

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:

Мизин Глеб Егорович

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.

Проработка примера из лабораторной работы:

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == "__main__":
5
6     # Определим универсальное множество
7     u = set("abcdefghijklmnopqrstuvwxyz")
8
9     a = {"b", "c", "h", "o"}
10    b = {"d", "f", "g", "o", "v", "y"}
11    c = {"d", "e", "j", "k"}
12    d = {"a", "b", "f", "g"}
13
14    x = (a.intersection(b)).union(c)
15    print(f"x = {x}")
16
17    # Найдем дополнения множеств
18    bn = u.difference(b)
19    cn = u.difference(c)
20
21    y = (a.difference(d)).union(cn.difference(bn))
22    print(f"y = {y}")
23
if __name__ == "__main__"
```

2_task × ex ×

↑ F:\GitLaby\Lab-2.7\venv\Scripts\python.exe F:\GitLaby\Lab-2.7\ex.py
↓ x = {'e', 'j', 'o', 'k', 'd'}
y = {'v', 'y', 'g', 'h', 'f', 'o', 'c'}

Рисунок 1 – код и результат работы программы из примера

Задание №1: решите задачу: подсчитайте количество гласных в строке, введенной с клавиатуры с использованием множеств.

```
1  ▶  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  ▶  if __name__ == "__main__":
5
6      punc_marks = set(".?;!,:;`' ")
7
8      # Input
9      s_1 = set(input("Enter the first string: "))
10     s_2 = set(input("Enter the second string: "))
11
12     # Finding intersection between s_1 and s_2
13     inter = s_1.intersection(s_2)
14
15     # Remove all punctuation marks
16     clear_set = inter.difference(punc_marks)
17
18     # Output of intersection
19     print(f"Same string characters: {clear_set}")
20
if __name__ == "__main__"
```

Run: 2_task ×

F:\GitLab\Lab-2.7\venv\Scripts\python.exe F:\GitLab\Lab-2.7\2_task.py
Enter the first string: *I want to break free*
Enter the second string: *Me too want!*
Same string characters: {'e', 'a', 'o', 'w', 'n', 't'}

Рисунок 2 – код и результат работы программы для задания №1

Задание №2: решите задачу: определите общие символы в двух строках, введенных с клавиатуры.

```
1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4  ▶  if __name__ == "__main__":
5
6      # All vowels letters
7      vowels = set("aeiouyYUOIEA")
8
9      # Input
10     s = input("Enter the string: ")
11
12     # Finding amount of vowels
13     kol_vo = 0
14     for i in s:
15         if i in vowels:
16             kol_vo += 1
17
18     # Output
19     print(f"Number of vowels: {kol_vo}")
20
if __name__ == "__main__":
Run: 1_taks ×
▶  F:\GitLab\Lab-2.7\venv\Scripts\python.exe F:\GitLab\Lab-2.7\1_taks.py
Enter the string: I want to break free!
Number of vowels: 7
Process finished with exit code 0
```

Рисунок 3 – код и результат работы программы для задания №2

Индивидуальное задание:

$$A = \{c, g, h, k, y\}; \quad B = \{a, b, k, n, u\}; \quad C = \{i, j, o, y, z\}; \quad D = \{a, b, f, g, y, z\};$$
$$X = (A \cup B) \cap D; \quad Y = (\bar{A} \cap D) \cup (\bar{C} / \bar{B}).$$

Решение:

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

$$X = (A \cup B) \cap D = \{a, b, g, y\}$$

$$\bar{A} = \{t, p, n, a, x, w, e, l, s, q, d, b, m, z, v, o, f, r, i, j, u\}$$

$$\bar{B} = \{t, p, x, w, y, e, g, l, h, s, q, d, m, z, v, o, c, f, r, i, j\}$$


$$\bar{C} = \{t, p, n, a, x, w, e, g, l, h, s, k, q, d, b, m, v, c, f, r, u\}$$


$$\bar{A} \cap D = \{a, b, f, z\}$$

$$\bar{C} / \bar{B} = \{n, b, k, u, a\}$$

$$Y = (\bar{A} \cap D) \cup (\bar{C} / \bar{B}) = \{n, b, k, u, a, f, z\}$$

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == "__main__":
5
6     # Universe
7     u = set("abcdefghijklmnopqrstuvwxyz")
8
9     # Set Data
10    a = {"c", "g", "h", "k", "y"}
11    b = {"a", "b", "k", "n", "u"}
12    c = {"i", "j", "o", "y", "z"}
13    d = {"a", "b", "f", "g", "y", "z"}
14
15    # Definition X
16    x = d.intersection(a.union(b))
17    print(f'X = {x}')
18
19    # Inverses for a b and c
20    ne_a = u.difference(a)
21    ne_b = u.difference(b)
22    ne_c = u.difference(c)
23
24    # Definition Y
25    y = (ne_a.intersection(d)).union(ne_c.difference(ne_b))
26    print(f'Y = {y}')
27
```

 ind x

 F:\GitLaby\Lab-2.7\venv\Scripts\python.exe F:\GitLaby\Lab-2.7\ind.py

X = {'a', 'g', 'b', 'y'}

Y = {'a', 'f', 'k', 'u', 'b', 'z', 'n'}

Рисунок 4 – код и результат работы программы для индивидуального задания

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?

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

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

```
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 comprehension?

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`. Аргументом в данном случае будет добавляемый элемент последовательности. В примере кода на Python добавим в множество элемент со значением 4.

```
a = {0, 1, 2, 3}  
a.add(4)  
print(a)  
  
{0, 1, 2, 3, 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'>
```