

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 кроссплатформенного программирования»

Выполнила:
Мурашко Анастасия Юрьевна
1 курс, группа ИТС-б-о-22-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети», 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 со словарями при написании программ с помощью языка программирования Python версии 3.x.

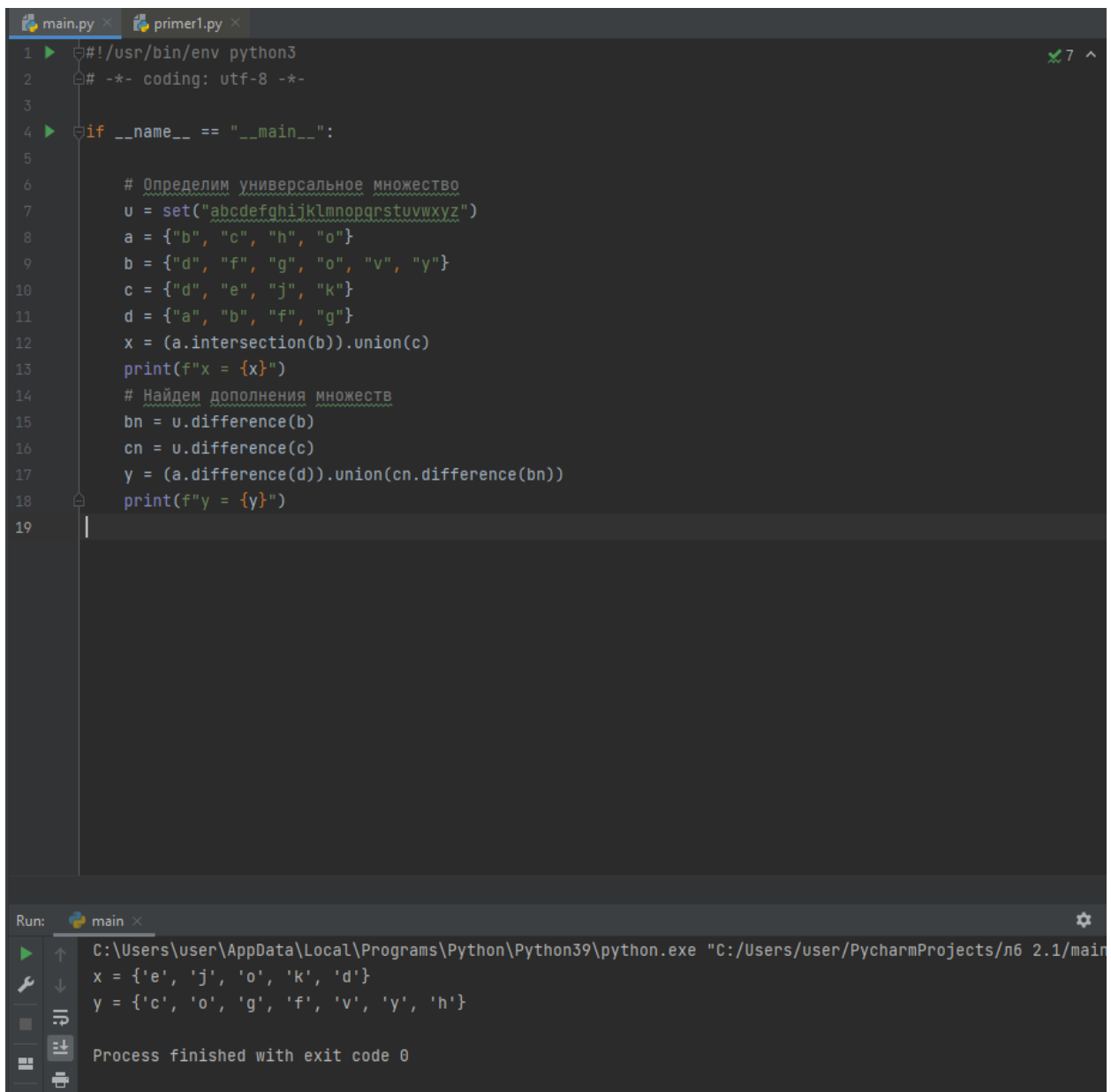
Ход работы:

Вариант №10

Пример 1

Определить результат выполнения операций над множествами.
Считать элементы множества строками.

