

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: Грובה
Софья Кирилловна,
2 курс, группа ПИЖ-б-о-20-1,
Проверил: Доцент кафедры
инфокоммуникаций, Воронкин Р.А.

Ставрополь, 2021 г

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

Пример операции над множествами

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

if __name__ == '__main__':
    u = set('abcdefghijklmnopqrstuvwxyz')
    a = {'a', 'f', 'i', 'n', 'o'}
    b = {'f', 'g', 'o', 'p', 'z'}
    c = {'i', 'j', 'u', 'w'}
    d = {'f', 'h', 'n', 't', 'u', 'y', 'z'}
    x = b.union(a.intersection(c))
    y = a.intersection(u.difference(b)).union(c.difference(d))
    print(f'x = {x}')
    print(f'y = {y}')
```

Код программы индивидуального задания (вариант 7)

Контрольные вопросы:

1. Что такое множества в языке Python

Множество в языке Python – это структура данных, эквивалентная множествам в математике. **Множество** может состоять из различных элементов, порядок элементов в **множестве** неопределен.

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

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

a = {0, 1, 2, 3}
print(2 not in a)
#False
```

5. Что такое set comprehension

Set comprehensions (генераторы множеств)

Генераторы множеств в целом аналогичны генераторам списков.

```
a = {i for i in [1, 2, 0, 1, 3, 2]}
print(a)
#{0, 1, 2, 3}
```

6. Как выполнить добавление элемента во множество

```
#Добавление элемента
a = {0, 1, 2, 3}
a.add(4)
print(a)
#{0, 1, 2, 3, 4}
```

7. Как выполнить удаление одного или всех элементов множества

```

#Удаление элемента
a = {0, 1, 2, 3}
a.remove(3)
print(a)
#{0, 1, 2}

#Полная очистка
a = {0, 1, 2, 3}
a.clear()
print(a)
#set()

```

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

```

#Объединение
a = {0, 1, 2, 3}
b = {4, 3, 2, 1}
c = a.union(b)
print(c)
#{0, 1, 2, 3, 4}

```

```

#Пересечение
a = {0, 1, 2, 3}
b = {4, 3, 2, 1}
c = a.intersection(b)
print(c)
#{1, 2, 3}

#Разность
a = {0, 1, 2, 3}
b = {4, 3, 2, 1}
c = a.difference(b)
print(c)
#{0}

```

9. Как определить, что некоторое множество является надмножеством или подмножеством другого множества

```

#Определение подмножества
a = {0, 1, 2, 3, 4}
b = {3, 2, 1}
print(a.issubset(b))
#False

#Определение надмножества
a = {0, 1, 2, 3, 4}
b = {3, 2, 1}
print(a.issuperset(b))
#True

```

10. Какого назначение множеств frozenset

Единственное отличие set от frozenset заключается в том, что set - изменяемый тип данных, а frozenset - нет. Примерно похожая ситуация с [списками](#) и [кортежами](#).

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'>
```