '-' * 4,
8             '-' * 30,
9             '-' * 14,
10            '-' * 19
11        )
12        print(line)
13        print(
14            '| {:^4} | {:^30} | {:^14} | {:^19} |'.format(
15                "№п/п",
16                "Фамилия Имя",
17                "Номер телефона",
18                "Дата рождения"
19            )
20        )
21        print(line)
22        for nmb, person in enumerate(people_list, 1):
23            print(
24                '| {:>4} | {:<30} | {:<14} | {:>19} |'.format(
25                    nmb,
26                    person.get('name', ''),
27                    person.get('pnumber', ''),
28                    person.get('birth', '').strftime("%d.%m.%Y")
29                )
30            )
31            print(line)
32    else:
33        print("Список людей пуст!")
```

Рисунок 13 – Файл display_people.py

```
help_str.py x
1 def help_str():
2     print("Список доступных команд:")
3     print("add - добавить человека;")
4     print("list - вывести список людей;")
5     print("select <месяц> ('Январь' / '01') - запросить людей, чьи "
6           "дни рождения приходятся на указанный месяц;")
7     print("help - отобразить справку;")
8     print("exit - завершить работу с программой.")
9
```

Рисунок 14 – Файл help_str.py

```
add_people.py x
1 import datetime as dt
2
3
4 def add_people():
5     """
6     Добавить людей
7     """
8     name = input("Введите фамилию и имя через пробел: ")
9     pnumber = input("Введите номер телефона: ")
10    birth = input("Введите дату рождения (01.01.2077): ").split('.')
11    birth_dt = dt.datetime(int(birth[2]), int(birth[1]), int(birth[0]))
12    return {
13        'name': name,
14        'pnumber': pnumber,
15        'birth': birth_dt
16    }
```

Рисунок 15 – Файл add_people.py

```
program.py x
1 from lab_package.add_people import add_people
2 from lab_package.correct_date import correct_date
3 from lab_package.display_people import display_people
4 from lab_package.help_str import help_str
5 from lab_package.select_people import select_people
6 import sys
7
8
9 def main():
10    people = []
11    print("Программа запущена, введите help для просмотра команд!")
12
13    while True:
14        command = input(">>> ").lower()
15
16        if command == "exit":
17            break
18
19        elif command == "add":
20            person = add_people()
21            people.append(person)
22            if len(people) > 1:
23                people.sort(key=lambda item: item.get('name', ''))
24
25        elif command == "list":
26            display_people(people)
27
28        elif command.startswith('select '):
29            parts = command.split(' ', maxsplit=1)
30            printed_month = parts[1]
31            corrected_month = correct_date(printed_month)
32            selected = select_people(people, corrected_month)
33            if len(selected) > 0:
34                display_people(selected)
35            else:
36                print("Людей, чьи дни рождения приходятся на указанный месяц нет!")
37
38        elif command == 'help':
39            help_str()
40
41        else:
42            print(f"Неизвестная команда: {command}", file=sys.stderr)
```

Рисунок 16 – Файл program.py


```
indiv_2.py x
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  from lab_package import *
6
7  if __name__ == '__main__':
8      program.main()
9
```

Рисунок 16 – Файл indiv_2.py за папкой пакета

```
Run: indiv_2 x
C:\git\lab2.13\2.13_programs\venv\Scripts\python.exe C:\git\lab2.13\2.13_programs\indiv_2\indiv_2.py
Программа запущена, введите help для просмотра команд!
>>> help
Список доступных команд:
add - добавить человека;
list - вывести список людей;
select <месяц> ('Январь' / '01') - запросить людей, чьи дни рождения приходятся на указанный месяц;
help - отобразить справку;
exit - завершить работу с программой.
>>> add
Введите фамилию и имя через пробел: Фамилия Имя
Введите номер телефона: +79734727427
Введите дату рождения (01.01.2077): 03.01.2000
>>> list
+-----+-----+-----+-----+
| №п/п |      Фамилия Имя      | Номер телефона |   Дата рождения   |
+-----+-----+-----+-----+
|    1 | Фамилия Имя          | +79734727427   |    03.01.2000    |
+-----+-----+-----+-----+
>>> select 01
+-----+-----+-----+-----+
| №п/п |      Фамилия Имя      | Номер телефона |   Дата рождения   |
+-----+-----+-----+-----+
|    1 | Фамилия Имя          | +79734727427   |    03.01.2000    |
+-----+-----+-----+-----+
>>> select 02
Людей, чьи дни рождения приходятся на указанный месяц нет!
>>> exit

Process finished with exit code 0
```

Рисунок 17 – Результат выполнения задания 2

Выполните слияние ветки для разработки с веткой main / master.
Отправьте сделанные изменения на сервер GitHub.

