

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное  
образовательное учреждение высшего образования**

**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**Отчет по лабораторной работе №2.9**

**Работа с множествами в языке Python.**

**по дисциплине «Технологии программирования и алгоритмизации»**

Выполнил студент группы ИВТ-б-о-20-1

Хашиев Х.М. « » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверил Воронкин Р.А. \_\_\_\_\_

(подпись)

Ставрополь 2021

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

### Ход работы: Пример 1

<https://github.com/Mirror-Shard/L2.7>

1. Создал репозиторий на github с лицензией MIT, добавил .gitignore и выбрал язык Python.

2. Проработал пример из учебника:

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

Рисунок 1 – Код примера

### 3. Результат выполнения:

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

Рисунок 2 – Результат работы примера

## Задание 1

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

1. Код задания 1 (было создано множество из английских и русских гласных)

```
#!/usr/bin/env python3  
# -*- coding: utf-8 -*-  
  
if __name__ == '__main__':  
  
    # Множество гласных  
    vocales = {'a', 'e', 'u', 'i', 'o', 'y', "y", "e", "ы", "а", "о", "э", "я", "и", "ю"}  
  
    # Количество гласных в строке  
    vocales_in_linea = 0  
  
    # Ввод строки  
    linea = input("Введите строку символов: ").lower()  
  
    # Проходит по всем гласным  
    for vocal in vocales:  
        # И проверяет их наличие в строке  
        if vocal in linea:  
            vocales_in_linea += 1  
  
    print(f"Гласных в строке: {vocales_in_linea}")
```

Рисунок 3 – Код первого задания

2. Результат работы

```
Введите строку символов: Здравствуй  
Гласных в строке: 3
```

Рисунок 4 – Результат работы первого задания

## Задание 2

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

### 1. Код задания 2

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

if __name__ == '__main__':

    # Ввод двух строк
    linea1 = set(input("Введите первую строку символов: "))
    linea2 = set(input("Введите вторую строку символов: "))

    # Вывод общих символов для двух строк
    print(f"Общие символы для строк: {linea1.intersection(linea2)}")
```

Рисунок 5 – Код второго задания

### 2. Результат работы

```
Введите первую строку символов: Здравствуйте
Введите вторую строку символов: Мефодий
Общие символы для строк: {'й', 'д', 'е'}
```

Рисунок 6 – Результат работы задания 2

## Индивидуальное задание

### 1. Код индивидуального задания

```

# Универсальное множество всех букв
all = set("abcdefghijklmnopqrstuvwxyz")

# Переменные из учебника
a = {'c', 'g', 'h', 'i', 'j'}
b = {'c', 'd', 'i', 'o', 's'}
c = {'i', 'j', 'r', 'z'}
d = {'b', 'c', 'f', 'i', 'w', 'x'}

# Операции из учебника
x = (a.union(b)).intersection(d)
y = (a.difference(d)).union((all - c).difference(all - b))

# Вывод результатов
print(x)
print(y)

```

Рисунок 7 – Код индивидуального задания

2. Результат выполнения:

```

{'c', 'i'}
{'o', 'c', 's', 'j', 'g', 'd', 'h'}

```

Рисунок 8 – Результат работы индивидуального задания

### Контрольные вопросы:

1. Что такое множества в языке Python?

Множеством в языке программирования Python называется неупорядоченная совокупность уникальных значений.

2. Как осуществляется создание множеств в Python?

$A = \{1, 2, 0, 1, 3, 2\}$

3. Как проверить присутствие/отсутствие элемента в множестве?

С помощью операторов `in` и `not in`

4. Как выполнить перебор элементов множества?

```
for a in {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.

Следующий код демонстрирует генерацию множества `a` с циклом `for` для нескольких чисел: `a = {i for i in [1, 2, 0, 1, 3, 2]}`

#### 6. Как выполнить добавление элемента во множество?

С помощью метода `.add`

#### 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.

Для полной очистки множества используется `clear`

#### 8. Как выполняются основные операции над множествами: объединение, пересечение, разность?

Объединение — `union`, пересечение — `intersection`, разность — `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`, как в следующем примере. Так как не все элементы набора чисел `a` присутствуют в `b`, функция вернет `False`.

Например:

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

```
b = {3, 2, 1}
```

```
print(a.issubset(b))
```

```
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`.

Например:

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

```
b = {3, 2, 1}
```

```
print(a.issuperset(b))
```

```
True
```

10. Каково назначение множеств `frozenset` ?

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

```
a = frozenset({"hello", "world"})
```

```
print(a)
```

```
frozenset({'hello', 'world'})
```

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. Запятая в кавычках выступает в качестве символа, разделяющего значения.

```
a = {'set', 'str', 'dict', 'list'}
```

```
b = ','.join(a)
print(b)
print(type(b))
set,dict,list,str
<class 'str'>
```

Чтобы получить из множества словарь, следует передать функции `dict` набор из нескольких пар значений, в каждом из которых будет находиться ключ.

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

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.

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

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