Рисунок 1. Пример 1



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == "__main__":
5
6      # Определим универсальное множество
7      u = set("abcdefghijklmnopqrstuvwxyz")
8      a = {"b", "c", "h", "o"}
9      b = {"d", "f", "g", "o", "v", "y"}
10     c = {"d", "e", "j", "k"}
11     d = {"a", "b", "f", "g"}
12     x = (a.intersection(b)).union(c)
13     print(f"x = {x}")
14     # Найдем дополнения множеств
15     bn = u.difference(b)
16     cn = u.difference(c)
17     y = (a.difference(d)).union(cn.difference(bn))
18     print(f"y = {y}")
19
```

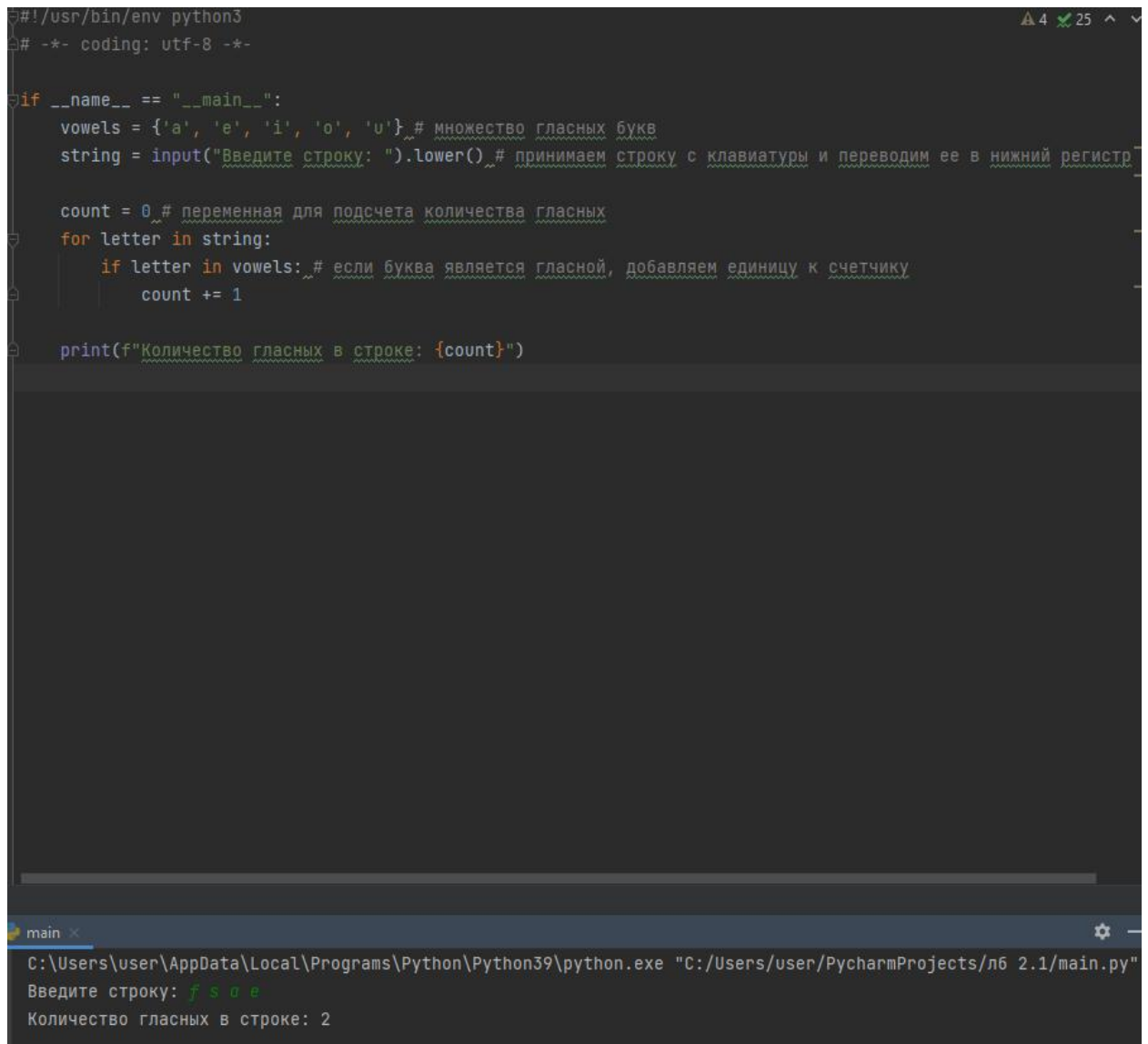
Run: main

