

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций
Институт цифрового развития

ОТЧЁТ

по лабораторной работе №2.7

Дисциплина: «Основы кроссплатформенного программирования»

Тема: «Работа с множествами в
языке Python»

Выполнил: студент 1 курса
группы ИВТ-б-о-21-1
Толубаев Рамиль Ахметович

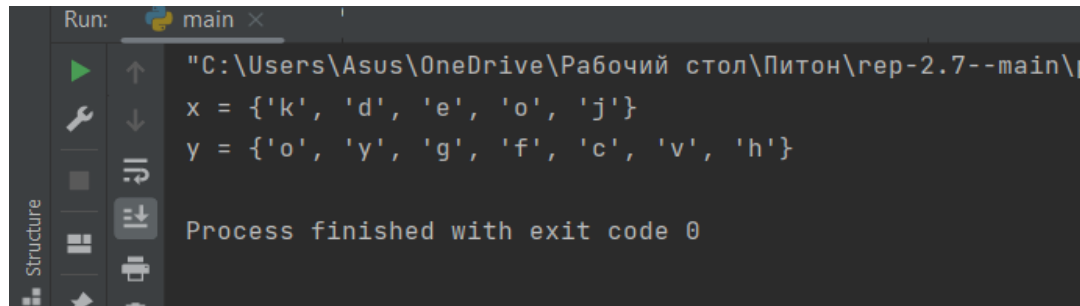
Ставрополь 2022

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

Ход работы:

Пример 1. Определить результат выполнения операций над множествами.

Считать элементы множества строками.

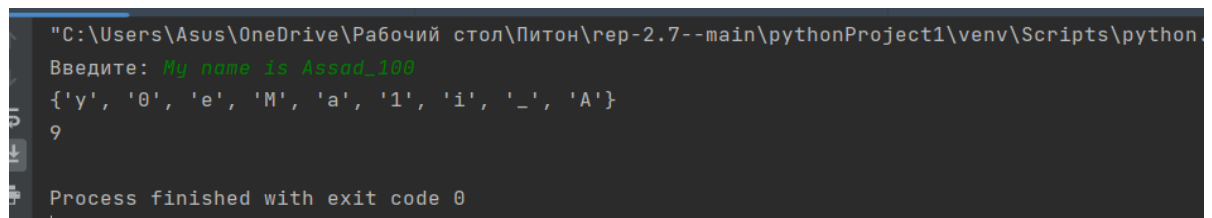


```
Run: main x
"C:\Users\Asus\OneDrive\Рабочий стол\Питон\rep-2.7--main\
x = {'k', 'd', 'e', 'o', 'j'}
y = {'o', 'y', 'g', 'f', 'c', 'v', 'h'}

Process finished with exit code 0
```

Рисунок 1. . Результаты выполнения программы

Задача 1. Подсчитайте количество гласных в строке, введенной с клавиатуры с использованием множеств.

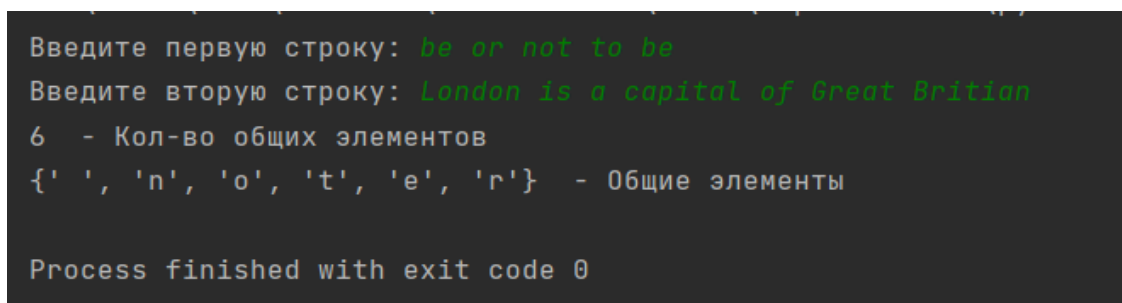


```
"C:\Users\Asus\OneDrive\Рабочий стол\Питон\rep-2.7--main\pythonProject1\venv\Scripts\python
Введите: My name is Assad_100
{'y', 'o', 'e', 'M', 'a', 'l', 'i', '_', 'A'}
9

Process finished with exit code 0
```

Рисунок 2. Результаты выполнения программ

Задача 2. Определите общие символы в двух строках, введенных с клавиатуры.

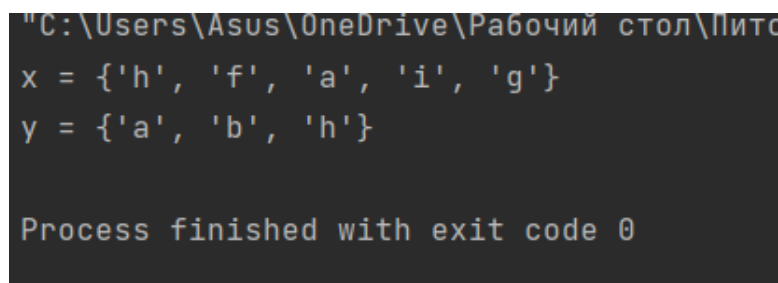


```
Введите первую строку: be or not to be
Введите вторую строку: London is a capital of Great Britian
6 - Кол-во общих элементов
{' ', 'n', 'o', 't', 'e', 'r'} - Общие элементы

Process finished with exit code 0
```

Рисунок 3. Результаты выполнения программы

Индивидуальное задание



```
"C:\Users\Asus\OneDrive\Рабочий стол\Питон
x = {'h', 'f', 'a', 'i', 'g'}
y = {'a', 'b', 'h'}

Process finished with exit code 0
```

Рисунок 4. Результаты выполнения программы

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?

Сделать это можно, просто присвоив переменной последовательность значений, выделив их фигурными скобками.

3. Как проверить присутствие/отсутствие элемента в множестве?

Проверка, есть ли данное значение в множестве. Для этого используется `in`. Наоборот, проверка отсутствия. Используется `not in`.

4. Как выполнить перебор элементов множества?

Перебрать все элементы множества (в неопределенном порядке) можно при помощи цикла `for`.

5. Что такое `set comprehension`?

`Set comprehension` – генератор, позволяющий заполнять списки, а также другие наборы данных с учетом неких условий.

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 не удалять каждый элемент отдельно,

используется 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. Чтобы найти общие элементы для двух разных множеств (пересечение), следует применить функцию `intersection` , принимающую в качестве аргумента один из наборов данных. Чтобы вычислить разность для двух разных множеств, необходимо воспользоваться методом `difference`. Функция позволяет найти элементы, уникальные для второго набора данных, которых в нем нет.

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. Чтобы выяснить, является ли множество `a` подмножеством `b`, стоит попробовать вывести на экран результат выполнения метода `issubset`, `print(a.issubset(b))`. Чтобы узнать, является ли множество `a` надмножеством `b`, необходимо вызвать метод `issuperset` и вывести результат его работы на экран, `print(a.issuperset(b))`.

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`. В этом случае ее аргументом является набор данных в виде нескольких строк. Можно получить список неких объектов. На этот раз используется вызов `list` , получающий в качестве аргумента множество `a`. Чтобы получить из множества словарь, следует

передать функции dict набор из нескольких пар значений, в каждом из которых будет находиться ключ.

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