

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ

Отчет о лабораторной работе №10 по дисциплине  
«Основы программной инженерии»

Выполнил:  
Прокопов Дмитрий  
Владиславович,  
2 курс, группа ПИЖ-б-о-21-1,

Проверил:  
Доцент кафедры инфокоммуникаций,  
Воронкин Р.А.

Ставрополь, 2022 г

## Ход работы

### Пример

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

if __name__ == "__main__":
    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}")
    bn = u.difference(b)
    cn = u.difference(c)
    y = (a.difference(d)).union(cn.difference(bn))
    print(f"y={y}")
```

C:\Users\dmidt\PycharmProjects\lb10\venv\Scripts>python3 script.py

x={'o', 'j', 'd', 'e', 'k'}

y={'o', 'c', 'g', 'y', 'h', 'v', 'f'}

### Индивидуальная работа

$$A = \{b, c, h, i, j\}; \quad B = \{e, h, i, s, w\}; \quad C = \{a, b, j, k, l, m\}; \quad D = \{a, h, i, w, x\};$$
$$X = (A/C) \cap \bar{B}; \quad Y = (A \cap \bar{B}) \cup (C/D).$$

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

if __name__ == "__main__":
    u = set("abcdefghijklmnopqrstuvwxyz")
    a = {"b", "c", "h", "i", "j"}
    b = {"e", "h", "i", "s", "w"}
    c = {"a", "b", "j", "k", "l", "m"}
    d = {"a", "h", "i", "w", "x"}
    bn = u.difference(b)
    ac = a.difference(c)
    x = ac.intersection(bn)
    print(f"x={x}")
    ab = a.intersection(bn)
    cd = c.difference(d)
    y = ab.union(cd)
    print(f"y={y}")
```

C:\Users\dmidt\PycharmProjects\lb10\venv\Scripts>python3 script.py

x={'c'}

y={'m', 'l', 'b', 'c', 'k', 'j'}

## Ответы на вопросы

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

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

2. Как осуществляется создание множеств в Python? Сделать это можно, просто присвоив переменной последовательность значений, выделив их фигурными скобками.

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

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

Существует и другой способ создания множеств, который подразумевает использование вызова `set`.

```
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`.

## 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'>
```