

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО РАБОТЕ №2.7
дисциплины «Основы кроссплатформенного программирования»

Выполнил:
Кондратенко Даниил Витальевич
1 курс, группа ИТС-б-о-22-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., канд. тех. наук, доцент,
доцент кафедры инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: работа с множествами в языке Python.

Цель работы: приобретение навыков по работе с множествами при написании программ с помощью языка программирования Python версии 3.x.

Порядок выполнения работы:

Задание 1.

Изучил теоретический материал работы, создал общедоступный репозиторий на GitHub, в котором использована лицензий MIT и язык программирования Python, также добавил файл .gitignore с необходимыми правилами.

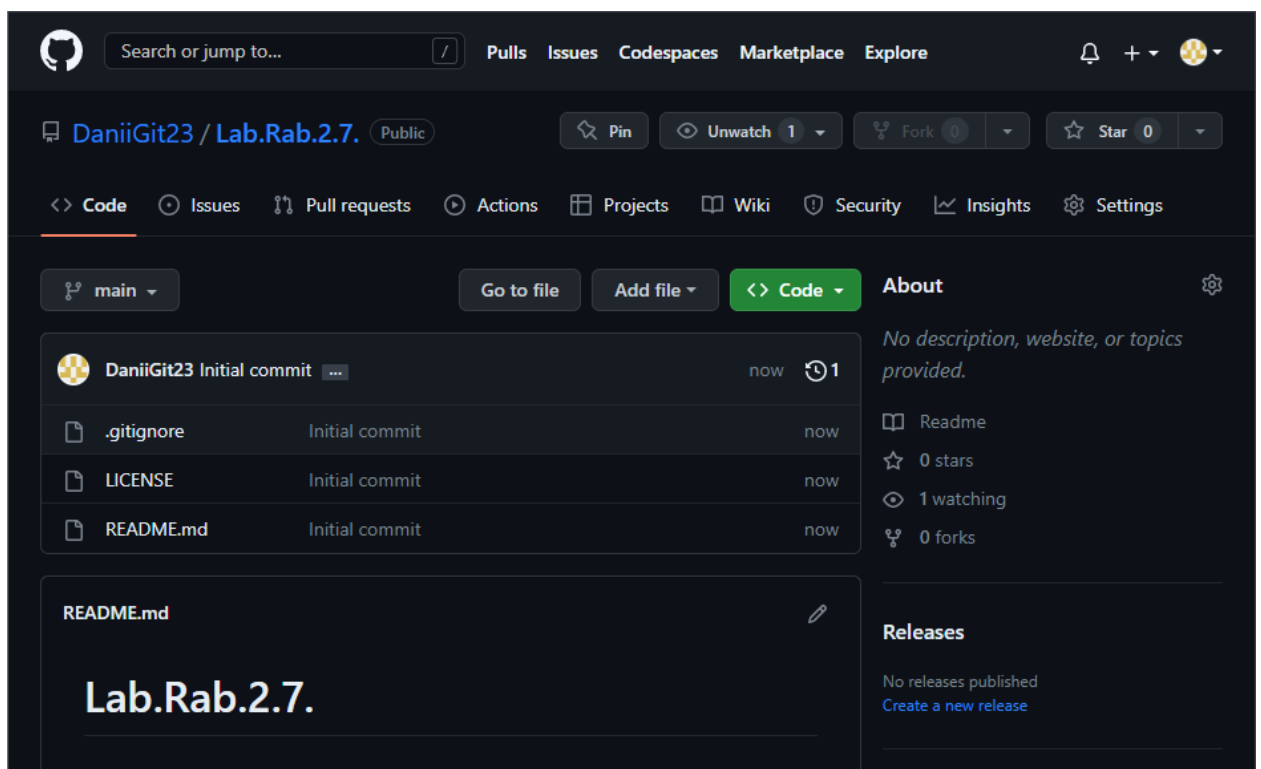


Рисунок 1 – Новый репозиторий

Задание 2.

Проклонировал свой репозиторий на свой компьютер.

