

**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 ПРАКТИЧЕСКОЙ РАБОТЕ №2.7  
дисциплины  
«Основы кроссплатформенного программирования»**

Выполнила студентка группы  
ИВТ-б-о-21-1  
Яковлева Е.А. « » \_\_\_\_\_ 20\_\_ г.  
Подпись студента \_\_\_\_\_  
Работа защищена  
« » \_\_\_\_\_ 20\_\_ г.

Проверил доцент  
Кафедры инфокоммуникаций,  
старший преподаватель  
Воронкин Р.А.

\_\_\_\_\_  
(подпись)

Ставрополь, 2022 г.

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

**Порядок выполнения работы:**

1. Создала общедоступный репозиторий на GitHub, в котором использованы лицензия MIT и язык программирования Python.

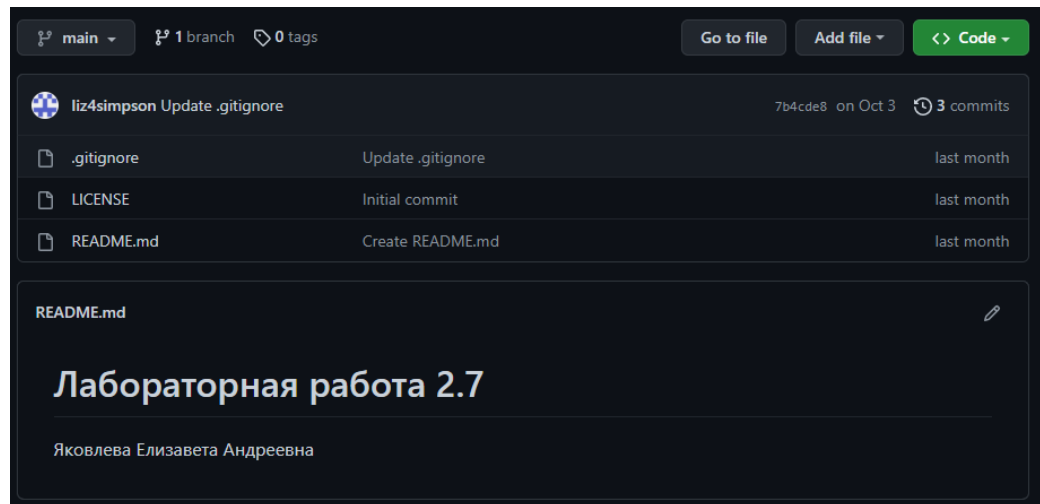


Рисунок 1. Создание репозитория

Выполнила клонирование созданного репозитория.

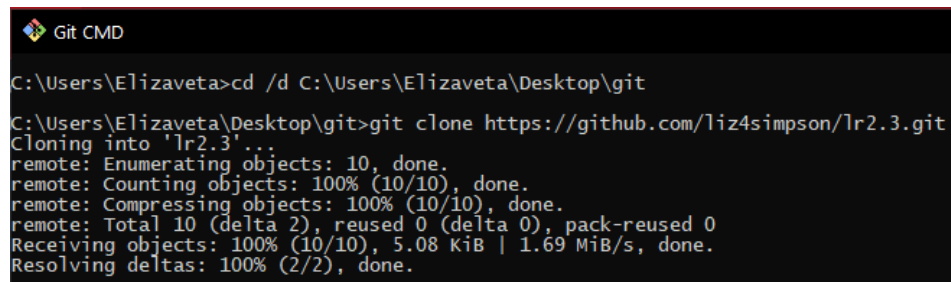


Рисунок 2. Клонирование репозитория

Организовала свой репозиторий в соответствии с моделью ветвления git-flow.

