

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

**«Лабораторная работа 2.7 Работа с множествами в
языке Python»**

ОТЧЕТ
по лабораторной работе №10
дисциплины
«Основы программной инженерии»

Выполнил:

Луценко Дмитрий Андреевич
2 курс, группа ПИЖ-б-о-21-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил:

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

Лабораторная работа 2.7 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.

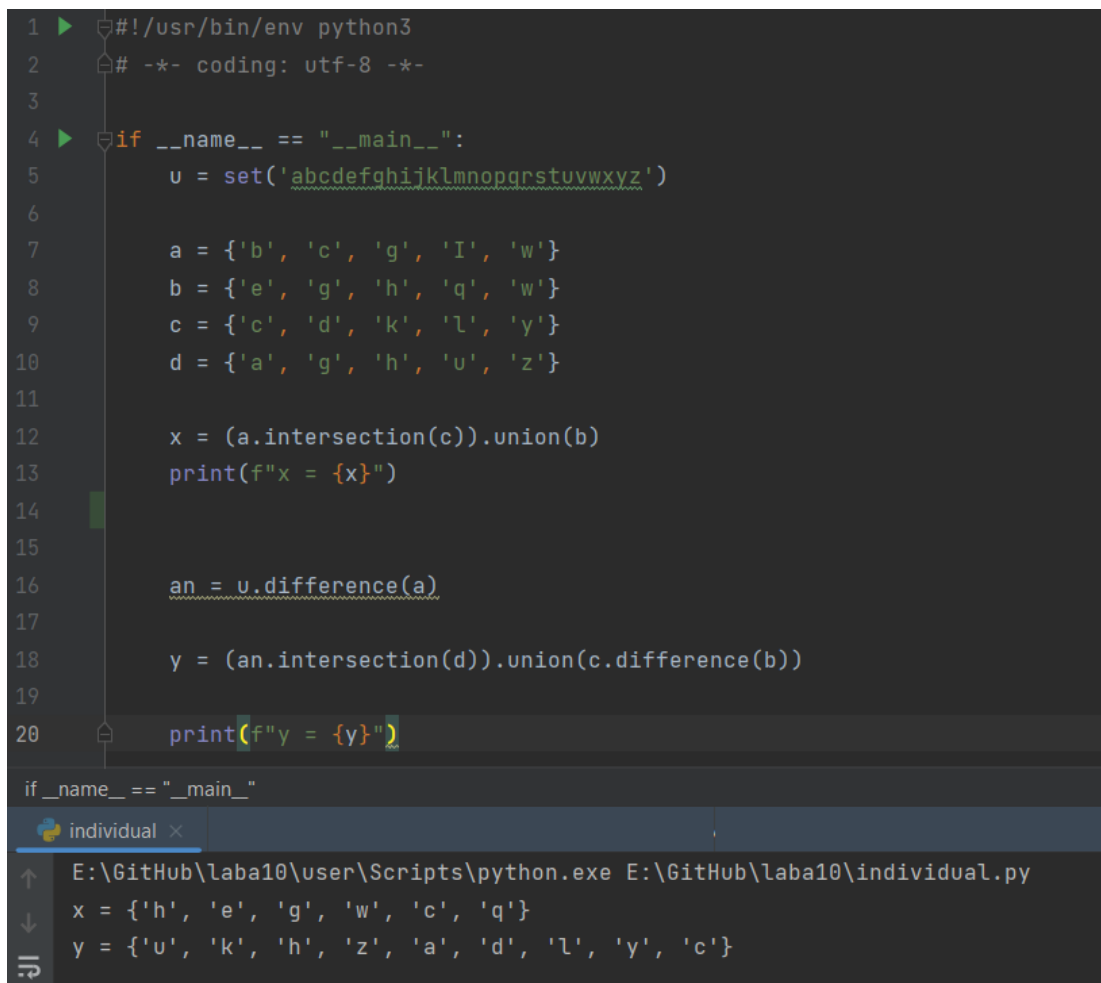
Ход работы:

Индивидуальное задание

$$A = \{b, c, g, I, w\}; \quad B = \{e, g, h, q, w\}; \quad C = \{c, d, k, l, y\}; \quad D = \{a, g, h, u, v, z\};$$

18.

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



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == "__main__":
5      u = set('abcdefghijklmnopqrstuvwxyz')
6
7      a = {'b', 'c', 'g', 'I', 'w'}
8      b = {'e', 'g', 'h', 'q', 'w'}
9      c = {'c', 'd', 'k', 'l', 'y'}
10     d = {'a', 'g', 'h', 'u', 'z'}
11
12     x = (a.intersection(c)).union(b)
13     print(f"x = {x}")
14
15
16     an = u.difference(a)
17
18     y = (an.intersection(d)).union(c.difference(b))
19
20     print(f"y = {y}")
if __name__ == "__main__"
```

individual ×

E:\GitHub\laba10\user\Scripts\python.exe E:\GitHub\laba10\individual.py

x = {'h', 'e', 'g', 'w', 'c', 'q'}

y = {'u', 'k', 'h', 'z', 'a', 'd', 'l', 'y', 'c'}

Рисунок 1 – Индивидуальное задание

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

Ответы на 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. Над одним, а также несколькими множествами можно выполнять ряд операций, благодаря функциям стандартной библиотеки языка программирования Python.

2. Как осуществляется создание множеств в Python? Присвоить переменной последовательность значений в фигурных скобках.

3. Как проверить присутствие/отсутствие элемента в множестве? `Print(<элемент> in <множество>)`. Результатом будет `True` или `False`.

4. Как выполнить перебор элементов множества? Через цикл `for a in {...}`

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.

6. Как выполнить добавление элемента во множество? Чтобы внести новые значения, потребуется вызывать метод `add`. Аргументом в данном случае будет добавляемый элемент последовательности. В примере кода на Python добавим в множество элемент со значением 4.

7. Как выполнить удаление одного или всех элементов множества? Для удаления элементов из множества используются следующие функции в Python (кроме очистки, которая будет рассмотрена ниже): `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. Избавиться от лишних значений в наборе данных с помощью `remove`. В качестве входного параметра здесь выступает элемент, который нужно удалить (в примере удалим число со значением 3).

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 добавить все элементы из одного множества к другому, необходимо вызывать метод `update` на первом объекте. Чтобы найти общие элементы для двух разных множеств, следует применить функцию `intersection`, принимающую в качестве аргумента один из наборов данных. Чтобы вычислить разность для двух разных множеств, необходимо воспользоваться методом `difference`.

9. Как определить, что некоторое множество является надмножеством или подмножеством другого множества? Чтобы выяснить, является ли множество `a` подмножеством `b`, стоит попробовать вывести на экран результат выполнения метода `issubset`. Чтобы узнать, является ли множество `a` надмножеством `b`, необходимо вызвать метод `issuperset` и вывести результат его работы на экран.

10. Каково назначение множеств `frozenset`? Множество, содержимое которого не поддается изменению имеет тип `frozenset`. Значения из этого набора нельзя удалить, как и добавить новые. В следующем примере демонстрируется создание при помощи стандартной функции.

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'>
```