Организовал свой репозиторий в соответствии с моделью ветвления git-flow, появилась новая ветка develop.

```

C:\Users\HUAWEI>git clone https://github.com/DaniiGit23/Lab.Rab.2.7.git
Cloning into 'Lab.Rab.2.7'...
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.

C:\Users\HUAWEI>cd C:\Users\HUAWEI\Lab.Rab.2.7

C:\Users\HUAWEI\Lab.Rab.2.7>git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

C:\Users\HUAWEI\Lab.Rab.2.7>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? [] t
Hooks and filters directory? [C:/Users/HUAWEI/Lab.Rab.2.7/.git/hooks]

```

Рисунок 2 – Клонирование и модель ветвления git-flow

Реализовывал примеры и индивидуальные задания на основе ветки develop, без создания дополнительной ветки feature/(название ветки) по указанию преподавателя.

Задание 3.

Создал проект PyCharm в папке репозитория.

Работа с примером №1.

Добавил новый файл *primer1.py*.

Условие примера: определить результат выполнения операций над множествами. Считать элементы множества строками.

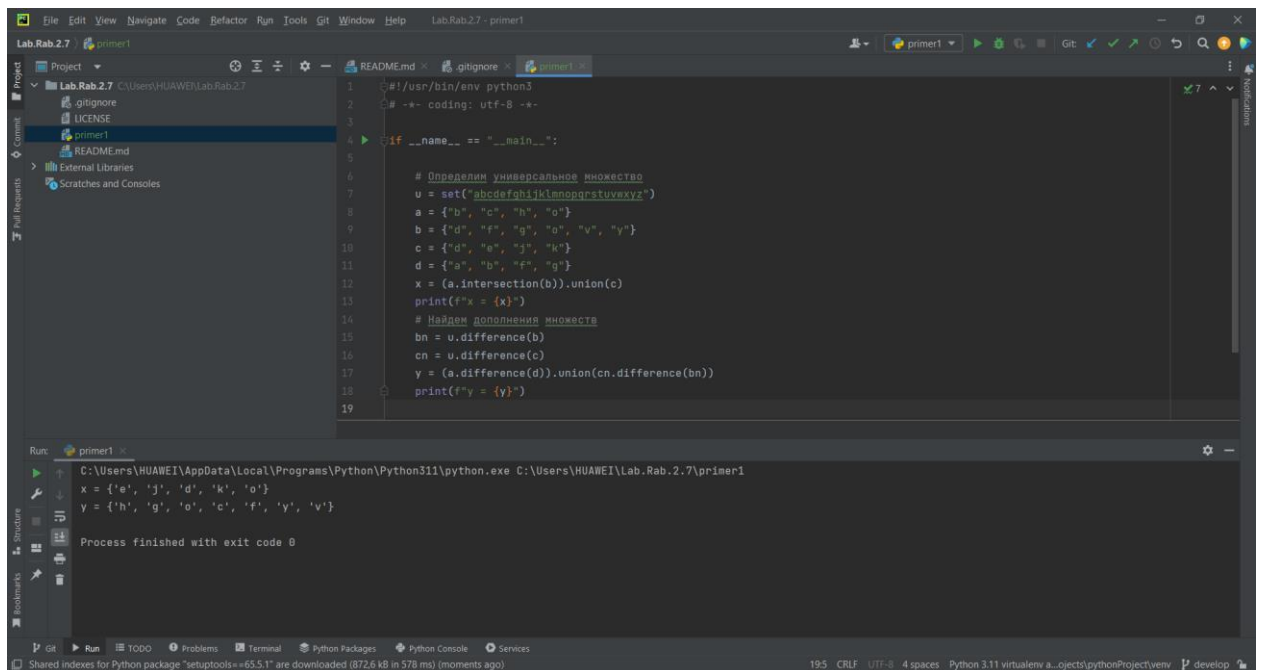


Рисунок 3 – Программа и ее результат

Зафиксировал данные изменения.

Задание 4.

Создал новый файл под названием *ObsheeZadaniye1.py*

Работа с общим заданием №1.

Условие примера: подсчитайте количество гласных в строке, введенной с клавиатуры с использованием множеств.

