

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

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №2.7
дисциплины «Основы программной инженерии»

Выполнил:
Баратов Семен Григорьевич
2 курс, группа ПИЖ-б-о-22-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

(подпись)

Преподаватель:
Воронкин Р.А., канд. тех. наук, доцент,
доцент кафедры инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: Работа с множествами в языке Python.

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

Результаты выполнения

1. Создали репозиторий с лицензией MIT, добавили в .gitignore необходимые правила для работы с IDE PyCharm, клонировали репозиторий, организовали репозиторий в соответствии с моделью git-flow.

```
Last login: Tue Oct 24 20:29:02 on ttys000
itssyoma@MacBook-Air-Sema Основы программной инженерии % git clone https://github.com/itssyoma/megarepo_21.git
Cloning into 'megarepo_21'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
itssyoma@MacBook-Air-Sema Основы программной инженерии % cd megarepo_21
itssyoma@MacBook-Air-Sema megarepo_21 % git checkout -b develop
Switched to a new branch 'develop'
itssyoma@MacBook-Air-Sema megarepo_21 % git branch release
itssyoma@MacBook-Air-Sema megarepo_21 % git branch develop
fatal: a branch named 'develop' already exists
itssyoma@MacBook-Air-Sema megarepo_21 % git branch hotfix
itssyoma@MacBook-Air-Sema megarepo_21 % git branch feature
itssyoma@MacBook-Air-Sema megarepo_21 %
```

Рисунок 1 – Работа с репозиторием в командной строке.

2. Проработали пример.

```
example.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  if __name__ == "__main__":
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     # Найдём 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
```

ПРОБЛЕМЫ	ВЫХОДНЫЕ ДАННЫЕ	КОНСОЛЬ ОТЛАДКИ	ТЕРМИНАЛ
			<pre>/usr/local/bin/python3 "/Users/itssyoma/Yandex.Disk.localiz itssyoma@MacBook-Air-Sema megarepo_27 % /usr/local/bin/pyth x = {'d', 'j', 'e', 'k', 'o'} y = {'y', 'c', 'g', 'v', 'o', 'h', 'f'}</pre>

Рисунок 2 – Пример. Код и результат выполнения программы.

3. Решили задачу №1. Подсчитали количество гласных в строке, введенной с клавиатуры с использованием множеств.

```
task1.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  if __name__ == "__main__":
6      # Ввод строки с клавиатуры
7      string = input("Введите строку: ")
8
9      # Создание множества гласных
10     vowels = {'a', 'e', 'i', 'o', 'u', 'y', 'a', 'e', 'ё', 'и', 'o', 'y', 'ы', 'э', 'ю', 'я'}
11
12     # Подсчет количества гласных в строке
13     count = 0
14     for char in string:
15         if char.lower() in vowels:
16             count += 1
17
18     print("Количество гласных в строке:", count)
19
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

```
itssyoma@MacBook-Air-Sema megarepo_27 % /usr/local/bin/python3 "/Users/itssyoma/Yandex.Disk.localized/Уче
Введите строку: привет
Количество гласных в строке: 2
itssyoma@MacBook-Air-Sema megarepo_27 % /usr/local/bin/python3 "/Users/itssyoma/Yandex.Disk.localized/Уче
Введите строку: hello
Количество гласных в строке: 2
itssyoma@MacBook-Air-Sema megarepo_27 % /usr/local/bin/python3 "/Users/itssyoma/Yandex.Disk.localized/Уче
Введите строку:
Количество гласных в строке: 0
itssyoma@MacBook-Air-Sema megarepo_27 % /usr/local/bin/python3 "/Users/itssyoma/Yandex.Disk.localized/Уче
Введите строку: ghhht
Количество гласных в строке: 0
```

Рисунок 3 – Задание №1. Код и результат выполнения программы.

4. Решили задачу №2. Определили общие символы в двух строках, введенных с клавиатуры.