```
(venv) PS C:\git\lab2.13\2.13_programs> git checkout main
2.13_programs/indiv_2/lab_package/__init__.py | 1 +
2.13_programs/indiv_2/lab_package/add_people.py | 16 ++++++++
2.13_programs/indiv_2/lab_package/correct_date.py | 27 ++++++++
.../indiv_2/lab_package/display_people.py | 33 ++++++++
2.13_programs/indiv_2/lab_package/help_str.py | 8 +++++
2.13_programs/indiv_2/lab_package/program.py | 42 ++++++++
2.13_programs/indiv_2/lab_package/select_people.py | 10 +++++
10 files changed, 163 insertions(+)
create mode 100644 2.13_programs/indiv_1/numdel_main.py
create mode 100644 2.13_programs/indiv_1/numdel_module.py
create mode 100644 2.13_programs/indiv_2/indiv_2.py
create mode 100644 2.13_programs/indiv_2/lab_package/__init__.py
create mode 100644 2.13_programs/indiv_2/lab_package/add_people.py
create mode 100644 2.13_programs/indiv_2/lab_package/correct_date.py
create mode 100644 2.13_programs/indiv_2/lab_package/display_people.py
create mode 100644 2.13_programs/indiv_2/lab_package/help_str.py
create mode 100644 2.13_programs/indiv_2/lab_package/program.py
create mode 100644 2.13_programs/indiv_2/lab_package/select_people.py
(venv) PS C:\git\lab2.13\2.13_programs> git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/afk552/lab2.13
5510a11..5307aec main -> main
(venv) PS C:\git\lab2.13\2.13_programs> █
```

Рисунок 18 – Окно терминала

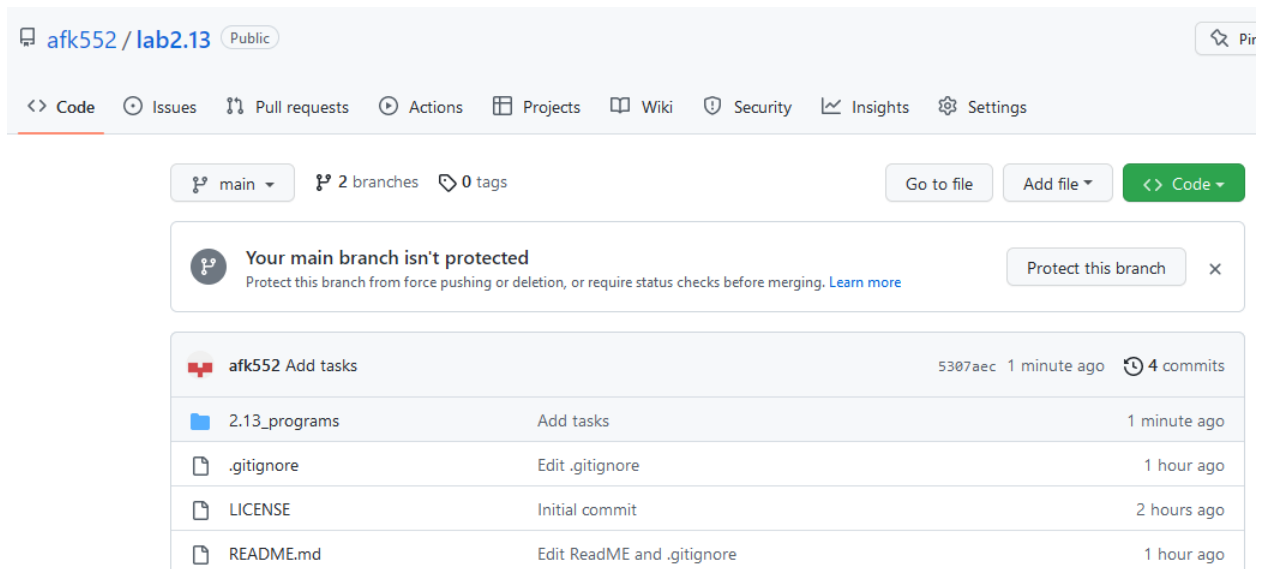


Рисунок 19 – Удаленный репозиторий на GitHub

Вывод: В результате выполнения работы были изучены модули и пакеты, их подключение и использование в программах на языке Python.

Контрольные вопросы:

1. Что является модулем языка Python?

Под модулем в Python понимается файл с расширением .py. Модули предназначены для того, чтобы в них хранить часто используемые функции, классы, константы и т. п. Можно условно разделить модули и программы: программы предназначены для непосредственного запуска, а модули для импортирования их в другие программы. Стоит заметить, что модули могут быть написаны не только на языке Python, но и на других языках (например C).

2. Какие существуют способы подключения модулей в языке Python?

```
import <name>; import <name1>, <name2>; import <name> as  
<new_name>; from <module> import <name1>,<name2>.
```

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 . Пакеты используются для формирования пространства имен, что позволяет работать с модулями через указание уровня вложенности (через точку).

4. Каково назначение файла __init__.py ?

В __init__.py файл заставляет Python рассматривать каталоги, содержащие его, как модули. Кроме того, это первый файл, загружаемый в модуль, поэтому вы можете использовать его для выполнения кода, который хотите запускать каждый раз при загрузке модуля, или для указания экспортируемых подмодулей.

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

Файл `__init__.py` может быть пустым или может содержать переменную `__all__`, хранящую список модулей, который импортируется при загрузке через конструкцию `__all__ = ["name", "name2"]`.