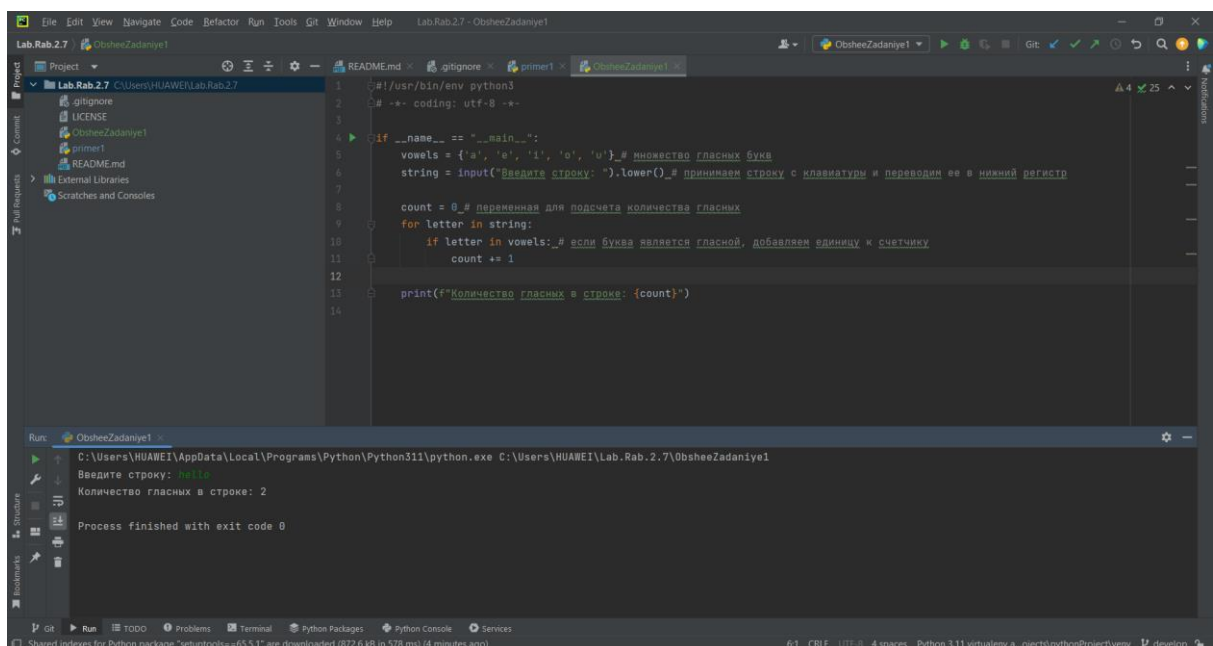


Рисунок 4 – Программа и ее результат

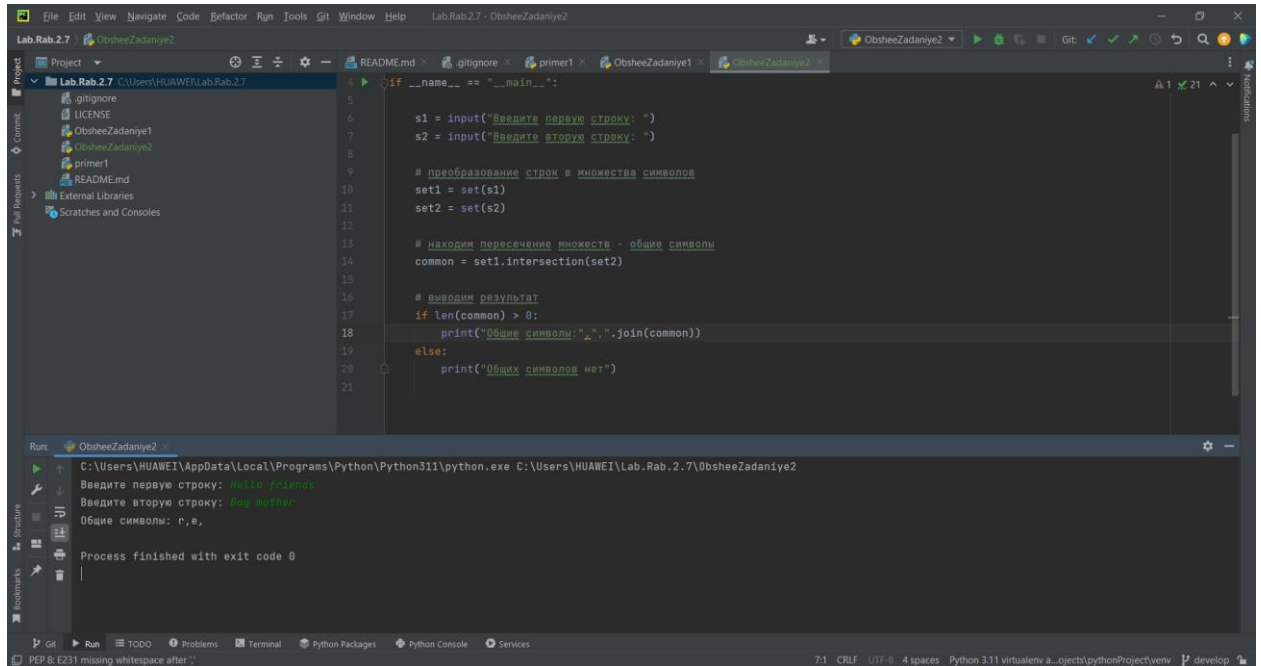
Зафиксировал данные изменения.

