

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**

**Работа с множествами в языке Python**

**По дисциплине «Технологии программирования и алгоритмизация»**

Выполнил студент группы ИВТ-б-о-20-1

Галяс Д. И. « » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверил Воронкин Р. А. \_\_\_\_\_

(подпись)

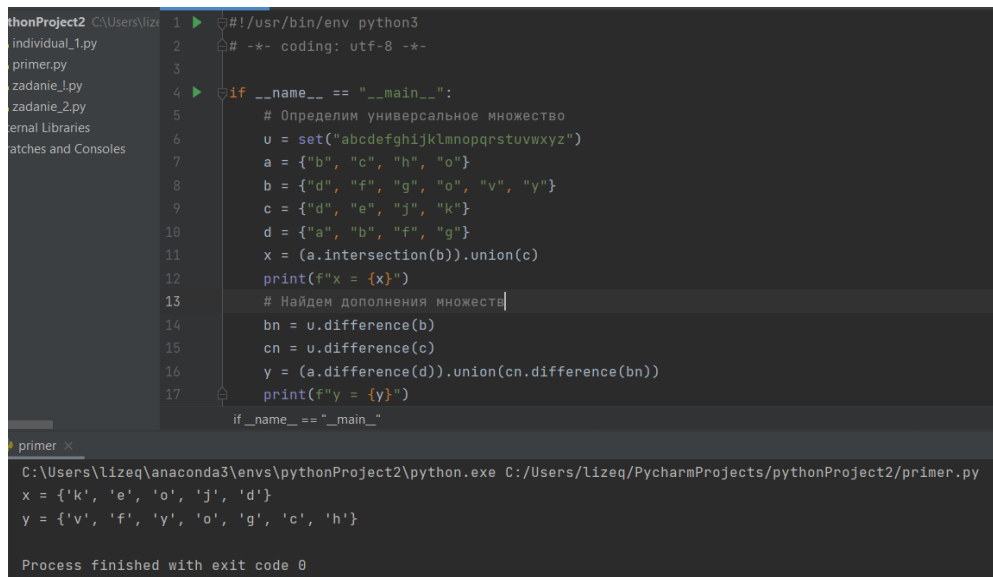
Ставрополь 2021

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

**Ход работы:**

**Ссылка на репозиторий:** <https://github.com/DIMITRY-GALYAS1/Rabota-9.git>

1. Создал новый репозиторий на github, после клонировал его и создал в папке репозитория новый проект PyCharm.
2. Проработал пример из лабораторной работы.



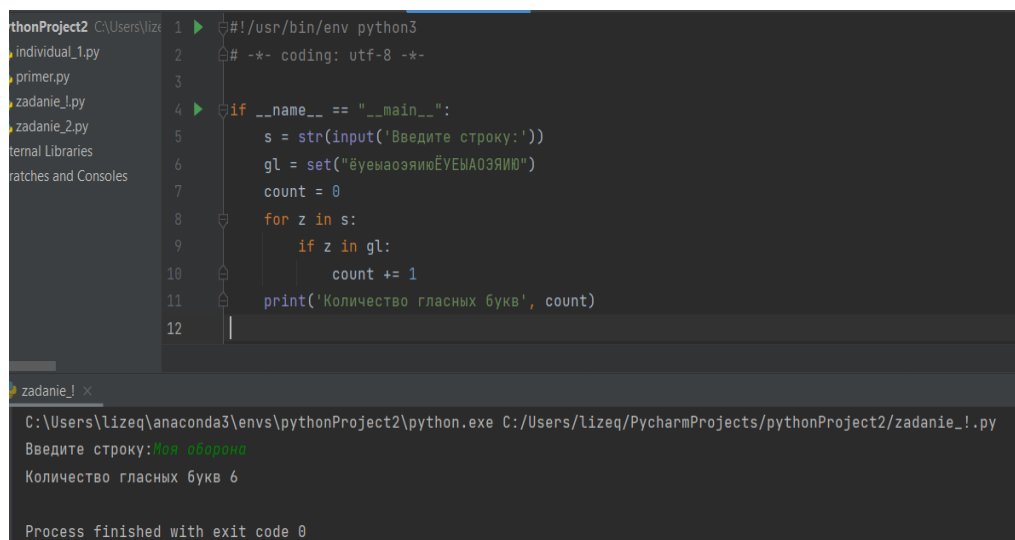
```
pythonProject2 C:\Users\lizeq> #!/usr/bin/env python3
1
individual_1.py 2 # -*- coding: utf-8 -*-
3
primer.py 4 if __name__ == "__main__":
5     # Определим универсальное множество
6     u = set("abcdefghijklmnopqrstuvwxyz")
7     a = {"b", "c", "h", "o"}
8     b = {"d", "f", "g", "o", "v", "y"}
9     c = {"d", "e", "j", "k"}
10    d = {"a", "b", "f", "g"}
11    x = (a.intersection(b)).union(c)
12    print(f"x = {x}")
13    # Найдём дополнения множества
14    bn = u.difference(b)
15    cn = u.difference(c)
16    y = (a.difference(d)).union(cn.difference(bn))
17    print(f"y = {y}")
if __name__ == "__main__"
```

primer x

```
C:\Users\lizeq\anaconda3\envs\pythonProject2\python.exe C:/Users/lizeq/PycharmProjects/pythonProject2/primer.py
x = {'k', 'e', 'o', 'j', 'd'}
y = {'v', 'f', 'y', 'o', 'g', 'c', 'h'}
Process finished with exit code 0
```

