

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

Институт цифрового развития  
Кафедра инфокоммуникаций

**«Работа с множествами в языке Python»**

**ОТЧЕТ**  
**по лабораторной работе №10**  
**дисциплины**  
**«Основы программной инженерии»**

Выполнил:

Гълбачева Доротея Андреева  
2 курс, группа ПИЖ-б-о-21-1,  
09.03.04 «Программная инженерия»,  
направленность (профиль) «Разработка  
и сопровождение программного  
обеспечения», очная форма обучения

---

(подпись)

Проверил:

---

(подпись)

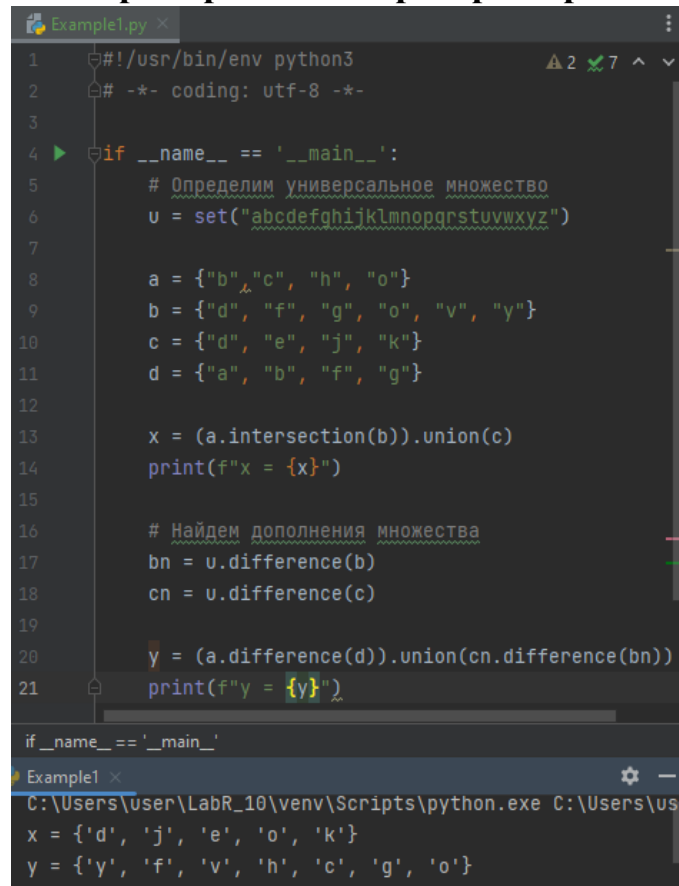
Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2022 г.

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

### Выполнения лабораторной работы:

#### 1. Проработка примеров из лабораторной работы:



```
Example1.py
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 множества
17     bn = u.difference(b)
18     cn = u.difference(c)
19
20     y = (a.difference(d)).union(cn.difference(bn))
21     print(f"y = {y}")

if __name__ == '__main__':
Example1
C:\Users\User\LabR_10\venv\Scripts\python.exe C:\Users\User\LabR_10\venv\Scripts\python.exe
x = {'d', 'j', 'e', 'o', 'k'}
y = {'y', 'f', 'v', 'h', 'c', 'g', 'o'}
```

Рисунок 9.3 - Код и результат работы программы примера №1

**Задание №1:** Решите задачу: подсчитайте количество гласных в строке, введенной с клавиатуры с использованием множеств.

```
Example1.py x Individual1.py x Example2.py x
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == '__main__':
5      a = set("аоуэыяеюи")
6
7      s = input("Введите строку: "). lower()
8
9      count = 0
10     for i in s:
11         if i in a:
12             count += 1
13
14     print(f"Количество гласных: {count}")
15
16 if __name__ == '__main__'
```

Example2 x

C:\Users\user\LabR\_10\venv\Scripts\python.exe C:\U  
Введите строку: Сегодня погода хорошая  
Количество гласных: 10

Рисунок 9.3 - Код и результат работы программы задание №1

Задание №2: Решите задачу: определите общие символы в двух строках, введенных с клавиатуры.

```
Example1.py x Individual1.py x Example2.py x Example3.py x
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == '__main__':
5      a = set(".,;:' !?@#()")
6
7      l1 = set(input("Введите первую строку: ").lower())
8      l2 = set(input("Введите вторую строку: ").lower())
9
10     p = l1.intersection(l2)
11     pn = p.difference(a)
12
13     print(f"Общие символы: {pn}")
14
15 if __name__ == '__main__'
```

Example3 x

C:\Users\user\LabR\_10\venv\Scripts\python.exe C:\Users\user\L  
Введите первую строку: Привет, сегодня погода хорошая  
Введите вторую строку: Собака спит.  
Общие символы: {'с', 'а', 'и', 'т', 'п', 'о'}