Задание 4.

Создал новый файл под названием *ObsheeZadaniye2.py*

Работа с общим заданием №2.

Условие примера: определите общие символы в двух строках, введенных с клавиатуры.



```
Lab.Rab.2.7 - ObsheeZadaniye2
File Edit View Navigate Code Refactor Run Tools Git Window Help
Project
  Lab.Rab.2.7 C:\Users\HUAWEI\Lab.Rab.2.7
    .gitignore
    LICENSE
    ObsheeZadaniye1
    ObsheeZadaniye2
    primer1
    README.md
    External Libraries
    Scratches and Consoles
  Run: ObsheeZadaniye2
    C:\Users\HUAWEI\AppData\Local\Programs\Python\Python311\python.exe C:\Users\HUAWEI\Lab.Rab.2.7\ObsheeZadaniye2
    Введите первую строку: Hello friends
    Введите вторую строку: Ray mother
    Общие символы: r,e,
    Process finished with exit code 0
  7:1 CRLF UTF-8 4 spaces Python 3.11 virtualenv a... pythonProject\venv develop
```

```
if __name__ == "__main__":
    s1 = input("Введите первую строку: ")
    s2 = input("Введите вторую строку: ")

    # преобразование строк в множества символов
    set1 = set(s1)
    set2 = set(s2)

    # находим пересечение множеств - общие символы
    common = set1.intersection(set2)

    # выводим результат
    if len(common) > 0:
        print("Общие символы: ", ".join(common))
    else:
        print("Общих символов нет")
```

Рисунок 5 – Программа и ее результат

Зафиксировал данные изменения.

Задание 5.

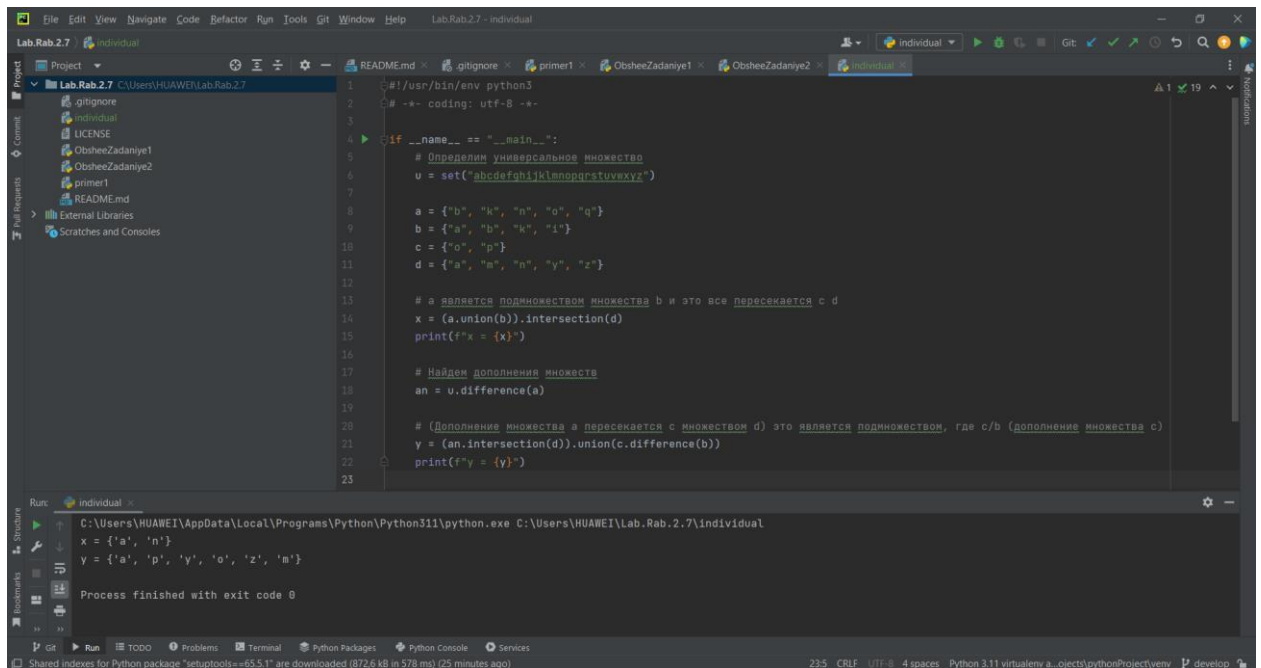
Создал новый файл под названием *individual.py*

