

**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
Работа с множествами в языке Python.**

по дисциплине «Технологии программирования и алгоритмизации»

Выполнила студентка группы ИВТ-б-о-20-1

Новикова В.С. « » _____ 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.

Ход работы:

Ссылка на репозиторий:

Пример 1. (рис. 1).

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}")
```

```
C:\Users\Валентина\AppData\Local\Programs\Python\Python38\Scripts>python3
x = {'o', 'd', 'j', 'e', 'k'}
y = {'o', 'h', 'v', 'g', 'y', 'f', 'c'}

Process finished with exit code 0
```

Рисунок 1 – Пример 1

Задача 1 (рис. 2):

Подсчитать количество гласных в строке, введенной с клавиатуры с использованием множеств

Код:

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

if __name__ == "__main__":
    # Определим универсальное множество
    u = set("aeiouy")
    d = 0

    # ввод строки
    a = input("Введите строку: ")
    for i in a:
        if i in u:
```

```
d += 1
print("Количество гласных: ", d)
```

```
C:\Users\Валентина\AppData\Local\Programs\Python\Python39\python.exe
Введите строку: аіауі аsyusyus doidois
Количество гласных: 14

Process finished with exit code 0
```

Рисунок 2 – Задача 1

Задача 2 (рис. 3):

Определить общие символы в двух строках, введенных с клавиатуры
Код:

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

if __name__ == "__main__":
    # Определим универсальное множество
    u = set("abcdefghijklmnopqrstuvwxyz")

    # ввод строк
    a = set(input("Введите 1 строку: "))
    b = set(input("Введите 2 строку: "))

    c = a.intersection(b)

    print("Пересечение строк: ", c)
```

```
C:\Users\Валентина\AppData\Local\Programs\Python\Python39\python.exe
Введите 1 строку: ыгшшвс зукршгуц втиво
Введите 2 строку: ырывгргшс цкцкцкеекв
Пересечение строк: {'с', 'ы', 'к', 'в', ' ', 'р', 'ш', 'ц', 'г'}

Process finished with exit code 0
```

Рисунок 3 – Задача 2

Вариант 11

Определить результат выполнения операций над множествами. Считать элементы множества строками. Проверить результат вручную (рис. 4-5).

11.

$$A = \{a, h, k\}; \quad B = \{c, d, h, p, r\}; \quad C = \{h, i, s\}; \quad D = \{c, g, j, v, w\};$$

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

Рисунок 4 – Задание

Код:

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

if __name__ == "__main__":
    # Определим универсальное множество
    u = set("abcdefghijklmnopqrstuvwxyz")
```

```

a = {"a", "h", "k"}
b = {"c", "d", "h", "p", "r"}
c = {"h", "i", "s"}
d = {"c", "g", "j", "v", "w"}

x = (a.union(b)).intersection(c)
print(f"x = {x}")

# Найдем дополнения множеств
an = u.difference(a)
bn = u.difference(b)
a_b = an.intersection(bn)
y = a_b.difference(c.union(d))
print(f"y = {y}")

```

```

C:\Users\Валентина\AppData\Local\Programs\Python\Python39\python.exe C
x = {'h'}
y = {'x', 'o', 'q', 't', 'l', 'e', 'm', 'b', 'f', 'y', 'z', 'u', 'n'}

Process finished with exit code 0
|

```

Рисунок 5 – Индивидуальное задание

Контрольные вопросы:

1. Что такое множества в языке Python?

Множеством в языке программирования 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?

Создать множество можно, присвоив переменной последовательность значений, выделив их фигурными скобками.

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

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)
```

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.

```
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`.

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

```
a.add(4)
```

7. Как выполнить удаление одного или всех элементов множества?

`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?

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

```
b = {4, 3, 2, 1}
```

Объединение:

```
c = a.union(b)
```

Пересечение:

```
c = a.intersection(b)
```

Разность:

```
c = a.difference(b)
```

9. Как определить, что некоторое множество является над множеством 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. Каково назначение множеств frozenset?

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

```
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.

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

```
b = ','.join(a)
```

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)
```

Для преобразования в список используется вызов `list`, получающий в качестве аргумента множество `a`.

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

```
b = list(a)
```

Вывод: при выполнении заданий были приобретены навыки по работе с множествами при написании программ с помощью языка программирования Python.