

**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 №2.7**

Дисциплина: «Программирование на Python»

Тема: «Работа с множествами в языке Python»

Выполнил: студент 1 курса

группы ИВТ-б-о-21-1

Кочкаров Умар Ахматович

Ставрополь 2022

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

**Ход работы:**

1) Создал репозиторий с использованием лицензии MIT и клонировал его

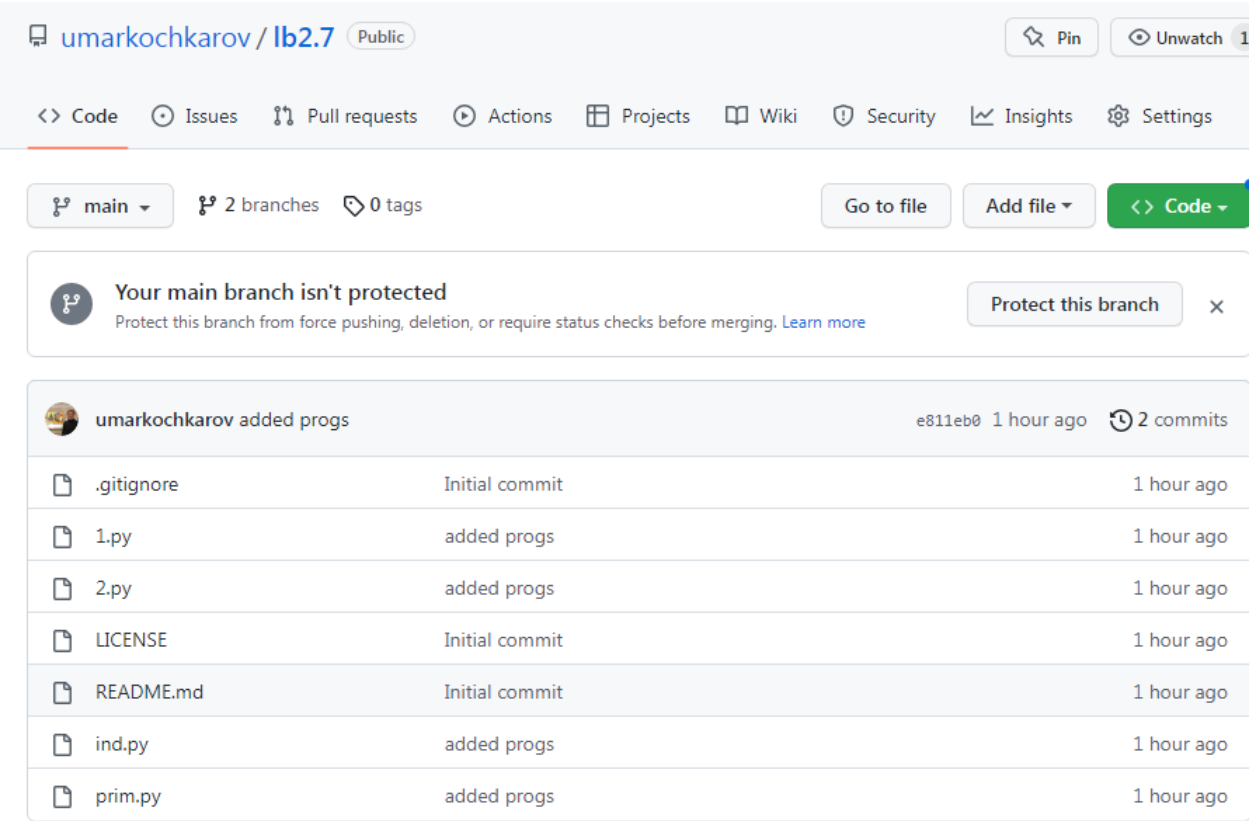


Рисунок 1. Репозиторий GitHub

2) Проработал примеры из лабораторной работы:

Пример 1. Определить результат выполнения операций над множествами. Считать элементы множества строками.

```
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
13     x = (a.intersection(b)).union(c)
14     print(f"x = {x}")
15
16     # Найдем дополнения множеств
17     bn = u.difference(b)
18     cn = u.difference(c)
19     y = (a.difference(d)).union(cn.difference(bn))
20
21     print(f"y = {y}")
```