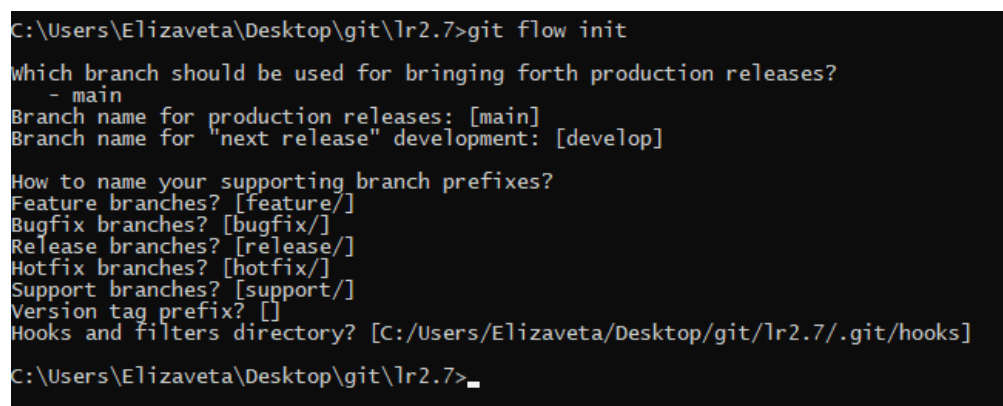


Рисунок 3. Организация репозитория согласно модели ветвления git-flow

2. Создала проект PyCharm в папке репозитория, проработала примеры из лабораторной работы.

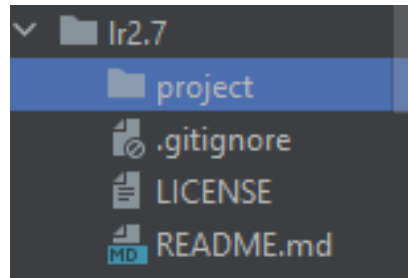


Рисунок 4. Проект PyCharm

```
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", "c", "h", "o"}
9      b = {"d", "f", "g", "o", "v", "y"}
10     c = {"d", "e", "j", "k"}
11     d = {"a", "b", "f", "g"}
12
if __name__ == "__main__"
```

ind × ind × zad2 × zad1 × zad2 ×

C:\nana\Desktop\Scripts\python.exe C:\Users\EL

x = {'d', 'j', 'k', 'e', 'o'}

y = {'g', 'y', 'h', 'o', 'f', 'v', 'c'}

Process finished with exit code 0

Рисунок 5. Результат выполнения примера

2. Решите задачу: подсчитайте количество гласных в строке, введенной с клавиатуры с использованием множеств

```
↑ C:\nana\Desktop\Scripts\python.exe C:\Users\EL
↓ Введите строку: привет, я тебя люблю
Кол-во гласных: 7
```

Рисунок 6. Результат выполнения задачи

3. Решите задачу: определите общие символы в двух строках, введенных с клавиатуры.

```

Введите первую строку: Шла Саша по шоссе
Введите вторую строку: И сосала сушку
Общие символы = {'о', ' ', 'с', 'ш', 'л', 'а'}

Process finished with exit code 0

```

Рисунок 7. Результат выполнения задачи

### 3. Выполнила индивидуальное задание (в26)

```

9      a = {"c", "d", "k", "l", "m", "z"}
10     b = {"b", "c", "d", "n", "w"}
11     c = {"m", "n", "y"}
12     d = {"b", "j", "l", "r", "s", "w", "x"}

14     x = (a.union(b)).intersection(c)
15     print(f"x = {x}")

17     cn = u.difference(c)
18     bn = u.difference(b)

19     |
20     y = (a.difference(d)).union(cn.difference(bn))
21     print(f"y = {y}")

if __name__ == "__main__"

```

ind × ind × zad2 × zad1 × zad2 × example1 ×

```

C:\nana\Desktop\Scripts\python.exe C:\Users\Elizaveta\Desktop\git\lr2
x = {'n', 'm'}
y = {'w', 'c', 'b', 'k', 'z', 'd', 'm'}

Process finished with exit code 0

```

Рисунок 7. Индивидуальное задание

4. Сделала коммит, выполнила слияние с веткой main, и запустила изменения в удаленный репозиторий.

```

C:\Users\Elizaveta\Desktop\git\lr2.7>git add .
C:\Users\Elizaveta\Desktop\git\lr2.7>git commit -m "add"
[develop c0e8d85] add
4 files changed, 88 insertions(+)
create mode 100644 project/example/example1.py
create mode 100644 project/ind/ind.py
create mode 100644 project/zad/zad1.py
create mode 100644 project/zad/zad2.py

```

Рисунок 9. Фиксация и коммит файлов

```
C:\Users\Elizaveta\Desktop\git\lr2.7>git merge develop
Updating 7b4cde8..c0e8d85
Fast-forward
 project/example/example1.py | 21 ++++++
 project/ind/ind.py           | 21 ++++++
 project/zad/zad1.py          | 29 ++++++
 project/zad/zad2.py          | 17 ++++++
 4 files changed, 88 insertions(+)
 create mode 100644 project/example/example1.py
 create mode 100644 project/ind/ind.py
 create mode 100644 project/zad/zad1.py
 create mode 100644 project/zad/zad2.py
```