Работа с индивидуальным заданием.

Условие примера:

$$A = \{b, k, n, o, q\}; \quad B = \{a, b, k, u\}; \quad C = \{o, p\}; \quad D = \{a, m, n, y, z\};$$
$$X = (A \cup B) \cap D; Y = (\bar{A} \cap D) \cup (C/B).$$

12.



The screenshot shows a code editor with a file named `individual.py`. The code defines sets `a`, `b`, `c`, and `d` based on the problem statement. It then calculates `x` as the intersection of the union of `a` and `b` with `d`, and `y` as the union of the intersection of the complement of `a` with `d` and the set difference `c/b`. The output of the script is displayed in the Run console.

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == "__main__":
5     # Определим универсальное множество
6     u = set("abcdefghijklmnopqrstuvwxyz")
7
8     a = {"b", "k", "n", "o", "q"}
9     b = {"a", "b", "k", "i"}
10    c = {"o", "p"}
11    d = {"a", "m", "n", "y", "z"}
12
13    # a является подмножеством множества b и это все пересекется с d
14    x = (a.union(b)).intersection(d)
15    print(f"x = {x}")
16
17    # Найдём дополнения множеств
18    an = u.difference(a)
19
20    # (Дополнение множества a пересекется с множеством d) это является подмножеством, где c/b (дополнение множества c)
21    y = (an.intersection(d)).union(c.difference(b))
22    print(f"y = {y}")
23
```

Run: individual

```
C:\Users\HUAWEI\AppData\Local\Programs\Python\Python311\python.exe C:\Users\HUAWEI\Lab.Rab.2.7\individual
x = {'a', 'n'}
y = {'a', 'p', 'y', 'o', 'z', 'm'}
```

Process finished with exit code 0

Рисунок 6 – Программа и ее результат

Зафиксировал данные изменения.

Задание 6.

Слил ветку develop с веткой main и отправил на удаленный сервер.

```
C:\Users\HUAWEI\Lab.Rab.2.7>git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

C:\Users\HUAWEI\Lab.Rab.2.7>git merge develop
Merge made by the 'ort' strategy.
 ObsheeZadaniye1 | 13 ++++++++
 ObsheeZadaniye2 | 20 ++++++++
 individual       | 22 ++++++++
 primer1          | 18 ++++++++
4 files changed, 73 insertions(+)
create mode 100644 ObsheeZadaniye1
create mode 100644 ObsheeZadaniye2
create mode 100644 individual
create mode 100644 primer1

