

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

Институт цифрового развития
Кафедра инфокоммуникаций

«Работа с множествами в языке Python»

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

Выполнил:
Зиёдуллаев Жавохир Эркин угли
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.

Пример 1

```
#!/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}")
```

set1 x

```
C:\Users\work\PycharmProjects\pythonProject4\venv\Scripts\python.exe C:\Users\work\lab-
x = {'d', 'j', 'o', 'k', 'e'}
y = {'f', 'c', 'o', 'v', 'h', 'y', 'g'}
```

Задание 10

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

if __name__ == "__main__":
    s1 = input("put first text: ")
    s1 = s1.replace(' ', '')
    a = {i for i in s1.lower()}
    s2 = input("put second text: ")
    s2 = s2.replace(' ', '')
    b = {i for i in s2.lower()}
    c = a.intersection(b)
    print(f"Intersection: {c}")
```

Задание10 ×

C:\Users\work\PycharmProjects\pythonProject4\venv\Scripts\python.exe C:\Users\work\1
put first text: *python is good*
put second text: *python is better java*
Intersection: {'i', 'h', 'o', 'y', 't', 'p', 'n', 's'}

Индивидуальное задание

$$A = \{a, b, d, I, x\}; \quad B = \{d, e, h, i, n, u\}; \quad C = \{e, f, m, n\}; \quad D = \{a, c, h, k, r, s, w, x\};$$
$$X = (A/C) \cap \bar{B}; \quad Y = (\bar{A} \cap D) \cup (C/B).$$

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

if __name__ == "__main__":
    u = set("abcdefghijklmnopqrstuvwxyz")

    a = {"a", "b", "d", "i", "x"}
    b = {"d", "e", "h", "i", "n", "u"}
    c = {"e", "f", "m", "n"}
    d = {"a", "c", "h", "k", "r", "s", "w", "x"}

    bu = u.difference(b)
    x = (a.difference(c)).intersection(bu)

    print(f"x = {x}")

    ba = u.difference(a)
    y = (ba.intersection(d)).union(c.difference(b))
    print(f"y = {y}")

__name__ == "__main__"
```

setindiv ×

C:\Users\work\PycharmProjects\pythonProject4\venv\Scripts\python.exe C:\Use

x = {'x', 'a', 'b'}

y = {'w', 'k', 'h', 'f', 'm', 'r', 's'}

Ссылка: <https://github.com/javoxir21/laba-10.git>

Контрольные вопросы:

1. Что такое множества в языке Python? Множеством в Python называется неупорядоченная совокупность уникальных значений. В качестве элементов этого набор 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.

```
a = {1, 2, 0, 1, 3, 2}
print(a)

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

Существует и другой способ создания множеств, который подразумевает использование вызова `set`.

```
a = set('data')
print(a)

{'d', 'a', 't'}
```

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)

0
1
2
```

5. Что такое set comprehensions?

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`.

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.

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, следует применить функцию `intersection`.

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.

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

Строка. 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. Запятая в кавычках выступает в качестве символа, разделяющего значения. Метод `type` возвращает тип данных объекта в

```
a = {'set', 'str', 'dict', 'list'}
b = ','.join(a)
print(b)
print(type(b))

set,dict,list,str
<class 'str'>
```

конце приведенного кода.

Словарь. Чтобы получить из множества словарь, следует передать функции `dict` набор из нескольких пар значений, в каждом из которых будет находиться ключ. Функция `print` демонстрирует на экране содержимое полученного объекта, а `type` отображает его тип.

```
a = {('a', 2), ('b', 4)}
b = dict(a)
print(b)
print(type(b))

{'b': 4, 'a': 2}
<class 'dict'>
```

Список. На этот раз используется вызов `list`, получающий в качестве аргумента множество `a`. На выходе функции `print` отображаются уникальные значения для изначального набора чисел.

```
a = {1, 2, 0, 1, 3, 2}
b = list(a)
print(b)
print(type(b))

[0, 1, 2, 3]
<class 'list'>
```

