

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:

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

(подпись)

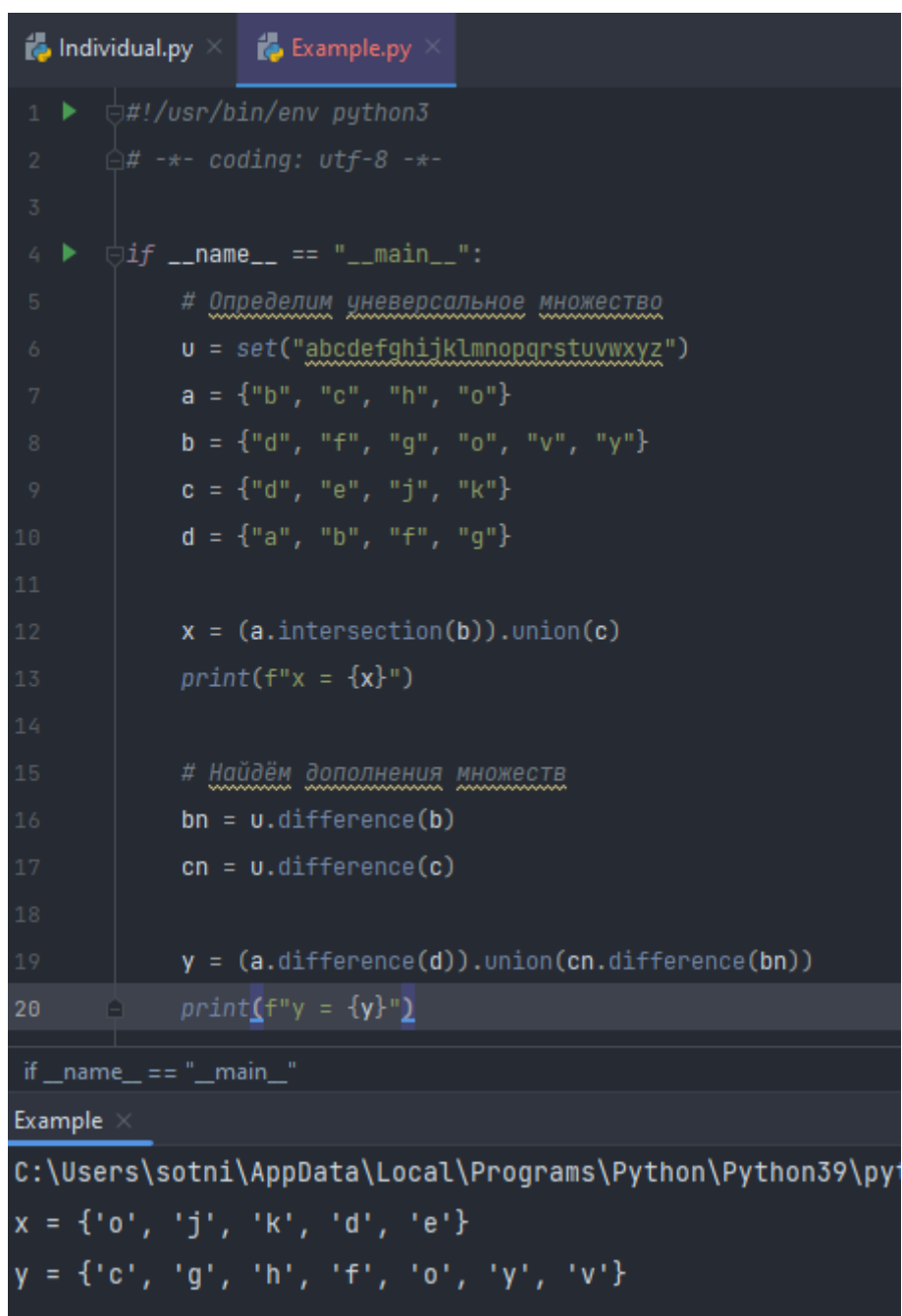
Проверил:

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

Проработка примера из лабораторной работы:



The image shows a screenshot of a Python IDE with two tabs: 'Individual.py' and 'Example.py'. The 'Example.py' tab is active, displaying a Python script. The script defines a universal set 'u' and four subsets 'a', 'b', 'c', and 'd'. It then calculates the intersection of 'a' and 'b' unioned with 'c' to get 'x', and the difference of 'u' from 'b' and 'c' unioned with the difference of 'a' from 'd' to get 'y'. The script prints the results of 'x' and 'y'.

