

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 «Программирование на Python»

Выполнил:
Дзуев Альберт Мухаметович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

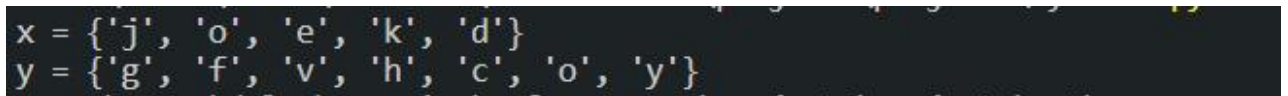
Тема: Работа с множествами в языке Python

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

Порядок выполнения работы:

1. Создал новый репозиторий, клонировал его, в нем создал ветку developer и перешел на нее.

2. Проработал пример лабораторной работы:



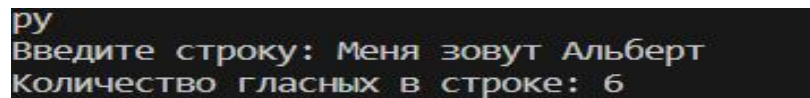
```
x = {'j', 'o', 'e', 'k', 'd'}
y = {'g', 'f', 'v', 'h', 'c', 'o', 'y'}
```

Рисунок 1. Запуск программы примера

3. Решил задачу 1: подсчитайте количество гласных в строке, введенной с клавиатуры с использованием множеств.

Код программы:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ == "__main__":
    vowels_all = {'a', 'e', 'i', 'o', 'u', 'y', 'a', 'e', 'ё', 'и', 'o', 'y', 'ы', 'э', 'ю', 'я'}
    string = input("Введите строку: ")
    count = len([i for i in string.lower() if i in vowels_all])
    print("Количество гласных в строке:", count)
```



```
ру
Введите строку: Меня зовут Альберт
Количество гласных в строке: 6
```

Рисунок 2. Вывод программы task_1

4. Решил задачу 2: определите общие символы в двух строках, введенных с клавиатуры.

Код программы:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if
name == " main ":
    string1 = input("Введите первую строку: ")
    string2 = input("Введите вторую строку: ")
    set1 = set(string1)
    set2 = set(string2)
    common_characters = set1.intersection(set2)
    print("Общие символы:", common_characters)
```

```

ру
Введите первую строку: Привет, меня зовут Альберт
Введите вторую строку: Я делаю лабораторную работу
Общие символы: {'т', 'о', 'р', 'б', 'л', 'н', ' ', 'у', 'е'}

```

Рисунок 3. Вывод программы task_2

5. Выполнил индивидуальное задание вариант 9:

$A = \{a, e, f, i\}; \quad B = \{a, b, k, n\}; \quad C = \{e, f, n, o, w, x\}; \quad D = \{a, d, e, o, p, t, u\};$ $X = (A \cup B) \cap D; \quad Y = (\bar{A} \cap \bar{B}) / (C \cup D).$
--

Рисунок 4. Индивидуальное задание вариант 9

Код программы:

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ == "__main__":
# Определим универсальное множество
u = set("abcdefghijklmnopqrstuvwxyz")
a = {"a", "e", "f", "i"}
b = {"a", "b", "k", "n"}
c = {"e", "f", "n", "o", "w", "x"}
d = {"a", "d", "e", "o", "p", "t", "u"}
x = (a.union(b)).intersection(d)
print(f'x = {x}')
# дополнения множеств
an = u.difference(a)
bn = u.difference(b)
y = (an.intersection(bn)).difference(c.union(d))
print(f'y = {y}')

```

```

x = {'a', 'e'}
y = {'h', 'm', 'j', 'y', 'z', 'v', 'q', 's', 'g', 'c', 'r', 'l'}PS C:\Use

```

Рисунок 4. Запуск программы индивидуального задания

Ответы на контрольные вопросы:

1. Что такое множества в языке Python?

Множеством в языке программирования Python называется неупорядоченная совокупность уникальных значений.

2. Как осуществляется создание множеств в Python?

Создать множество можно, просто присвоив переменной последовательность значений, выделив их фигурными скобками или с помощью метода `set`.

3. Как проверить присутствие/отсутствие элемента в множестве?

Для проверки присутствия используется `in`, для проверки отсутствия `not in`.

4. Как выполнить перебор элементов множества?

`for a in set: print(a)`.

5. Что такое `set comprehension`?

Генератор, позволяющий заполнять списки, а также другие наборы данных с учетом неких условий.

6. Как выполнить добавление элемента во множество?

Чтобы внести новые значения, потребуется вызывать метод `add`. Аргументом в данном случае будет добавляемый элемент последовательности.

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.

Иногда необходимо полностью убрать все элементы. Чтобы не удалять каждый элемент отдельно, используется метод `clear()`, не принимающий аргументов.

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.

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 результате выполнения работы были приобретены навыки по работе с множествами при написании программ с помощью языка программирования Python версии 3.x.