

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Отчет по лабораторной работе №2.7

Работа с множествами в языке Python3.

по дисциплине «Технологии программирования и алгоритмизации»

Выполнил студент группы ИВТ-б-о-20-1

Малышев А.Ю. « »_____20__г.

Подпись студента_____

Работа защищена « »_____20__г.

Проверила Воронкин Р.А. _____

(подпись)

Ставрополь 2021

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

Ход работы: <https://github.com/AlexandrM333/labrab2.7.git>

1. Проработал пример:

```
C:\Games\python.exe C:/Users/Александр/projects/labrab2.7/primer1.py
x = {'j', 'd', 'e', 'o', 'k'}
y = {'y', 'c', 'f', 'h', 'v', 'g', 'o'}
```

Рисунок 1. Вывод примера.

2. Выполнил 1-ое задание:

```
C:\Games\python.exe C:/Users/Александр/projects/labrab2.7/zadan1.py
Введите слово: Привет
2
Введите слово: Как дела
3
```

Рисунок 2. Выполненное первое задание.

3. Выполнил второе задание:

```
C:\Games\python.exe C:/Users/Александр/projects/labrab2.7/zadan2.py
Введите 1-ое слово: огурец
Введите 2-ое слово: помидор
Одинаковые символы: {'o', 'p'}
```

Рисунок 3. Выполненное второе задание.

4. Выполнил индивидуальное задание:

Вариант – 9

```
C:\Games\python.exe C:/Users/Александр/projects/labrab2.7/indiv.py
X = {'n', 'e', 'f'}
Y = {'s', 'h', 'c', 'z', 'q', 'r', 'y', 'g', 't', 'v', 'j', 'u', 'l', 'm'}
```

Рисунок 4. Выполненное индивидуальное задание.

5. Проверил программы на flake8:

```
Anaconda Powershell Prompt (Games)
(base) PS C:\Users\Александр> cd projects
(base) PS C:\Users\Александр\projects> cd labrab2.7
(base) PS C:\Users\Александр\projects\labrab2.7> flake8
(base) PS C:\Users\Александр\projects\labrab2.7>
```

Рисунок 5. Проверка программ.

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. В отличие от массивов и списков, порядок следования значений не учитывается при обработке его содержимого.

2. Как осуществляется создание множеств в Python?

Для создания множества нужно просто присвоить переменной последовательность значений, выделив их фигурными скобками:

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

Существует и другой способ создания множеств, который подразумевает использование вызова `set`. Аргументом этой функции может быть набор неких данных или даже строка с текстом.

3. Как проверить присутствие/отсутствие элемента в множестве?

Для этого используется `in`:

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

```
print(2 in a)
```

```
True
```

4. Как выполнить перебор элементов множества?

Перебор всех элементов выполняется циклом `for`:

```
for a in {0, 1, 2}:
```

```
print(a)
```

5. Что такое `set comprehension`?

Для создания множества в Python можно воспользоваться генератором Set Comprehensions, позволяющим 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`.

Аргументом в данном случае будет добавляемый элемент последовательности.

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.

Чтобы полностью убрать все элементы, надо использовать метод `clear`, не принимающий аргументов. Если вывести содержимое после этой операции, на экране появится только его название

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:

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

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

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

Чтобы найти общие элементы для двух разных множеств, следует применить функцию `intersection`, принимающую в качестве аргумента один из наборов данных: `c = a.intersection(b)`

Чтобы вычислить разность для двух разных множеств, необходимо воспользоваться методом `difference`. Функция позволяет найти элементы, уникальные для второго набора данных, которых в нем нет:

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

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

Для определения подмножеств и надмножеств существуют специальные функции, возвращающие True или False в зависимости от результата выполнения. Для определения принадлежности элемента к множеству используется оператор `in`. Для того, чтобы проверить, что элемент не входит в множество используется оператор `not in`.

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?

Для преобразования множества в строку используется конкатенация текстовых значений, которую обеспечивает функция `join`.

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

По аналогии с предыдущими преобразованиями можно получить список неких объектов. На этот раз используется вызов `list`, получающий в качестве аргумента множество `a`.

Вывод: в ходе работы были приобретены навыки по работе с множествами при написании программ с помощью языка программирования Python версии 3.x.