Рисунок 1. Пример 1

3. Решил общие задания.



```
pythonProject2 C:\Users\lizeq> #!/usr/bin/env python3
1
individual_1.py 2 # -*- coding: utf-8 -*-
3
primer.py 4 if __name__ == "__main__":
5     s = str(input('Введите строку:'))
6     gl = set("ёуыаоэяиюЁУЫАОЭЯИЮ")
7     count = 0
8     for z in s:
9         if z in gl:
10            count += 1
11    print('Количество гласных букв', count)
12
```

zadanie\_1 x

```
C:\Users\lizeq\anaconda3\envs\pythonProject2\python.exe C:/Users/lizeq/PycharmProjects/pythonProject2/zadanie_1.py
Введите строку: Аааааааааа
Количество гласных букв 6
Process finished with exit code 0
```

Рисунок 2. Задание 1

```
pythonProject2 C:\Users\lizeq\ 1 #!/usr/bin/env python3
individual_1.py 2 # -*- coding: utf-8 -*-
primer.py 3
zadanie_1.py 4 if __name__ == "__main__":
zadanie_2.py 5     s1 = str(input('Введите первую строку:'))
 ernal Libraries 6     s2 = str(input('Введите вторую строку:'))
atches and Consoles 7     u = set('абвгдеёжзийклмнопрстуфхцшщъыьэяАБВГДЕЁЖЗИЙКЛМ'
8         'НОПРТСУФХЦШЩЪЫЬЭЯ')
9     d = set(s1)
10    q = set(s2)
11    d.remove(' ')
12    q.remove(' ')
13    x = (d.intersection(q))
14    print(f"x = {x}")
15

zadanie_2 x
C:\Users\lizeq\anaconda3\envs\pythonProject2\python.exe C:/Users/lizeq/PycharmProjects/pythonProject2/zadanie_2.py
Введите первую строку:мой абракадабра
Введите вторую строку:мои программирования
x = {'я', 'о', 'р', 'а', 'м', 'н'}

Process finished with exit code 0
```

Рисунок 3. Задание 2

#### 4. Выполнил индивидуальное задание.

```
pythonProject2 C:\Users\lizeq\ 1 #!/usr/bin/env python3
individual_1.py 2 # -*- coding: utf-8 -*-
primer.py 3
zadanie_1.py 4 if __name__ == "__main__":
zadanie_2.py 5     u = set("abcdefghijklmnopqrstuvwxyz")
 ernal Libraries 6     a = {"a", "h", "m", "o", "r"}
atches and Consoles 7     b = {"j", "k", "o", "u", "y"}
8     c = {"g", "h", "j"}
9     d = {"g", "j", "q"}
10    x = (a.intersection(c)).union(d.intersection(b))
11    print(f"x = {x}")
12    y = (a.intersection(b)).union(d.difference(c))
13    print(f"y = {y}")
14

individual_1 x
C:\Users\lizeq\anaconda3\envs\pythonProject2\python.exe C:/Users/lizeq/PycharmProjects/pythonProject2/individual_1.py
x = {'j', 'h'}
y = {'o', 'q'}

Process finished with exit code 0
```

Рисунок 4. Индивидуальное задание

### 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?

Перед тем как начать работу с множеством, необходимо для начала его создать. Сделать это можно, просто присвоив переменной последовательность значений, выделив их фигурными скобками.

3. Как проверить присутствие/отсутствие элемента в множестве?

Чтобы проверить, есть ли данное значение в множестве используется оператор `in`.

4. Как выполнить перебор элементов множества?

Для перебора элементов используется цикл `for 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, нужно вызывать метод `add`. Аргументом в данном случае будет добавляемый элемент последовательности.

7. Как выполнить удаление одного или всех элементов множества?

Для удаления одного элемента можно использовать функцию `remove`. А для удаления всех элементов используют метод `clear`.

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`, 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. Поскольку содержимое `frozenset` должно всегда оставаться статичным, перечень функций, с которыми такое множество может взаимодействовать, имеет ограничения.

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. Запятая в кавычках выступает в качестве символа, разделяющего значения. 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 выполнения лабораторной работы приобрел навыков по работе с множествами при написании программ с помощью языка программирования Python версии 3.x.