Рисунок 10. Слияние ветки develop с main

```
C:\Users\Elizaveta\Desktop\git\lr2.7>git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/liz4simpson/lr2.7.git
 7b4cde8..c0e8d85  main -> main
```

Рисунок 11. Отправка изменений на удаленный репозиторий

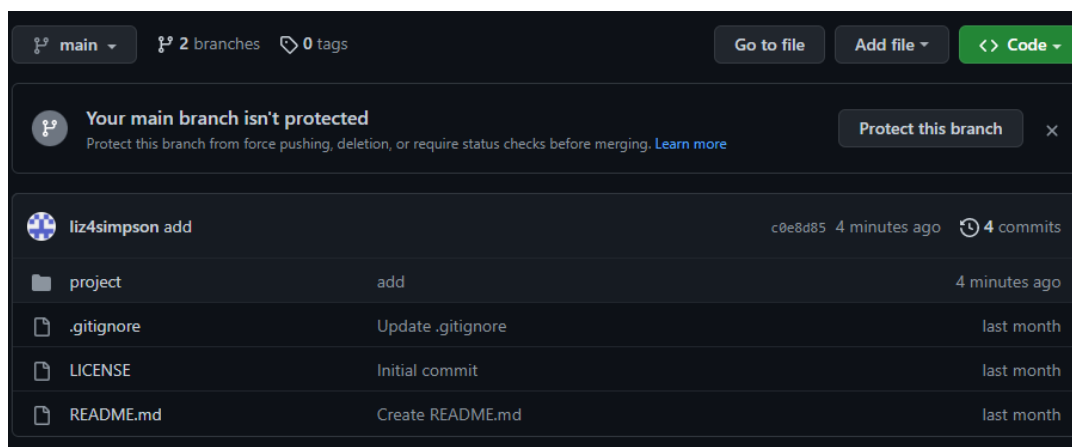


Рисунок 12. Изменения на удаленном репозитории

## Контрольные вопросы

### 1. Что такое множества в языке Python?

Множеством в языке программирования Python называется неупорядоченная совокупность уникальных значений.

### 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. Существует и другой способ создания множеств, который подразумевает использование вызова `set`.

3. Как проверить присутствие/отсутствие элемента в множестве?

Для проверки присутствия используется `in`, для проверки отсутствия – `not in`.

4. Как выполнить перебор элементов множества?

С помощью цикла `for`.

5. Что такое `set comprehension`?

Это метод для создания множеств из других итерируемых объектов.

6. Как выполнить добавление элемента во множество?

Чтобы внести новые значения, потребуется вызывать метод `add`.

7. Как выполнить удаление одного или всех элементов множества?

Для удаления элементов из множества используются следующие функции в Python:

`remove` — удаление элемента с генерацией исключения в случае, если такого элемента нет;

`discard` — удаление элемента без генерации исключения, если элемент отсутствует;

`pop` — удаление первого элемента, генерируется исключение при попытке удаления из пустого множества.

Иногда необходимо полностью убрать все элементы. Чтобы не удалять каждый элемент отдельно, используется метод `clear`, не принимающий аргументов.

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

Чтобы объединить все элементы двух разных множеств, стоит воспользоваться методом `union` на одном из объектов.

Чтобы найти общие элементы для двух разных множеств, следует применить функцию `intersection`, принимающую в качестве аргумента один из наборов данных.

Чтобы вычислить разность для двух разных множеств, необходимо воспользоваться методом `difference`.

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

Чтобы выяснить, является ли множество `a` подмножеством `b`, стоит попробовать вывести на экран результат выполнения метода `issubset`.

Чтобы узнать, является ли множество `a` надмножеством `b`, необходимо вызвать метод `issuperset` и вывести результат его работы на экран.

10. Каково назначение множеств `frozenset` ?

Множество, содержимое которого не поддается изменению имеет тип `frozenset`. Значения из этого набора нельзя удалить, как и добавить новые.

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

Для преобразования множества в строку используется конкатенация текстовых значений, которую обеспечивает функция `join`. В этом случае ее аргументом является набор данных в виде нескольких строк.

Чтобы получить из множества словарь, следует передать функции `dict` набор из нескольких пар значений, в каждом из которых будет находиться ключ.

По аналогии с предыдущими преобразованиями можно получить список неких объектов. На этот раз используется вызов `list`, получающий в качестве аргумента множество `a`.

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