```
C:\Users\user\AppData\Local\Programs\Python\Python39\python.exe "C:/Users/user/PycharmProjects/n6 2.1/main"
x = {'e', 'j', 'o', 'k', 'd'}
y = {'c', 'o', 'g', 'f', 'v', 'y', 'h'}

Process finished with exit code 0
```

Общее задание 1

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



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

if __name__ == "__main__":
    vowels = {'a', 'e', 'i', 'o', 'u'} # множество гласных букв
    string = input("Введите строку: ").lower() # принимаем строку с клавиатуры и переводим ее в нижний регистр

    count = 0 # переменная для подсчета количества гласных
    for letter in string:
        if letter in vowels: # если буква является гласной, добавляем единицу к счетчику
            count += 1

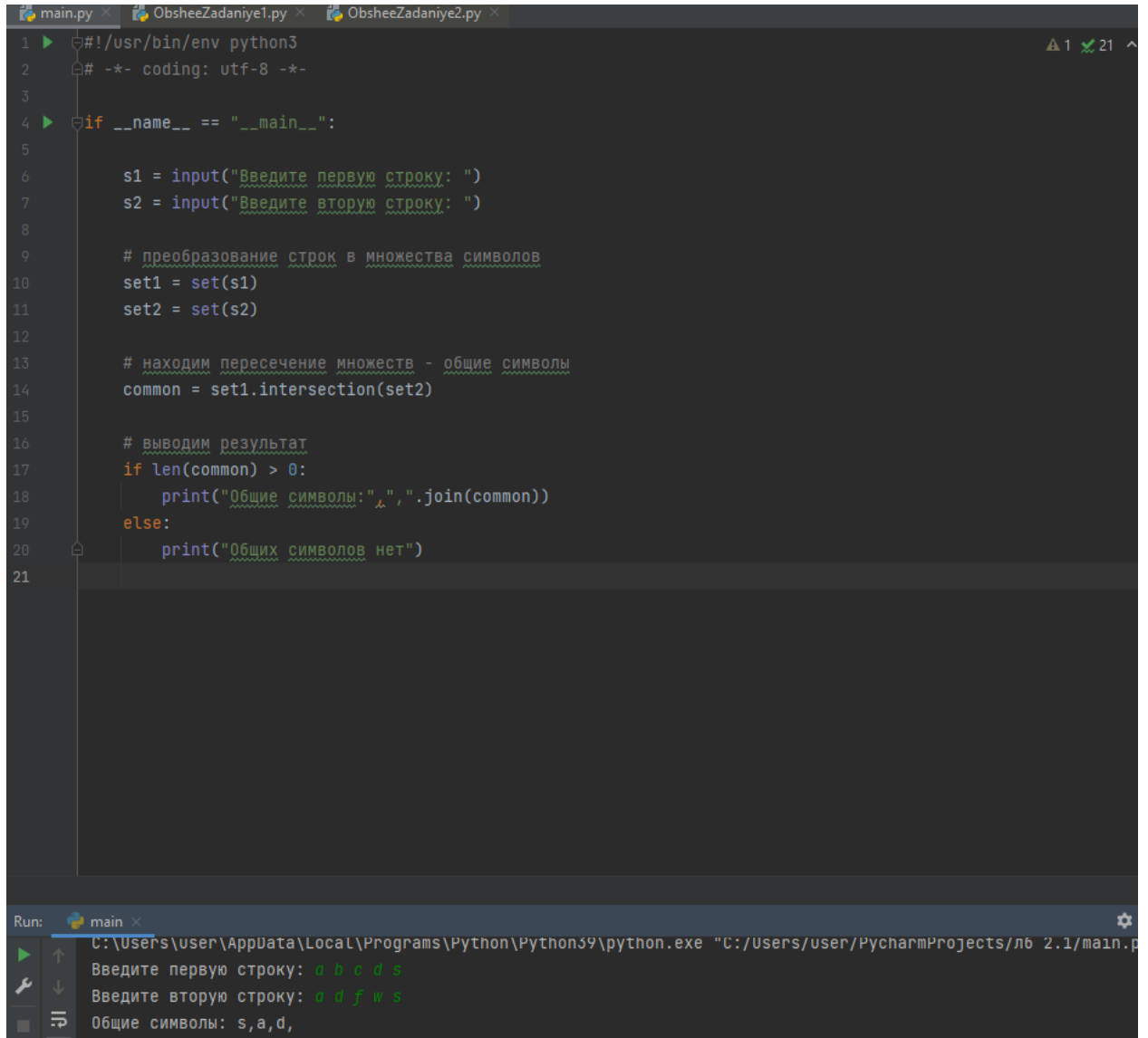
    print(f"Количество гласных в строке: {count}")
```

