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 №10**

**дисциплины «Программирование на Python»**

**Вариант 23**

|  |  |  |
| --- | --- | --- |
|  | Выполнил:  Омонкулов Исомиддин Валижон угли  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.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_ Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ставрополь, 2024

**Тема:** Работа с множествами в языке Python

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

**Пример 1.** Определить результат выполнения операций над множествами. Считать элементы множества строками.

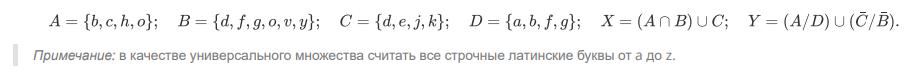


Рисунок 1. Содержание задачи

**Листинг программы:**

#!/usr/bin/env python3  
# -\*- coding: utf-8 -\*-  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 # Определим универсальное множество  
 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}")  
  
 # Найдем дополнения множеств  
 bn = u.difference(b)  
 cn = u.difference(c)  
  
 y = (a.difference(d)).union(cn.difference(bn))  
 print(f"y = {y}")

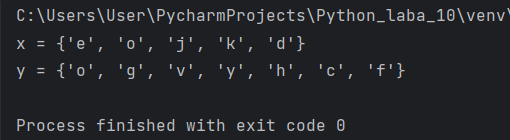


Рисунок 2. Результат выполнения программы

**Задание 1.** Подсчитайте количество гласных в строке, введенной с клавиатуры с использованием множеств.

**Листинг программы:**

#!/usr/bin/env python3  
# -\*- coding: utf-8 -\*-  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 glassnie = {'a', 'e', 'i', 'o', 'u', 'y', 'а', 'е', 'ё', 'и', 'о', 'у', 'ы', 'э', 'ю', 'я'}  
  
 s = input("Введите строку: ")  
  
 counter = len([i for i in s.lower() if i in glassnie])  
  
 print("Количество гласных в строке:", counter)

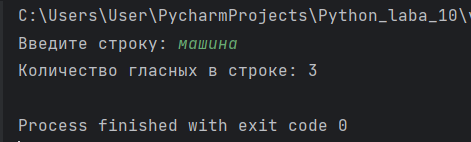


Рисунок 3. Тест №1

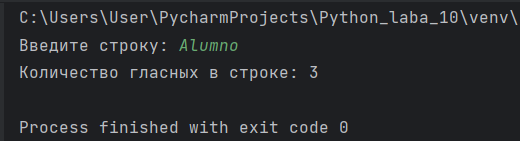


Рисунок 4. Тест №2

**Задание 2.** Определите общие символы в двух строках, введенных с клавиатуры.

**Листинг программы:**

#!/usr/bin/env python3  
# -\*- coding: utf-8 -\*-  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 s\_1 = input("Enter the first line: ")  
 s\_2 = input("Enter the second line: ")  
  
 set1 = set(s\_1)  
 set2 = set(s\_2)  
  
 common\_characters = set1.intersection(set2)  
  
 print("Common characters:", common\_characters)

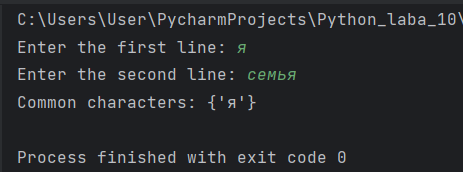


Рисунок 5. Тест №1

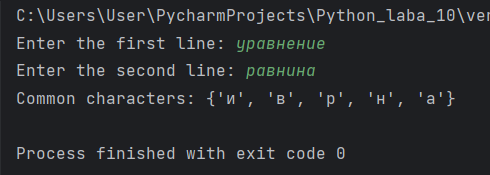


Рисунок 6. Тест №2

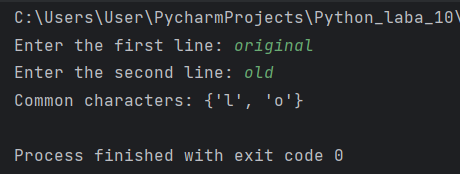


Рисунок 7. Тест №3

**Индивидуальное задание.** Определить результат выполнения операций над множествами. Считать элементы множества строками.



Рисунок 8. Содержание индивидуального задания

**Листинг программы:**

#!/usr/bin/env python3  
# -\*- coding: utf-8 -\*-  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 # Определим универсальное множество  
 u = set("abcdefghijklmnopqrstuvwxyz")  
  
 a = {"b", "f", "g", "m", "o"}  
 b = {"b", "g", "h", "l", "u"}  
 c = {"e", "f", "m"}  
 d = {"e", "g", "l", "p", "q", "u", "v"}  
  
 x = (a.intersection(c)).union(b)  
 print(f"x = {x}")  
  
 # Найдём дополнения множества  
 bn = u.difference(b)  
  
 y = (a.intersection(bn)).union(c.difference(d))  
 print(f"y = {y}")

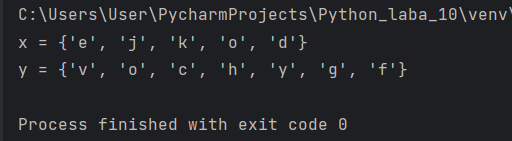


Рисунок 9. Результат программы

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

**Ответы на контрольные вопросы**

1. Множество в языке Python - это неупорядоченная коллекция уникальных элементов.

2. Множество можно создать, перечислив элементы в фигурных скобках, разделенных запятыми, например: my\_set = 1, 2, 3.

3. Для проверки присутствия/отсутствия элемента в множестве используется оператор in, например: if element in my\_set:.

4. Перебор элементов множества можно осуществить с помощью цикла for, например: for element in my\_set:.

5. Set comprehension - это способ создания множества на основе итерации, аналогичный списковому включению, например: my\_set = x for x in range(10).

6. Добавление элемента во множество осуществляется с помощью метода add(), например: my\_set.add(4).

7. Удаление одного или всех элементов множества можно выполнить с помощью методов remove() и clear() соответственно.

8. Основные операции над множествами в Python:

- Объединение

- Пересечение

- Разность

9. Для определения надмножества или подмножества используются операторы <= (является подмножеством) и >= (является надмножеством).

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

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