

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ  
ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных технологий имени  
профессора Н.И. Червякова

Кафедра инфокоммуникаций

**ОТЧЕТ**

**по лабораторной работе №2**

Дисциплина: «Языки программирования»

Выполнил студент группы

ИТС-б-о-20-1 (1)

Абдуллаев Р.Р. « » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

Работа защищена «

» \_\_\_\_\_ 20\_\_ г.

Проверил

к.т.н., доцент

кафедры инфокоммуникаций

доцент

Воронкин Р.А.

---

(подпись)

Ставрополь, 2021 г.

## ЛАБОРАТОРНАЯ РАБОТА №2

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

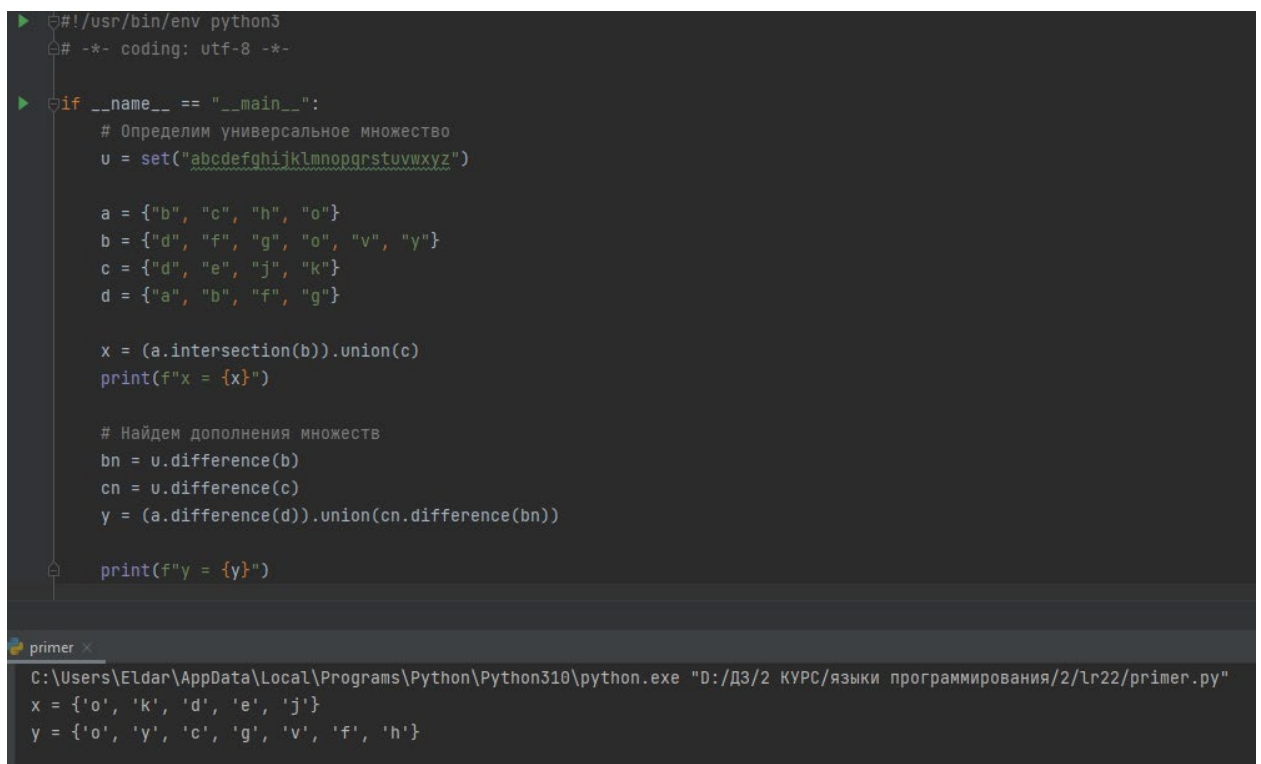
**Ссылка на репозиторий:**

<https://github.com/SindiCATT/lr22.git>

**Порядок выполнения работы:**

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

$$A = \{b, c, h, o\}; \quad B = \{d, f, g, o, v, y\}; \quad C = \{d, e, j, k\}; \quad D = \{a, b, f, g\}; \quad X = (A \cap B) \cup C; \quad Y = (A/D) \cup (C/B). \quad (1)$$



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

if __name__ == "__main__":
    # Определим универсальное множество
    u = set("abcdefghijklmnopqrstuvwxyz")

    a = {"b", "c", "h", "o"}
    b = {"d", "f", "g", "o", "v", "y"}
    c = {"d", "e", "j", "k"}
    d = {"a", "b", "f", "g"}