The screenshot shows a Python IDE with a file named 'main.py'. The code defines a set of vowels and iterates through a user-input string to count them. The terminal output shows the input 'f s a e' and the resulting count of 2.

Рисунок 2. Общее задание 1

Общее задание 2

Определите общие символы в двух строках, введенных с клавиатуры.



```
1  #!/usr/bin/env python3
2  #- coding: utf-8 -*-
3
4  if __name__ == "__main__":
5
6      s1 = input("Введите первую строку: ")
7      s2 = input("Введите вторую строку: ")
8
9      # преобразование строк в множества символов
10     set1 = set(s1)
11     set2 = set(s2)
12
13     # находим пересечение множеств - общие символы
14     common = set1.intersection(set2)
15
16     # выводим результат
17     if len(common) > 0:
18         print("Общие символы: " + ",".join(common))
19     else:
20         print("Общих символов нет")
21
```

Run: main ×

C:\Users\user\AppData\Local\Programs\Python\Python39\python.exe "C:/Users/user/PycharmProjects/lb 2.1/main.p

Введите первую строку: a b c d s

Введите вторую строку: a d f k s

Общие символы: s,a,d,

Рисунок 3. Общее задание 2

Индивидуальное задание

Условия примера:

10.

$$X = (A \cap C) \cup B; \quad Y = (\bar{A} \cap \bar{B}) / (C \cup D).$$
$$A = \{a, h, k\}; \quad B = \{c, d, h, p, r\}; \quad C = \{h, i, s\}; \quad D = \{c, g, j, v, w\};$$

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ == "__main__":
    # Определим универсальное множество
    u = set("abcdefghijklmnopqrstuvwxyz")
    a = {"a", "h", "k"}
    b = {"c", "d", "h", "p", "r"}
    c = {"h", "i", "s"}
    d = {"c", "g", "j", "v", "w"}
    x = (a.intersection(c)).union(b)
    print(f"x = {x}")

