

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

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ПРАКТИЧЕСКОЙ РАБОТЕ №2.7**  
**дисциплины «Программирование на Python»**  
Вариант №2

Выполнила:  
Беседина Инга Олеговна  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника», очная  
форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Р. А., канд. технических  
наук, доцент, доцент кафедры  
инфокоммуникаций

---

(подпись)

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

Ставрополь, 2023 г.

## Тема: Работа с множествами в языке Python

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

### Ход работы

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

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

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

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

Рисунок 1. Результат работы программы

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

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

if __name__ == "__main__":
    u = set("aeiouy")

    line = input("Строка: ")

    cnt = len([s for s in line if s in u])
    print("Гласных букв в строке:", cnt)
```

```
Строка: qwerty
Гласных букв в строке: 2
```

Рисунок 2. Результат работы программы

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

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

if __name__ == "__main__":
    a = set(input("Первая строка: "))
    b = set(input("Вторая строка: "))

    c = a.intersection(b)
    print("Общие символы:", c)
```

```
Первая строка: hqsljkor
Вторая строка: gxcolmnb
Общие символы: {'l', 'o', 'g'}
```

Рисунок 3. Результат работы программы

### Индивидуальное задание:

2.

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

$$A = \{a, h, m, o, r\}; \quad B = \{j, k, o, u, y\}; \quad C = \{g, h, j\}; \quad D = \{g, j, q\};$$

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

if __name__ == "__main__":
    u = set("abcdefghijklmnopqrstuvwxyz")

    a = {"a", "h", "m", "o", "r"}
    b = {"j", "k", "o", "u", "y"}
    c = {"g", "h", "j"}
    d = {"g", "j", "q"}

    bn = u.difference(b)

    x = (a.difference(c)).intersection(bn)
    print(f"x = {x}")

    y = (a.intersection(bn)).union((c.difference(d)))
    print(f"y = {y}")
```

```
x = {'r', 'm', 'a'}
y = {'h', 'm', 'a', 'r'}
```

Рисунок 4. Результат работы программы

### Контрольные вопросы:

1. Множество в языке Python - это неупорядоченная коллекция уникальных элементов.
2. Для создания множества в Python используется фигурные скобки или функция set().

3. Для проверки присутствия/отсутствия элемента в множестве можно использовать оператор `in` или `not in`.

4. Перебор элементов множества можно осуществить с помощью цикла `for`.

5. `Set comprehension` - это способ создания множества на основе итерации по другой последовательности с использованием синтаксиса, аналогичного списковому включению.

6. Для добавления элемента во множество используется метод `add()`.

7. Для удаления одного или всех элементов множества используются методы `remove()`, `discard()` и `clear()`.

8. 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`, 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. Чтобы выяснить, является ли множество `a` подмножеством `b`, стоит попробовать вывести на экран результат выполнения метода `issubset`, как в следующем примере. Так как не все элементы набора чисел `a` присутствуют в `b`, функция вернет `False`. 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. По аналогии с

предыдущими преобразованиями можно получить список неких объектов. На этот раз используется вызов `list`, получающий в качестве аргумента множество `a`.

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