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

    # Найдем дополнения множеств
    bn = u.difference(b)
    cn = u.difference(c)
    y = (a.difference(d)).union(cn.difference(bn))

    print(f"y = {y}")
```

primer x

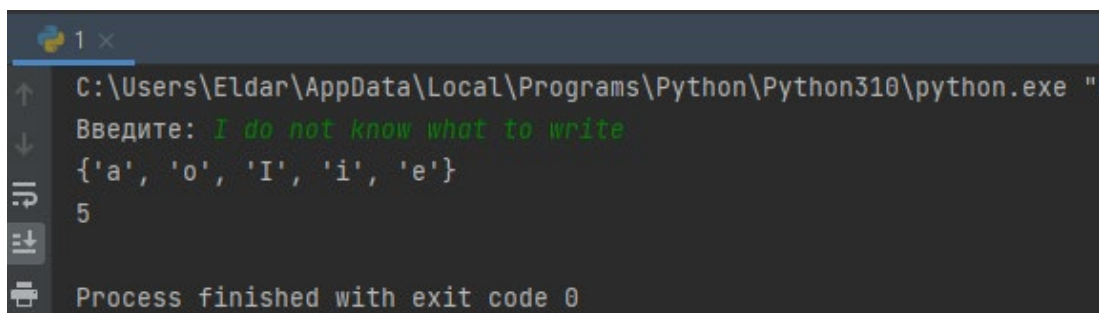
C:\Users\Eldar\AppData\Local\Programs\Python\Python310\python.exe "D:/ДЗ/2 КУРС/языки программирования/2/lr22/primer.py"

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

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

Рисунок 1 – Результат выполнения

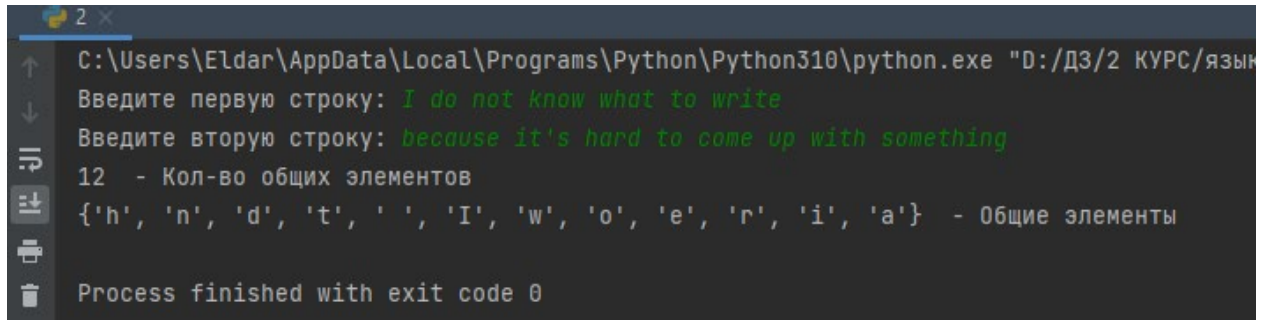
**Задача 1.** Подсчитайте количество гласных в строке, введенной с клавиатуры с использованием множеств.



```
1 x
C:\Users\Eldar\AppData\Local\Programs\Python\Python310\python.exe "
Введите: I do not know what to write
{'a', 'o', 'I', 'i', 'e'}
5
Process finished with exit code 0
```

Рисунок 2 – Результат выполнения программы

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

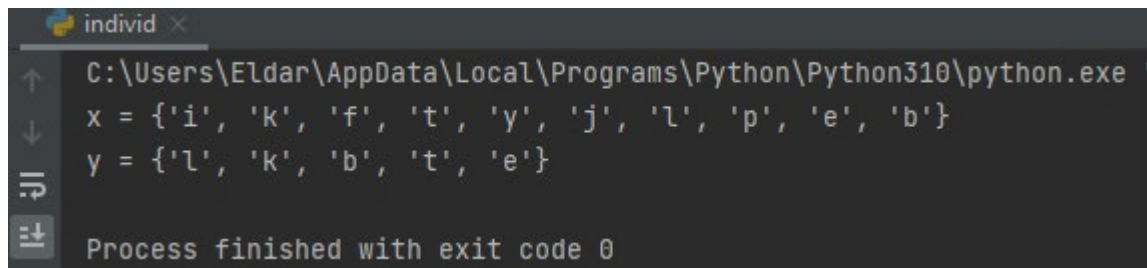


```
C:\Users\Eldar\AppData\Local\Programs\Python\Python310\python.exe "D:/ДЗ/2 КУРС/язык
Введите первую строку: I do not know what to write
Введите вторую строку: because it's hard to come up with something
12 - Кол-во общих элементов
{'h', 'n', 'd', 't', ' ', 'I', 'w', 'o', 'e', 'r', 'i', 'a'} - Общие элементы
Process finished with exit code 0
```

Рисунок 3 – Результат выполнения программы

### Индивидуальное задание

$$A = \{b, e, f, k, t\}; \quad B = \{f, i, j, p, y\}; \quad C = \{j, k, l, y\}; \quad D = \{i, j, s, t, u, y, z\};$$
$$X = (A \cap C) \cup (B \cap C); \quad Y = (A \cap \bar{B}) \cup (D/C).$$



```
individ x
C:\Users\Eldar\AppData\Local\Programs\Python\Python310\python.exe
x = {'i', 'k', 'f', 't', 'y', 'j', 'l', 'p', 'e', 'b'}
y = {'l', 'k', 'b', 't', 'e'}
Process finished with exit code 0
```

Рисунок 4 – Результат выполнения 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.