C:\Users\HUAWEI\Lab.Rab.2.7>git status
On branch main
Your branch is ahead of 'origin/main' by 7 commits.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\Users\HUAWEI\Lab.Rab.2.7>git push
Enumerating objects: 20, done.
Counting objects: 100% (20/20), done.
Delta compression using up to 8 threads
Compressing objects: 100% (18/18), done.
Writing objects: 100% (18/18), 2.88 KiB | 2.88 MiB/s, done.
Total 18 (delta 8), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (8/8), completed with 1 local object.
To https://github.com/DaniiGit23/Lab.Rab.2.7.git
33a2311..e5d549d main -> main
```

Рисунок 7 – Слияние веток

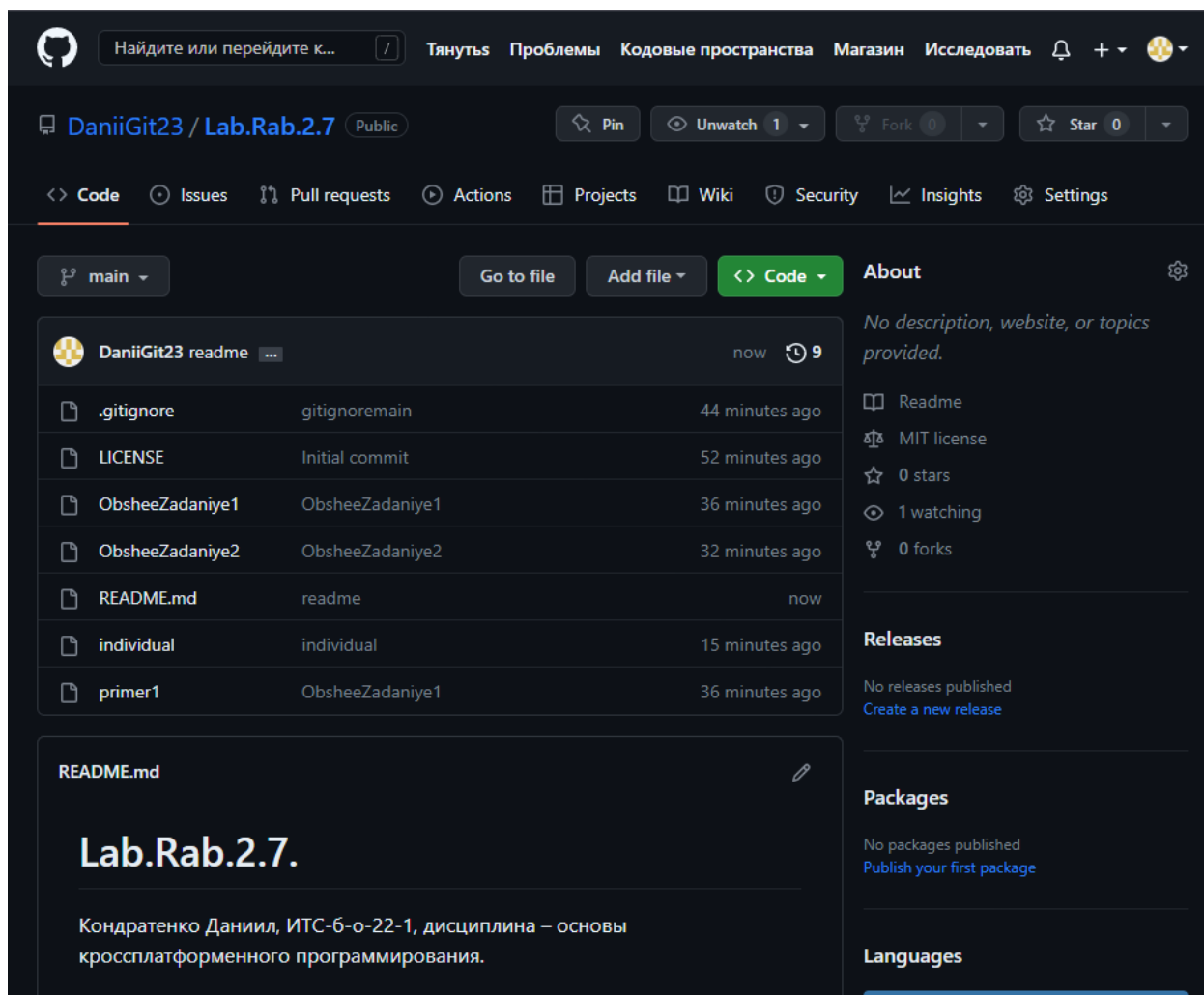


Рисунок 8 – Удаленный сервер

Ссылка на репозиторий: <https://github.com/DaniiGit23/Lab.Rab.2.7.git>

Ответы на контрольные вопросы:

1) Что такое множества в языке Python?

Множеством в языке программирования Python называется неупорядоченная совокупность уникальных значений. В качестве элементов этого набора данных могут выступать любые неизменяемые объекты, такие как числа, символы, строки. В отличие от массивов и списков, порядок следования значений не учитывается при обработке его содержимого. Над одним, а также несколькими множествами можно выполнять ряд операций, благодаря функциям стандартной библиотеки языка программирования Python.

2) Как осуществляется создание множеств в Python?

Перед тем как начать работу с множеством, необходимо для начала его создать. Сделать это можно, просто присвоив переменной

последовательность значений, выделив их фигурными скобками. Следующий пример показывает код, в котором создается

множество целых чисел под названием `a`, после функция `print` выводит на экран его содержимое.

Как можно заметить, все элементы полученной последовательности являются уникальными, без повторений. Существует и другой

способ создания множеств, который подразумевает использование вызова `set`. Аргументом этой функции может быть набор

неких данных или даже строка с текстом, как это показано в следующем примере.

3) Как проверить присутствие/отсутствие элемента в множестве?

Для проверки присутствия/отсутствия элемента в множестве в Python используется ключевое слово `in`.

4) Как выполнить перебор элементов множества?

Перебор элементов множества можно выполнить с помощью цикла `for`. Вот пример кода, который позволяет перебрать все элементы множества и вывести их на экран.

5) Что такое `set comprehension`?

`Set comprehension` (генератор множеств) - это способ создания и заполнения множества в Python одной строки кода.

Он похож на список и словарь `comprehensions`, но вместо списка или словаря, генератор множества создает множество.

