

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 №10
дисциплины «Программирование на Python»

Выполнил:

Лейс Алексей Вячеславович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем»

(подпись)

Руководитель практики: кандидат тех.
наук доцент кафедры
инфокоммуникаций: Воронкин Р.А

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023г.

Тема: Работа с множествами в языке Python

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

Порядок выполнения работы:

Пример 1:

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  if __name__ == "__main__":
6      # Определим универсальное множество
7      u = set("abcdefghijklmnopqrstuvwxyz")
8      a = {"b", "c", "h", "o"}
9      b = {"d", "f", "g", "o", "v", "y"}
10     c = {"d", "e", "j", "k"}
11     d = {"a", "b", "f", "g"}
12     x = (a.intersection(b)).union(c)
13     print(f"x = {x}")
14     # Найдём дополнения множеств
15     bn = u.difference(b)
16     cn = u.difference(c)
17     y = (a.difference(d)).union(cn.difference(bn))
18     print(f"y = {y}")
```

Индивидуальное задание:

$$\begin{aligned} C &= \{c, d, k, l, y\}; & D &= \{a, g, h, u, v, z\}; \\ Y &= (\bar{A} \cap D) \cup (C/B). \end{aligned} \tag{19}$$

19.

$$X = (A \cap B) \cup C;$$

$$A = \{c, g, h, k, y\}; \quad B = \{a, b, k, n, u\};$$

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5
6 # Заданные множества строк
7 A = {'c', 'g', 'h', 'k', 'y'}
8 B = {'a', 'b', 'k', 'n', 'u'}
9 C = {'c', 'd', 'k', 'l', 'y'}
10 D = {'a', 'g', 'h', 'u', 'v', 'z'}
11 #u = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h',
12 # 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r',
13 # 's', 't', 'u', 'v', 'w', 'x', 'y', 'z'}
14 g = set("abcdefghijklmnopqrstuvwxyz")
15 # Операции с использованием union вместо or
16
17 an = g.difference(A)
18
19 Y = (an.intersection(D)).union(C.difference(B))
20 X = A.intersection(B).union(C)
21
22 # Вывод результатов
23 print(f"Результат Y: {Y}")
24 print(f"Результат X: {X}")

```

Ответы на вопросы:

1. Что такое множества в языке Python?

- Множества в Python представляют собой неупорядоченные коллекции уникальных элементов. Они используются для выполнения операций над множествами, таких как объединение, пересечение и разность.

2. Как осуществляется создание множеств в Python?

- Множества создаются с использованием фигурных скобок или функции `set()`, например: `my_set = {1, 2, 3}`.

3. Как проверить присутствие/отсутствие элемента в множестве?

- Присутствие элемента можно проверить с использованием оператора `in`, например: `result = element in my_set`.

4. Как выполнить перебор элементов множества?

- Элементы множества можно перебирать с использованием цикла `for`, например: `for item in my_set: print(item)`.

5. Что такое set comprehension?

- Set comprehension - это компактный способ создания множества в одной строке с использованием синтаксиса, аналогичного списковому включению, например: ``my_set = {x for x in iterable}``.

6. Как выполнить добавление элемента во множество?

- Элемент можно добавить во множество с использованием метода ``add()``, например: ``my_set.add(new_element)``.

7. Как выполнить удаление одного или всех элементов множества?

- Один элемент можно удалить с использованием метода ``remove()`` или ``discard()``, а все элементы - методом ``clear()``.

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

- Операции выполняются с использованием соответствующих методов или операторов: объединение (``union()`` или ``|``), пересечение (``intersection()`` или ``&``), разность (``difference()`` или ``-``).

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

- Это можно проверить с использованием методов ``issuperset()`` и ``issubset()`` соответственно.

10. Каково назначение множеств ``frozenset``?

- ``frozenset`` - это неизменяемая версия множества. Его элементы нельзя изменять после создания, что делает его подходящим для использования в качестве ключей в словарях.

11. Как осуществляется преобразование множеств в строку, список, словарь?

- Преобразование можно выполнить с использованием соответствующих функций: ``str(my_set)``, ``list(my_set)``, ``dict.fromkeys(my_set)``.

Вывод В ходе работы приобрёл навыки по работе с множествами при написании программ с помощью языка программирования Python версии 3.x.