PC Run - 2.7

Run: main ×

▶ ↑ C:\Users\erken\Desktop\Универ\python\Примеры\2.7\venv\  
x = {'e', 'd', 'k', 'o', 'j'}  
y = {'f', 'v', 'y', 'g', 'h', 'c', 'o'}

Рисунок 2. Выполнение программы

**Задача 1.** Подсчитайте количество гласных в строке, введенной с клавиатуры с использованием множеств.



```
1  #!/usr/bin/env python3
2  #- coding: utf-8 -*-
3
4  if __name__ == "__main__":
5      a = set(input("Введите: "))
6      b = {'b', 'c', 'd', 'f', 'g', 'h', 'j', 'k', 'l', 'm', 'n', 'p', 'q', 'r', 's', 't', 'v', 'w', 'x', 'z', 'y'}
7
8      c = a.difference(b)
9      print(c)
10     print(len(c))
```

Run - 2.7

Run: 1 x

C:\Users\erken\Desktop\Универ\python\Примеры\2.7\venv\Scripts\python.exe C:\Users\erken\Desktop\Универ\python\Пр

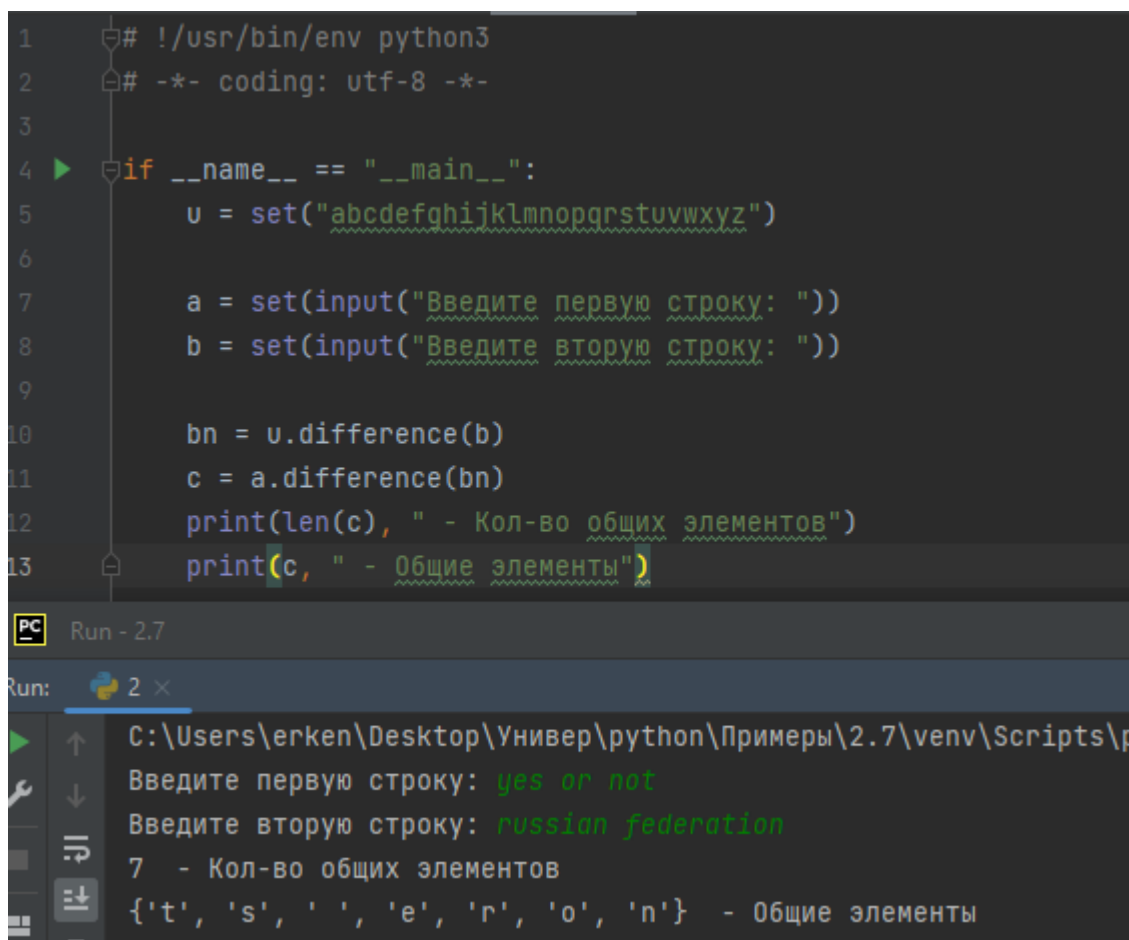
Введите: a

{'a'}

1

Рисунок 3. Выполнение программы

**Задача 2.** Определите общие символы в двух строках, введенных с клавиатуры.



```
1  #!/usr/bin/env python3
2  #- coding: utf-8 -*-
3
4  if __name__ == "__main__":
5      u = set("abcdefghijklmnopqrstuvwxyz")
6
7      a = set(input("Введите первую строку: "))
8      b = set(input("Введите вторую строку: "))
9
10     bn = u.difference(b)
11     c = a.difference(bn)
12     print(len(c), " - Кол-во общих элементов")
13     print(c, " - Общие элементы")
```

Run - 2.7

Run: 2 x

C:\Users\erken\Desktop\Универ\python\Примеры\2.7\venv\Scripts\p

Введите первую строку: yes or not

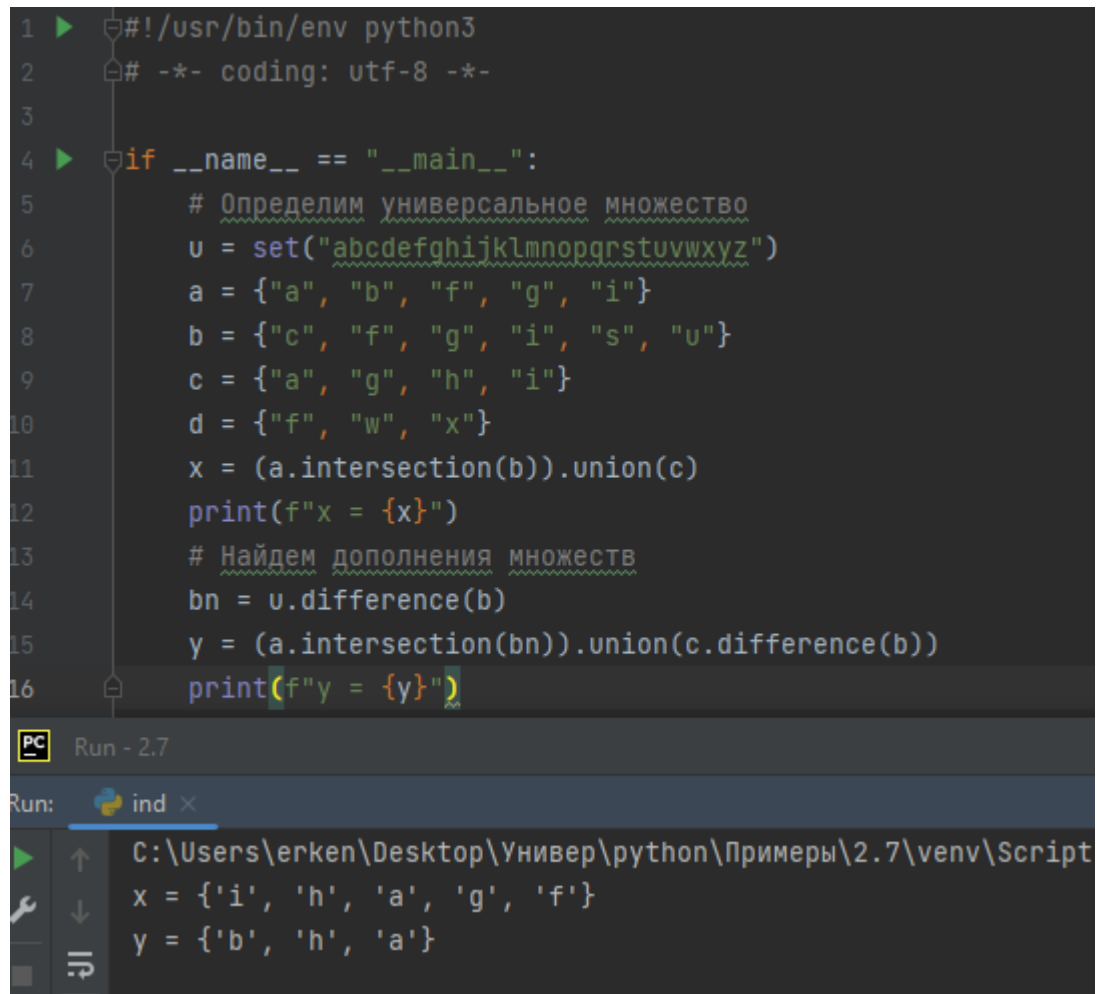
Введите вторую строку: russian federation