6) Как выполнить добавление элемента во множество?

Для добавления элемента во множество в Python используется метод `add()`.

7) Как выполнить удаление одного или всех элементов множества?

Для удаления одного элемента из множества в Python используется метод `remove()` или `discard()`. Разница между ними заключается в том, что если

попытаться удалить несуществующий элемент соответствующим методом, то `remove()` выбросит ошибку `KeyError`, а `discard()` ничего не сделает.

Для удаления всех элементов множества можно использовать метод `clear()`.

8) Как выполняются основные операции над множествами: объединение, пересечение, разность?

Основные операции над множествами в Python:

- Объединение (`union`) - создает новое множество, содержащее все элементы из двух или более множеств.

- Пересечение (`intersection`) - создает новое множество, содержащее только те элементы, которые присутствуют в обоих множествах.

- Разность (`difference`) - создает новое множество, состоящее из элементов первого множества, которые не присутствуют во втором множестве.

9) Как определить, что некоторое множество является надмножеством или подмножеством другого множества?

Для определения, является ли некоторое множество подмножеством или надмножеством другого множества в Python, можно воспользоваться методами `issubset()` и `issuperset()` соответственно.

Метод `issubset()` проверяет, является ли одно множество подмножеством другого. Если все элементы первого множества присутствуют во втором множестве, то метод вернет `True`, иначе `False`.

Метод `issuperset()` проверяет, является ли одно множество надмножеством другого. Если все элементы второго множества присутствуют в первом множестве, то метод вернет `True`, иначе `False`.

10) Каково назначение множеств `frozenset` ?

`frozenset` - это неизменяемый тип данных в Python, который является аналогом множества (`set`). Его элементы не могут быть изменены после создания `frozenset`.

Множества `frozenset` используются там, где требуется неизменяемость данных, например, в качестве ключей для других множеств и словарей.

За счет неизменяемости `frozenset` в него нельзя добавлять, удалять или изменять элементы. Поэтому `frozenset` не имеет методов, связанных с изменением данных.

11) Как осуществляется преобразование множеств в строку, список, словарь?

Для преобразования множества в строку, список или словарь в языке Python можно использовать следующие функции:

- `str()` - для преобразования множества в строку
- `list()` - для преобразования множества в список
- `dict.fromkeys()` - для преобразования множества в словарь