

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

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЁТ

по лабораторной работе №2.7

Дисциплина: «Программирование на Python»
Тема: «Работа с множествами в языке Python»

Выполнил:
Епифанов Алексей Александрович
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 Роман Александрович

(подпись)

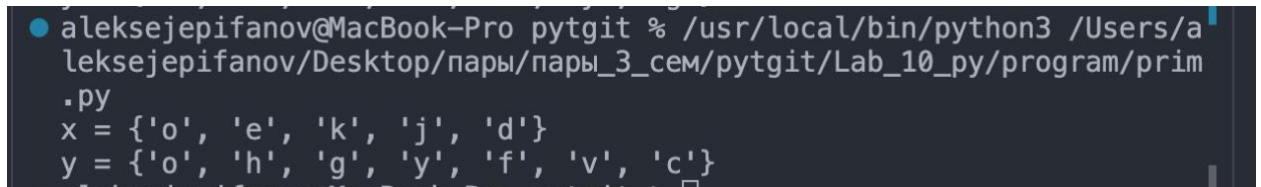
Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

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:

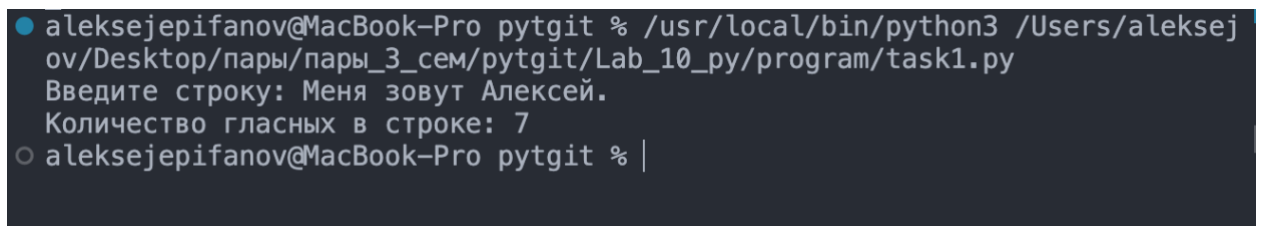


```
aleksejepifanov@MacBook-Pro pytgit % /usr/local/bin/python3 /Users/aleksejepifanov/Desktop/пары/пары_3_сем/pytgit/Lab_10_py/program/prim.py
x = {'o', 'e', 'k', 'j', 'd'}
y = {'o', 'h', 'g', 'y', 'f', 'v', 'c'}
```

Рисунок 1. Запуск программы примера

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

```
#!/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 if i in vowels_all])
    print("Количество гласных в строке:", count)
```



```
aleksejepifanov@MacBook-Pro pytgit % /usr/local/bin/python3 /Users/aleksejepifanov/Desktop/пары/пары_3_сем/pytgit/Lab_10_py/program/task1.py
Введите строку: Меня зовут Алексей.
Количество гласных в строке: 7
aleksejepifanov@MacBook-Pro pytgit % |
```

Рисунок 2. Вывод программы task_1

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

```
#!/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)
```

```
● aleksejepifanov@MacBook-Pro pytgit % /usr/local/bin/python3 /Users/aleksej
epifanov/Desktop/пары/пары_3_сем/pytgit/Lab_10_py/program/task2.py
Введите первую строку: Привет, это Алексей.
Введите вторую строку: Пока, Алексей уходит.
Общие символы: {'к', 'т', 'с', 'е', 'л', ' ', ',', 'А', 'П', 'и', 'о', '.',
, 'й'}
● aleksejepifanov@MacBook-Pro pytgit % /usr/local/bin/python3 /Users/aleksej
epifanov/Desktop/пары/пары_3_сем/pytgit/Lab_10_py/program/task2.py
Введите первую строку: Привет друг.
Введите вторую строку: Пока человек.
Общие символы: {'е', ' ', 'П', 'в', '.'}
○ aleksejepifanov@MacBook-Pro pytgit % |
```

Рисунок 3. Вывод программы task_2

5. Выполнил индивидуальное задание вариант 10:

$$A = \{a, b, h, k, o, r\}; \quad B = \{b, g, h, l, s\}; \quad C = \{k, l, z\}; \quad D = \{g, j, p, q, u, v\};$$

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

Рисунок 4. Индивидуальное задание вариант 10

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ == "__main__":
    # Определим универсальное множество
    u = set("abcdefghijklmnopqrstuvwxyz")
    a = {"a", "b", "h", "k", "o", "r"}
    b = {"b", "g", "h", "l", "s"}
    c = {"k", "l", "z"}
    d = {"g", "j", "p", "q", "u", "v"}
    x = (a.intersection(c)).union(b)
    print(f"x = {x}")
    # Найдём дополнения множеств
    an = u.difference(a)
    bn = u.difference(b)
    y = (an.intersection(bn)).difference(c.union(d))
    print(f"y = {y}")
```

```
● aleksejepifanov@MacBook-Pro pytgit % /usr/local/bin/python3 /Users/aleksej
epifanov/Desktop/пары/пары_3_сем/pytgit/Lab_10_py/program/ind.py
x = {'k', 'h', 'l', 'b', 's', 'g'}
y = {'w', 't', 'i', 'c', 'd', 'x', 'y', 'f', 'm', 'e', 'n'}
○ aleksejepifanov@MacBook-Pro pytgit %
```

Рисунок 4. Запуск программы индивидуального задания

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.