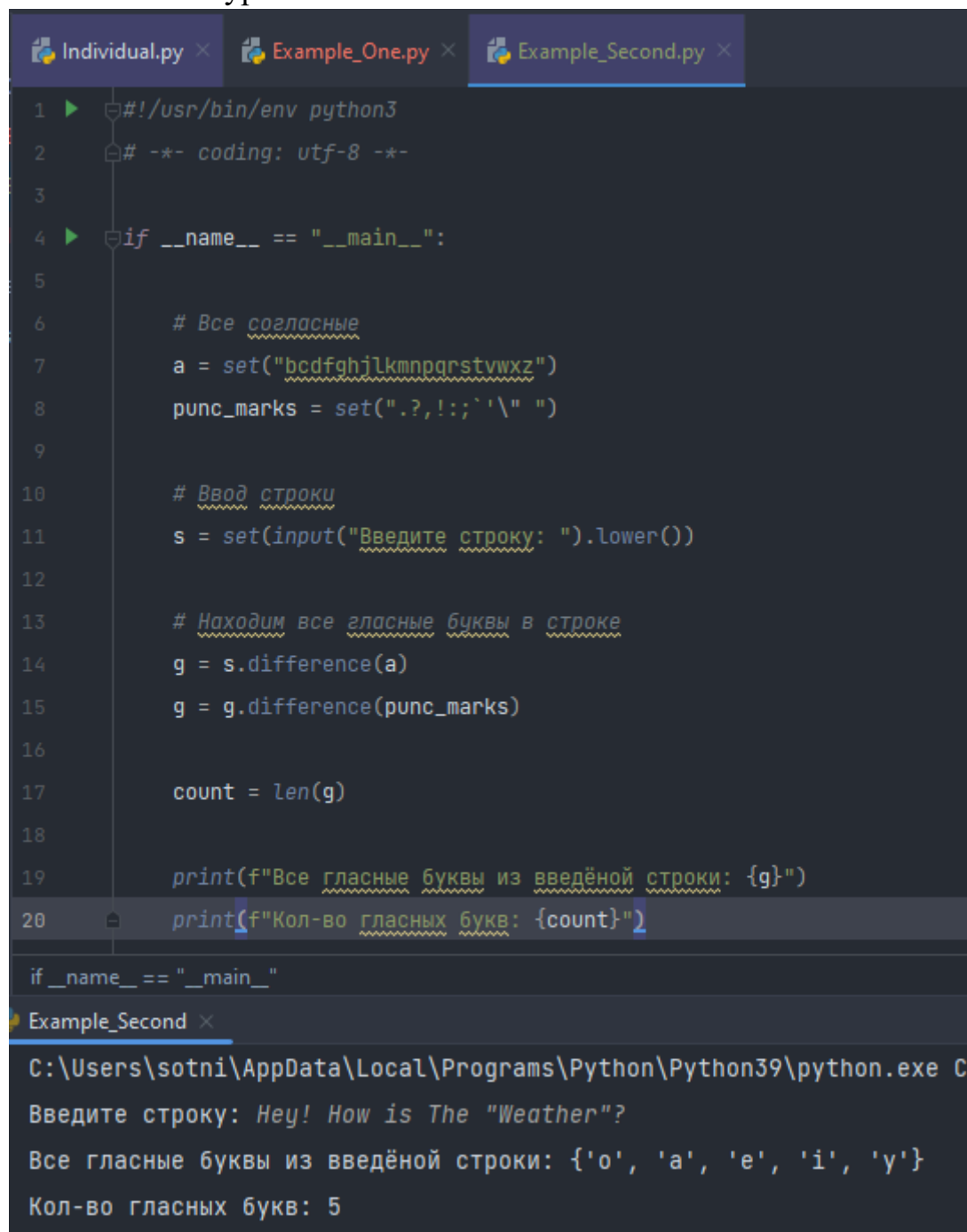
```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == "__main__":
5     # Определим универсальное множество
6     u = set("abcdefghijklmnopqrstuvwxyz")
7     a = {"b", "c", "h", "o"}
8     b = {"d", "f", "g", "o", "v", "y"}
9     c = {"d", "e", "j", "k"}
10    d = {"a", "b", "f", "g"}
11
12    x = (a.intersection(b)).union(c)
13    print(f"x = {x}")
14
15    # Найдём дополнения множеств
16    bn = u.difference(b)
17    cn = u.difference(c)
18
19    y = (a.difference(d)).union(cn.difference(bn))
20    print(f"y = {y}")
```