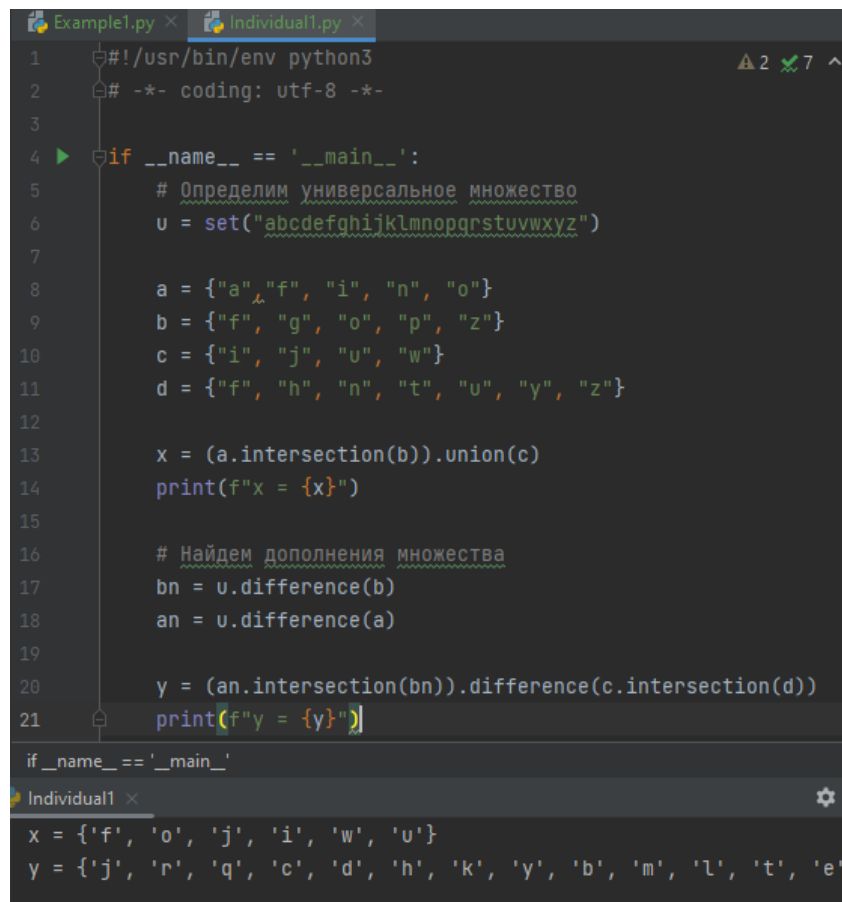
Рисунок 9.3 - Код и результат работы программы задание №2

## 2. Индивидуальные задания:

### Задание №1:

$$A = \{a, f, I, n, o\}; \quad B = \{f, g, o, p, z\}; \quad C = \{i, j, u, w\}; \quad D = \{f, h, n, t, u, y, z\};$$
$$X = (A \cap B) \cup C; \quad Y = (\bar{A} \cap \bar{B}) / (C \cup D).$$

8.



```
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 = {"a", "f", "i", "n", "o"}
9      b = {"f", "g", "o", "p", "z"}
10     c = {"i", "j", "u", "w"}
11     d = {"f", "h", "n", "t", "u", "y", "z"}
12
13     x = (a.intersection(b)).union(c)
14     print(f"x = {x}")
15
16     # Найдем дополнения множества
17     bn = u.difference(b)
18     an = u.difference(a)
19
20     y = (an.intersection(bn)).difference(c.intersection(d))
21     print(f"y = {y}")
```

if \_\_name\_\_ == '\_\_main\_\_':

Individual1

```
x = {'f', 'o', 'j', 'i', 'w', 'u'}
y = {'j', 'r', 'q', 'c', 'd', 'h', 'k', 'y', 'b', 'm', 'l', 't', 'e'}
```

Рисунок 9.3 - Код и результат работы программы задания №1

## 3. Контрольные вопросы:

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

Множеством в Python называется неупорядоченная совокупность уникальных значений. В качестве элементов этого набор данных могут выступать любые неизменяемые объекты, такие как числа, символы, строки.

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

```
a = {1, 2, 0, 1, 3, 2}
print(a)

{0, 1, 2, 3}
```

```
a = set('data')
print(a)

{'d', 'a', 't'}
```

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

- Проверка, есть ли данное значение в множестве. Для этого используется `in`.

```
a = {0, 1, 2, 3}
print(2 in a)

True
```

- Наоборот, проверка отсутствия. Используется `not in`.

```
a = {0, 1, 2, 3}
print(2 not in a)

False
```

### 4. Как выполнить перебор элементов множества?

- Перебор всех элементов.

```
for a in {0, 1, 2}:
    print(a)

0
1
2
```

### 5. Что такое set comprehensions?

Для создания множества можно в Python воспользоваться генератором, позволяющих заполнять списки, а также другие наборы данных с учетом неких условий.

```
a = {i for i in [1, 2, 0, 1, 3, 2]}
print(a)

{0, 1, 2, 3}
```

## 6. Как выполнить добавление элемента во множество?

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

```
a = {0, 1, 2, 3}
a.add(4)
print(a)

{0, 1, 2, 3, 4}
```

## 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.

## 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`.

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?

Строка. 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. Запятая в кавычках выступает в качестве символа, разделяющего значения. Метод `type` возвращает тип данных объекта в конце приведенного кода.

```
a = {'set', 'str', 'dict', 'list'}
b = ','.join(a)
print(b)
print(type(b))

set,dict,list,str
<class 'str'>
```

Словарь. Чтобы получить из множества словарь, следует передать функции `dict` набор из нескольких пар значений, в каждом из

которых будет находиться ключ. Функция `print` демонстрирует на экране содержимое полученного объекта, а `type` отображает его тип.

```
a = {('a', 2), ('b', 4)}  
b = dict(a)  
print(b)  
print(type(b))  
  
{'b': 4, 'a': 2}  
<class 'dict'>
```

Список. На этот раз используется вызов `list`, получающий в качестве аргумента множество `a`. На выходе функции `print` отображаются уникальные значения для изначального набора чисел

```
a = {1, 2, 0, 1, 3, 2}  
b = list(a)  
print(b)  
print(type(b))  
  
[0, 1, 2, 3]  
<class 'list'>
```