```
task2.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6
7  if __name__ == "__main__":
8      # Ввод двух строк с клавиатуры
9      string1 = input("Введите первую строку: ")
10     if not(string1):
11         print("Введенная строка является пустой", file=sys.stderr)
12         exit(1)
13     string2 = input("Введите вторую строку: ")
14     if not(string2):
15         print("Введенная строка является пустой", file=sys.stderr)
16         exit(1)
17
18     # Создание множеств символов из каждой строки
19     set1 = set(string1)
20     set2 = set(string2)
21
22     # Нахождение общих символов
23     common_chars = set1.intersection(set2)
24
25     if common_chars:
26         print("Общие символы в двух строках:", common_chars)
27     else:
28         print("Общих символов не обнаружено")
29
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

```
itssyoma@MacBook-Air-Sema megarepo_27 % /usr/local/bin/python3 "/Users/itssyoma/Yandex.Disk.localized/Уче
ero_27/task2.py"
Введите первую строку:
Введенная строка является пустой
itssyoma@MacBook-Air-Sema megarepo_27 % /usr/local/bin/python3 "/Users/itssyoma/Yandex.Disk.localized/Уче
ero_27/task2.py"
Введите первую строку: абвгде
Введите вторую строку:
Введенная строка является пустой
itssyoma@MacBook-Air-Sema megarepo_27 % /usr/local/bin/python3 "/Users/itssyoma/Yandex.Disk.localized/Уче
ero_27/task2.py"
Введите первую строку: абвгде
Введите вторую строку: гдзжикл
Общие символы в двух строках: {'e', 'д', 'г'}
```

Рисунок 4 – Задание №2. Код и результат выполнения программы.

5. Выполнили индивидуальное задание (вариант 1). Определить результат выполнения операций над множествами. Считать элементы множества строками. Проверить результаты вручную.

$$A = \{b, e, f, k, t\}; \quad B = \{f, i, j, p, y\}; \quad C = \{j, k, l, y\}; \quad D = \{i, j, s, t, u, y, z\};$$

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

```

individual.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  if __name__ == "__main__":
6      alphabet = set(chr(i) for i in range(ord('a'), ord('z')))
7      A = set('befkt')
8      B = set('fijpy')
9      C = set('jkly')
10     D = set('ijstuyz')
11
12     X = A.intersection(C).union(B.intersection(C))
13     Y = A.difference(B).union(D.difference(C))
14
15     print('Множество X: ', X, '\nМножество Y: ', Y)
16

```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

```

/usr/local/bin/python3 "/Users/itssyoma/Yandex.Disk.localized/Учеба/Осн
itssyoma@MacBook-Air-Sema megarepo_27 % /usr/local/bin/python3 "/Users,
epo_27/individual.py"
Множество X: {'j', 'y', 'k'}
Множество Y: {'u', 'b', 'z', 't', 's', 'i', 'e', 'k'}

```

Рисунок 5 – Индивидуальное задание.

Код и результат выполнения программы.

Ручная проверка результатов:

$$X = (A \cap C) \cup (B \cap C)$$

$$A \cap C = \{b, e, f, k, t\} \cap \{j, k, l, y\} = \{k\}$$

$$B \cap C = \{f, i, j, p, y\} \cap \{j, k, l, y\} = \{j, y\}$$

$$X = \{k\} \cup \{j, y\} = \{k, j, y\} - \text{верно}$$

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

$$(A \cap \bar{B}) = (A / B) = \{b, e, f, k, t\} / \{f, i, j, p, y\} = \{b, e, k, t\}$$

$$(D / C) = \{i, j, s, t, u, y, z\} / \{j, k, l, y\} = \{i, s, t, u, z\}$$

$$Y = \{b, e, k, t\} \cup \{i, s, t, u, z\} = \{b, e, i, k, s, t, u, z\} - \text{верно}$$

Ответы на контрольные вопросы

1. Что такое множества в языке Python?

Множество в языке Python - это неупорядоченная коллекция уникальных элементов.

2. Как осуществляется создание множеств в Python?

Множество создается с помощью фигурных скобок или функции `set()`.

3. Как проверить присутствие/отсутствие элемента в множестве?

Для проверки присутствия/отсутствия элемента в множестве используются операторы `in` и `not in`.

4. Как выполнить перебор элементов множества?

Перебор элементов множества можно осуществить с помощью цикла `for`.

5. Что такое `set comprehension`?

`Set comprehension` - это способ создания множества на основе итерации по другой последовательности.

6. Как выполнить добавление элемента во множество?

Элемент добавляется во множество с помощью метода `add()`.

7. Как выполнить удаление одного или всех элементов множества?

Удаление одного или всех элементов множества выполняется с помощью методов `remove()`, `discard()` и `clear()`.

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

Операции над множествами (объединение, пересечение, разность) выполняются с помощью соответствующих операторов или методов: `|`, `&`, `-`, `union()`, `intersection()`, `difference()`.

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

Чтобы определить, что некоторое множество является надмножеством или подмножеством другого множества, используются операторы `<=` (`subset`) и `>=` (`superset`).

10. Каково назначение множеств frozenset ?

Множества frozenset - это неизменяемые множества, которые могут быть использованы в качестве ключей словаря или элементов других множеств.

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

Преобразование множеств в строку, список, словарь осуществляется с помощью функций str(), list(), dict().