Below the code editor, the output of the program is displayed:

```
Example x
C:\Users\sotni\AppData\Local\Programs\Python\Python39\pyt
x = {'o', 'j', 'k', 'd', 'e'}
y = {'c', 'g', 'h', 'f', 'o', 'y', 'v'}
```

Рисунок 1 – код и результат работы программы из примера

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

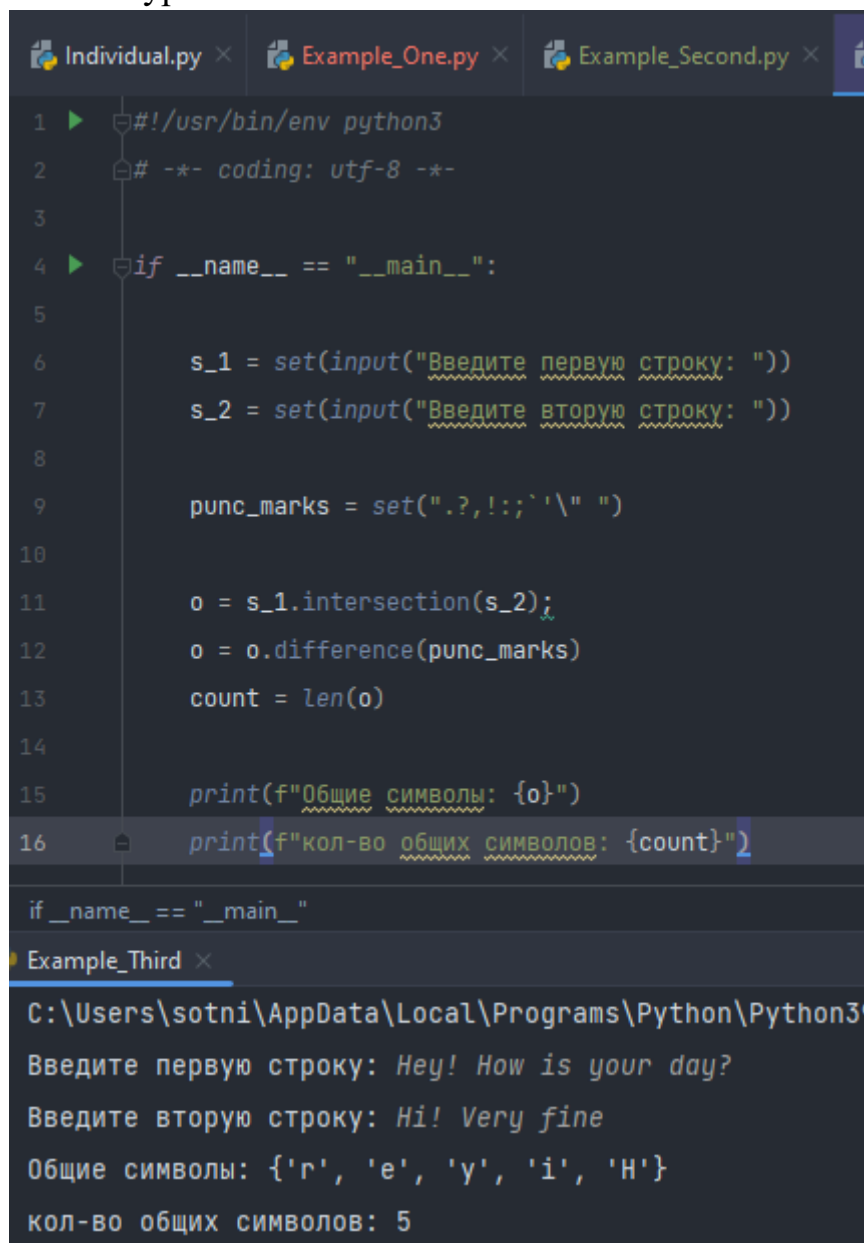


```
Individual.py × Example_One.py × Example_Second.py ×
1  ▶  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  ▶  if __name__ == "__main__":
5
6      # Все согласные
7      a = set("bcdfghjklmnpqrstvwxyz")
8      punc_marks = set(".?,:!;'\" ")
9
10     # Ввод строки
11     s = set(input("Введите строку: ").lower())
12
13     # Находим все гласные буквы в строке
14     g = s.difference(a)
15     g = g.difference(punc_marks)
16
17     count = len(g)
18
19     print(f"Все гласные буквы из введённой строки: {g}")
20     print(f"Кол-во гласных букв: {count}")

if __name__ == "__main__"
Example_Second ×
C:\Users\sotni\AppData\Local\Programs\Python\Python39\python.exe C
Введите строку: Hey! How is The "Weather"?
Все гласные буквы из введённой строки: {'o', 'a', 'e', 'i', 'y'}
Кол-во гласных букв: 5
```

