

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

**Отчет по лабораторной работе №2.7**

Работа с множествами в языке Python

по дисциплине «Технологии программирования и алгоритмизация»

Выполнил

студент группы ИВТ-б-о-20-1

Дыбов Д.В. « » \_\_\_\_\_ 20\_\_ г.

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

1. Создал новый репозиторий для лабораторной работы №2.7;
2. Клонировал созданный репозиторий на компьютер;
3. Создал новый PyCharm проект в папке репозитория;
4. Проработал пример:

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

if __name__ == "__main__":
    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}")

    bn = u.difference(b)
    cn = u.difference(c)

    y = (a.difference(d)).union(cn.difference(bn))
    print(f"y = {y}")
```

Рисунок 1 – Разобранный пример

5. Проверил пример на работоспособность;

```
C:\Users\Дмитрий\PycharmProjects\pythonProject1\venv\Scripts\p
x = {'d', 'j', 'o', 'k', 'e'}
y = {'y', 'c', 'o', 'h', 'v', 'g', 'f'}

Process finished with exit code 0
```

Рисунок 2 – Результат выполнения примера

6. Написал код для решения примера из индивидуального задания;

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

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

    a = {"a", "b", "h", "j", "l"}
    b = {"b", "c", "h", "l", "r", "u"}
    c = {"j", "k", "n", "t", "z"}
    d = {"b", "i", "k", "v", "w"}

    x = (a.union(b)).intersection(c)
    print(f"x = {x}")

    bA = u.difference(a)
    bB = u.difference(b)

    y = (bA.intersection(bB)).difference(c.union(d))
    print(f"y = {y}")
```

Рисунок 3 – Написанный код

7. Проверил код на работоспособность и получил результат:

```
C:\Users\Дмитрий\PycharmProjects\pythonProject1\venv\Scripts\py
x = {'j'}
y = {'m', 'p', 'o', 'd', 'q', 'x', 's', 'y', 'g', 'e', 'f'}

Process finished with exit code 0
```

Рисунок 4 – Результат выполнения индивидуального задания

8. С помощью сайта проверил пример на наличие ошибок;

9. Результат не выдал ошибок;

## Python code

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == "__main__":
5     u = set("abcdefghijklmnopqrstuvwxyz")
6
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
12    x = (a.intersection(b)).union(c)
13    print(f"x = {x}")
14
```

No syntax errors detected :)

Рисунок 5 – Проверка кода на наличие ошибок

10. Проверил индивидуальное задание на наличие ошибок, с помощью этого же сайта;

## Python code

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == "__main__":
5     u = set("abcdefghijklmnopqrstuvwxyz")
6
7     a = {"a", "b", "h", "j", "l"}
8     b = {"b", "c", "h", "l", "r", "u"}
9     c = {"j", "k", "n", "t", "z"}
10    d = {"b", "i", "k", "v", "w"}
11
12    x = (a.union(b)).intersection(c)
13    print(f"x = {x}")
14
```

No syntax errors detected :)

Рисунок 6 – Проверка индивидуального задания на наличие ошибок

11. Отправил все изменения на репозиторий;

```
C:\Program Files\Git\laba2.7>git add
Nothing specified, nothing added.
hint: Maybe you wanted to say 'git add .'?
hint: Turn this message off by running
hint: "git config advice.addEmptyPaths false"

C:\Program Files\Git\laba2.7>git add .
warning: LF will be replaced by CRLF in .idea/inspectionProfiles/profiles_settings.xml
The file will have its original line endings in your working directory

C:\Program Files\Git\laba2.7>git commit -m "Add python files"
[main d6afc02] Add python files
8 files changed, 85 insertions(+)
create mode 100644 .idea/.gitignore
create mode 100644 .idea/inspectionProfiles/profiles_settings.xml
create mode 100644 .idea/misc.xml
create mode 100644 .idea/modules.xml
create mode 100644 .idea/pythonProject1.iml
create mode 100644 Primer.py
create mode 100644 Zadanie1.py
create mode 100644 main.py

C:\Program Files\Git\laba2.7>git push
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 8 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (12/12), 2.22 KiB | 2.22 MiB/s, done.
Total 12 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/dibovdmitry/laba2.7.git
```

Рисунок 7 – Отправка изменений на репозиторий

## Контрольные вопросы

### 1. Что такое множества в языке Python?

Множеством в языке Python называется неупорядоченная совокупность уникальных значений.

### 2. Как осуществляется создание множеств в Python?

Множества в языке Python можно создать, написав фигурные скобки «{ }», наполненными числами, перечисленные с помощью запятой.

### 3. Как проверить присутствие/отсутствие элемента в множестве?

Чтобы проверить наличие элемента в множестве можно использовать команду `print` («значение, которое необходимо проверить на наличие» `in` «словарь, в котором необходимо проверить это значение»).

### 4. Как выполнить перебор элементов множества?

Чтобы выполнить перебор элементов множества необходимо написать `for «множества» in {«элементы множества»}`:

#### 5. Что такое set comprehension?

set comprehension – это генератор, который позволяет заполнять списки, а также другие наборы данных с учетом неких условий

#### 6. Как выполнить добавление элемента во множество?

Чтобы добавить элемент в множество необходимо использовать метод «add». Пример написания: `множества.add(элемент)`.

#### 7. Как выполнить удаление одного или всех элементов множества?

Чтобы удалить определённый элемент необходимо использовать метод «remove». Пример написания: `множества.remove(элемент)`

Чтобы удалить все элементы сразу, достаточно использовать метод «clear». Пример написания: `множества.clear()`

8. Как выполняются основные операции над множествами: объединение, пересечение, разность?

Чтобы выполнить объединение используется метод «union». Пример написания `«первое множество».union(«второе множество»)`

Чтобы выполнить пересечение используется метод «intersection». Пример написания `«первое множество».intersection(«второе множество»)`

Чтобы выполнить объединение используется метод «difference». Пример написания `«первое множество».difference(«второе множество»)`

9. Как определить, что некоторое множество является надмножеством или подмножеством другого множества?

Чтобы определить, является ли множество надмножеством необходимо использовать метод `issuperset`.

Чтобы определить, является ли множество подмножеством необходимо использовать метод `issubset`.

#### 10. Каково назначение множеств frozenset?

Множество с типом `frozenset` не позволяет изменять своё содержимое.

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

Чтобы преобразовать множество в строку необходимо использовать функцию `join`.

Чтобы преобразовать множество в словарь необходимо использовать вызов `dict`.

Чтобы преобразовать множество в список необходимо использовать вызов `list`.

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