

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Лабораторная работа 9

Работа с множествами в языке Python

Выполнил студент группы ИВТ-б-о-20-1

Симанский М.Ю « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____

(подпись)

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

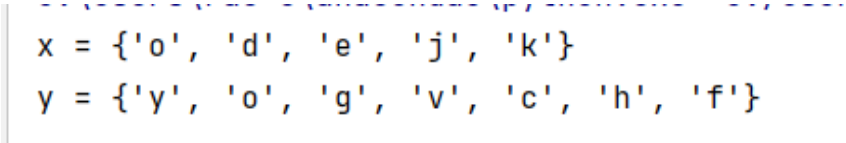
Пример 1. Определить результат выполнения операций над множествами. Считать элементы множества строками.

Примечание: в качестве универсального множества считать все строчные латинские буквы от a до z.

Код

```
#!/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}")
```

Результат выполнения



```
x = {'o', 'd', 'e', 'j', 'k'}
y = {'y', 'o', 'g', 'v', 'c', 'h', 'f'}
```

Рисунок 1 – Результат

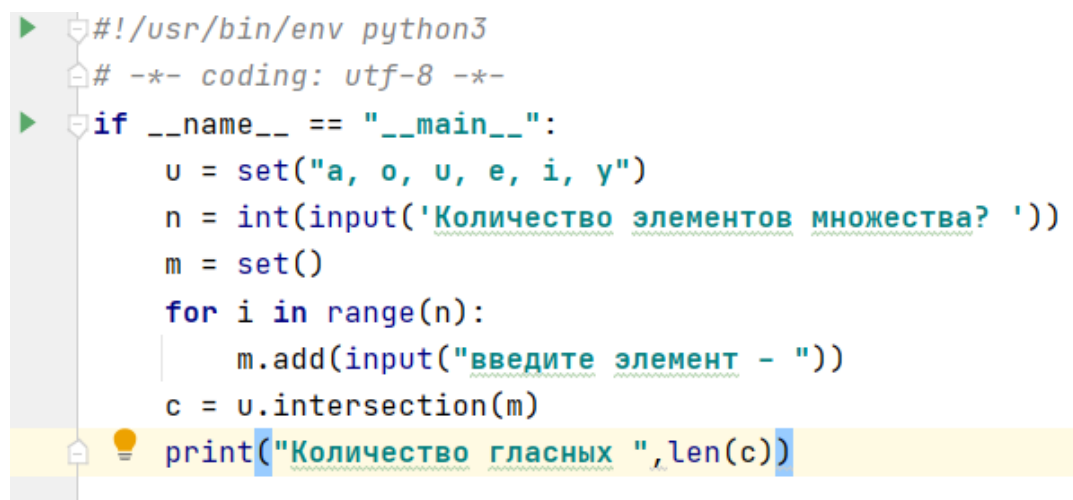
Задание 1

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

Код

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ == "__main__":
    u = set("a, o, u, e, i, y")
    n = int(input('Количество элементов множества? '))
    m = set()
    for i in range(n):
        m.add(input("введите элемент - "))
    c = u.intersection(m)
    print("Количество гласных ", len(c))
```

Результат



```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ == "__main__":
    u = set("a, o, u, e, i, y")
    n = int(input('Количество элементов множества? '))
    m = set()
    for i in range(n):
        m.add(input("введите элемент - "))
    c = u.intersection(m)
    print("Количество гласных ", len(c))
```

Рисунок 2 – Результат выполнения программы

Задание 2

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

Код

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ == "__main__":
    m = (input("введите элемент - "))
    k = (input("введите элемент 2- "))
    m = set(m)
    k = set(k)
```

```
c = k.intersection(m)
print("Общие элементы", c)
```

Результат

```

введите элемент - цйууцкеукнзш
введите элемент 2- йцку
Общие элементы {'й', 'ц', 'к', 'у'}
```

Рисунок 3 – Результат исполнения программы

Индивидуальное задание. Вариант 18.

$$A = \{b, c, g, I, w\}; \quad B = \{e, g, h, q, w\}; \quad C = \{c, d, k, l, y\}; \quad D = \{a, g, h, u, v, z\};$$

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

Код

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ == "__main__":
    # Определим универсальное множество
    u = set("abcdefghijklmnopqrstuvwxyz")
    a = {"b", "c", "g", "I", "w"}
    b = {"e", "g", "h", "q", "w"}
    c = {"c", "d", "k", "l", "y"}
    d = {"a", "g", "h", "u", "v", "z"}
    x = (a.intersection(c)).union(b)
    print(f"x = {x}")
    # Найдем дополнения множеств
    an = u.difference(a)
    y = (a.difference(d)).union(c.difference(b))
    print(f"y = {y}")
```

Результат

```
x = {'q', 'e', 'h', 'c', 'g', 'w'}  
y = {'l', 'k', 'd', 'b', 'I', 'c', 'y', 'w'}
```

Рисунок 4 – Результат выполнения

Ответы на контрольные вопросы

1. Множеством в языке программирования Python называется неупорядоченная совокупность уникальных значений.
2. Перед тем как начать работу с множеством, необходимо для начала его создать. Сделать это можно, просто присвоив переменной последовательность значений, выделив их фигурными скобками.
3. Проверка, есть ли данное значение в множестве. Для этого используется `in`. Наоборот, проверка отсутствия. Используется `not in`.
4. Перебор всех элементов - `for a in {0, 1, 2}: print(a)`
5. Для создания множества можно в Python воспользоваться генератором, позволяющих заполнять списки, а также другие наборы данных с учетом неких условий.
6. Чтобы внести новые значения, потребуется вызывать метод `add`. Аргументом в данном случае будет добавляемый элемент последовательности.
7. 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; `discard` — удаление элемента без генерации исключения, если элемент отсутствует; `pop` — удаление первого элемента, генерируется исключение при попытке удаления из пустого множества.
8. 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` .

9. 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. Поскольку все элементы набора чисел `b` присутствуют в `a`, функция возвращает `True`.

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

11. Для преобразования множества в строку используется конкатенация текстовых значений, которую обеспечивает функция `join`. 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. По аналогии с предыдущими преобразованиями можно получить список неких объектов. На этот раз используется вызов `list` , получающий в качестве аргумента множество `a` . На выходе функции `print` отображаются уникальные значения для изначального набора чисел.