Рисунок 2 – код и результат работы программы для задания №1

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



```
Individual.py × Example_One.py × Example_Second.py ×  
1 ▶ #!/usr/bin/env python3  
2 # -*- coding: utf-8 -*-  
3  
4 ▶ if __name__ == "__main__":  
5  
6     s_1 = set(input("Введите первую строку: "))  
7     s_2 = set(input("Введите вторую строку: "))  
8  
9     punc_marks = set(".,!,:;`'\ " )  
10  
11     o = s_1.intersection(s_2);  
12     o = o.difference(punc_marks)  
13     count = len(o)  
14  
15     print(f"Общие символы: {o}")  
16     print(f"кол-во общих символов: {count}")  
  
if __name__ == "__main__"  
Example_Third ×  
C:\Users\sotni\AppData\Local\Programs\Python\Python3  
Введите первую строку: Hey! How is your day?  
Введите вторую строку: Hi! Very fine  
Общие символы: {'r', 'e', 'y', 'i', 'H'}  
кол-во общих символов: 5
```

Рисунок 3 – код и результат работы программы для задания №2

Индивидуальное задание:

$$A = \{c, g, h, k, y\}; \quad B = \{a, b, k, n, u\}; \quad C = \{i, j, o, y, z\}; \quad D = \{a, b, f, g, y, z\};$$

$$X = (A \cup B) \cap D; \quad Y = (\bar{A} \cap D) \cup (\bar{C} / \bar{B}).$$

$$A = \{b, e, g, h, k, s\}; \quad B = \{c, g, p, q\}; \quad C = \{f, g, s, x, y, z\}; \quad D = \{a, c, d, g, u, v, z\};$$

$$X = (A \cup B) \cap C; \quad Y = (\bar{A} \cap D) \cup (C / B).$$

Решение:

$$A \cup B = \{c, g, n, b, u, a, k, y, h\}$$

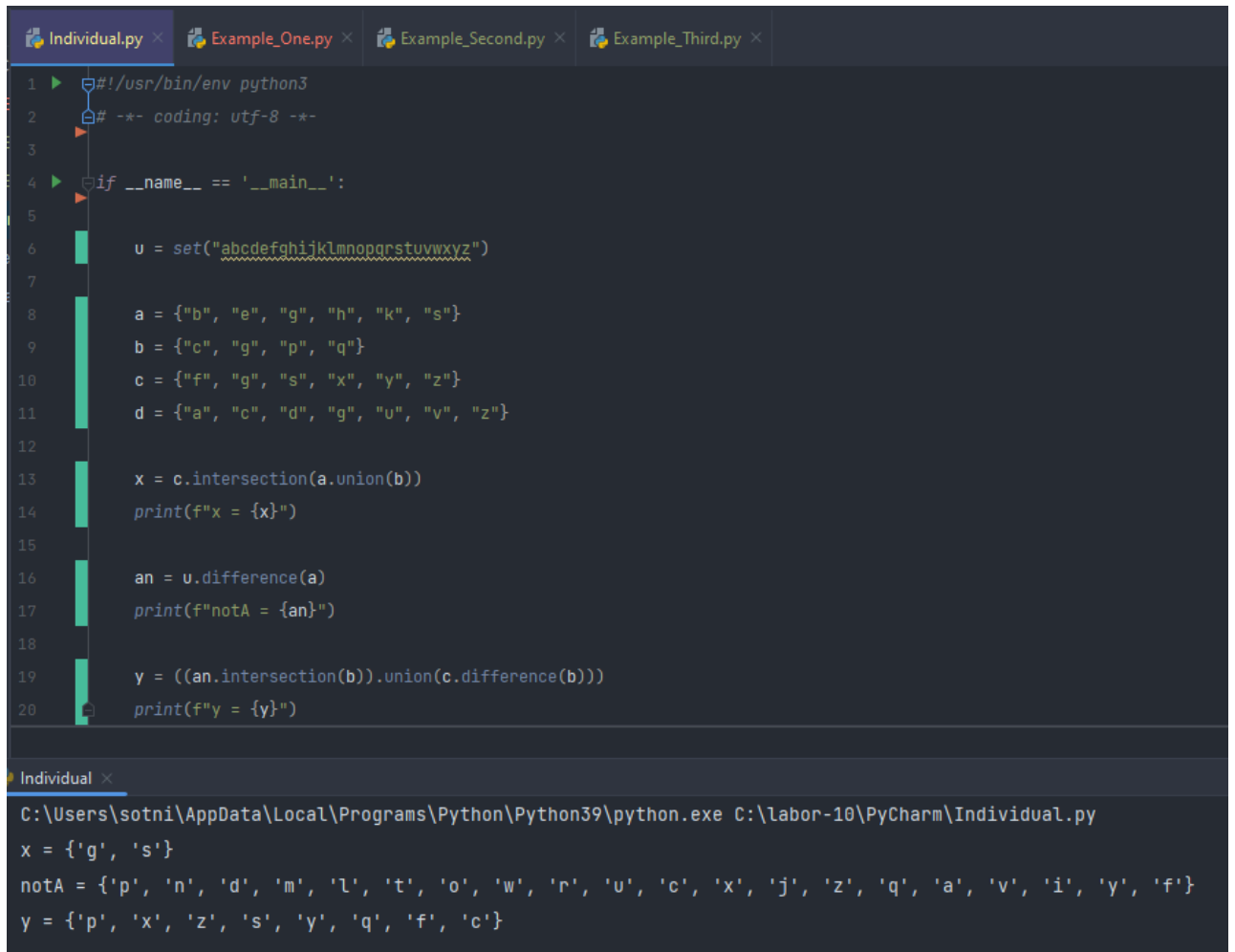
$$X = (A \cup B) \cap C = \{g, s\}$$

$$\bar{A} = \{l, u, I, d, c, r, o, q, v, a, m, x, y, f, p, z, n, t, w, j\}$$

$$\bar{A} \cap D = \{p, c, q\}$$

$$C / B = \{y, x, s, f, z\}$$

$$Y = (\bar{A} \cap D) \cup (C / B) = \{y, c, x, s, z, f, p, q\}$$



The image shows a PyCharm IDE with four tabs: Individual.py, Example_One.py, Example_Second.py, and Example_Third.py. The 'Individual.py' tab is active, displaying a Python script. The script defines a universal set 'u' as the set of all lowercase letters. It then defines four sets: 'a' (b, e, g, h, k, s), 'b' (c, g, p, q), 'c' (f, g, s, x, y, z), and 'd' (a, c, d, g, u, v, z). The script calculates the intersection of 'c' and the union of 'a' and 'b', printing the result 'x'. It also calculates the difference of 'u' and 'a', printing 'notA'. Finally, it calculates the union of the intersection of 'notA' and 'b' and the difference of 'c' and 'b', printing the result 'y'.

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == '__main__':
5
6     u = set("abcdefghijklmnopqrstuvwxyz")
7
8     a = {"b", "e", "g", "h", "k", "s"}
9     b = {"c", "g", "p", "q"}
10    c = {"f", "g", "s", "x", "y", "z"}
11    d = {"a", "c", "d", "g", "u", "v", "z"}
12
13    x = c.intersection(a.union(b))
14    print(f"x = {x}")
15
16    an = u.difference(a)
17    print(f"notA = {an}")
18
19    y = ((an.intersection(b)).union(c.difference(b)))
20    print(f"y = {y}")
```

The output window shows the execution of the script, displaying the following results:

```
C:\Users\sotni\AppData\Local\Programs\Python\Python39\python.exe C:\labor-10\PyCharm\Individual.py
x = {'g', 's'}
notA = {'p', 'n', 'd', 'm', 'l', 't', 'o', 'w', 'r', 'u', 'c', 'x', 'j', 'z', 'q', 'a', 'v', 'i', 'y', 'f'}
y = {'p', 'x', 'z', 's', 'y', 'q', 'f', 'c'}
```

Рисунок 4 – код и результат работы программы для индивидуального задания

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?

```
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 comprehension?

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`. Аргументом в данном случае будет добавляемый элемент последовательности. В примере кода на Python добавим в множество элемент со значением 4.

```
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 (кроме очистки, которая будет рассмотрена ниже):

`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.

Избавиться от лишних значений в наборе данных с помощью remove. В качестве входного параметра здесь выступает элемент, который нужно удалить (в примере удалим число со значением 3).

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 добавить все элементы из одного множества к другому, необходимо вызывать метод update на первом объекте.

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. В следующем примере демонстрируется создание при помощи стандартной функции.

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

Строка:

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

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

Словарь:

```
a = {('a', 2), ('b', 4)}
b = dict(a)
print(b)
print(type(b))

{'b': 4, 'a': 2}
<class 'dict'>
```

Список:

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

[0, 1, 2, 3]
<class 'list'>
```