    # Найдем дополнения множеств
    an = u.difference(a)
    bn = u.difference(b)
    y = (an.intersection(bn)).difference(c.union(d))
    print(f"y = {y}")
```

if __name__ == "__main__"

main x

C:\Users\user\AppData\Local\Programs\Python\Python39\python.exe "C:/Users/user/PycharmProjects/

x = {'h', 'c', 'p', 'r', 'd'}

y = {'x', 'u', 'q', 'm', 'f', 'l', 't', 'e', 'y', 'n', 'z', 'o', 'b'}

Рисунок 4. Задание 1

Ответы на контрольные вопросы:

1) Что такое множества в языке Python?

Множеством в языке программирования Python называется неупорядоченная совокупность уникальных значений. В качестве элементов этого набора данных могут выступать любые неизменяемые объекты, такие как числа, символы, строки. В отличие от массивов и списков, порядок следования значений не учитывается при обработке его содержимого. Над

одним, а также несколькими множествами можно выполнять ряд операций, благодаря функциям стандартной библиотеки языка программирования Python.

2) Как осуществляется создание множеств в Python?

Перед тем как начать работу с множеством, необходимо для начала его создать. Сделать это можно, просто присвоив переменной последовательность значений, выделив их фигурными скобками. Следующий пример показывает код, в котором создается множество целых чисел под названием `a`, после функция `print` выводит на экран его содержимое. Как можно заметить, все элементы полученной последовательности являются уникальными, без повторений. Существует и другой способ создания множеств, который подразумевает использование вызова `set`. Аргументом этой функции может быть набор неких данных или даже строка с текстом, как это показано в следующем примере.

3) Как проверить присутствие/отсутствие элемента в множестве?

Для проверки присутствия/отсутствия элемента в множестве в Python используется ключевое слово `in`.

4) Как выполнить перебор элементов множества?

Перебор элементов множества можно выполнить с помощью цикла `for`. Вот пример кода, который позволяет перебрать все элементы множества и вывести их на экран.

5) Что такое set comprehension?

Set comprehension (генератор множеств) - это способ создания и заполнения множества в Python одной строки кода. Он похож на список и словарь comprehensions, но вместо списка или словаря, генератор множества создает множество.

6) Как выполнить добавление элемента во множество?

Для добавления элемента во множество в Python используется метод `add()`.

7) Как выполнить удаление одного или всех элементов множества?

Для удаления одного элемента из множества в Python используется метод `remove()` или `discard()`. Разница между ними заключается в том, что если попытаться удалить несуществующий элемент соответствующим методом, то `remove()` выбросит ошибку `KeyError`, а `discard()` ничего не сделает. Для удаления всех элементов множества можно использовать метод `clear()`.

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

Основные операции над множествами в Python: - Объединение (`union`) - создает новое множество, содержащее все элементы из двух или более множеств. - Пересечение (`intersection`) - создает новое множество, содержащее только те элементы, которые присутствуют в обоих множествах. - Разность (`difference`) - создает новое множество, состоящее из элементов первого множества, которые не присутствуют во втором множестве.

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

Для определения, является ли некоторое множество подмножеством или надмножеством другого множества в Python, можно воспользоваться методами `issubset()` и `issuperset()` соответственно. Метод `issubset()` проверяет, является ли одно множество подмножеством другого. Если все элементы первого множества присутствуют во втором множестве, то метод вернет `True`, иначе `False`. Метод `issuperset()` проверяет, является ли одно множество надмножеством другого. Если все элементы второго множества присутствуют в первом множестве, то метод вернет `True`, иначе `False`.

10) Каково назначение множеств `frozenset` ?

frozenset - это неизменяемый тип данных в Python, который является аналогом множества (set). Его элементы не могут быть изменены после создания frozenset. Множества frozenset используются там, где требуется неизменяемость данных, например, в качестве ключей для других множеств и словарей. За счет неизменяемости frozenset в него нельзя добавлять, удалять или изменять элементы. Поэтому frozenset не имеет методов, связанных с изменением данных.

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

Для преобразования множества в строку, список или словарь в языке Python можно использовать следующие функции: - str() - для преобразования множества в строку - list() - для преобразования множества в список - dict.fromkeys() - для преобразования множества в словарь

Вывод: В ходе выполнения лабораторной работы приобретены навыки по работе со словарями при написании программ с помощью языка программирования Python версии 3.x.