7 - Кол-во общих элементов

{'t', 's', ' ', 'e', 'r', 'o', 'n'} - Общие элементы

Рисунок 4. Выполнение программы

## Индивидуальное задание



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == "__main__":
5     # Определим универсальное множество
6     u = set("abcdefghijklmnopqrstuvwxyz")
7     a = {"a", "b", "f", "g", "i"}
8     b = {"c", "f", "g", "i", "s", "u"}
9     c = {"a", "g", "h", "i"}
10    d = {"f", "w", "x"}
11    x = (a.intersection(b)).union(c)
12    print(f"x = {x}")
13    # Найдем дополнения множеств
14    bn = u.difference(b)
15    y = (a.intersection(bn)).union(c.difference(b))
16    print(f"y = {y}")
```

Run - 2.7

Run: ind x

C:\Users\erken\Desktop\Универ\python\Примеры\2.7\venv\Script

x = {'i', 'h', 'a', 'g', 'f'}

y = {'b', 'h', 'a'}

Рисунок 4. Выполнение программы

### 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.

#### 2. Как осуществляется создание множеств в Python?

Сделать это можно, просто присвоив переменной последовательность значений, выделив их фигурными скобками.

#### 3. Как проверить присутствие/отсутствие элемента в множестве?

Проверка, есть ли данное значение в множестве. Для этого используется `in`. Наоборот, проверка отсутствия. Используется `not in`.

#### **4. Как выполнить перебор элементов множества?**

Перебрать все элементы множества (в неопределенном порядке) можно при помощи цикла `for`.

#### **5. Что такое `set comprehension`?**

`Set comprehension` – генератор, позволяющий заполнять списки, а также другие наборы данных с учетом неких условий.

#### **6. Как выполнить добавление элемента во множество?**

Чтобы внести новые значения, потребуется вызывать метод `add`. Аргументом в данном случае будет добавляемый элемент последовательности.

#### **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. Чтобы не удалять каждый элемент отдельно, используется метод `clear`, не принимающий аргументов. Если вывести содержимое после этой операции, на экране появится только его название.

#### **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 (пересечение), следует применить

функцию `intersection` , принимающую в качестве аргумента один из наборов данных. Чтобы вычислить разность для двух разных множеств, необходимо воспользоваться методом `difference`. Функция позволяет найти элементы, уникальные для второго набора данных, которых в нем нет.

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

Для определения подмножеств и надмножеств существуют специальные функции, возвращающие `True` или `False` в зависимости от результата выполнения.

Чтобы выяснить, является ли множество `a` подмножеством `b`, стоит попробовать вывести на экран результат выполнения метода `issubset`, `print(a.issubset(b))`. Чтобы узнать, является ли множество `a` надмножеством `b`, необходимо вызвать метод `issuperset` и вывести результат его работы на экран, `print(a.issuperset(b))`.

## **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. Можно получить список неких объектов. На этот раз используется вызов `list` , получающий в качестве аргумента множество `a`. Чтобы получить из множества словарь, следует передать функции `dict` набор из нескольких пар значений, в каждом из которых будет находиться ключ.

**Вывод:** в ходе выполнения лабораторной работы приобрели навыки по работе с множествами при написании программ с помощью языка программирования Python версии 3.x