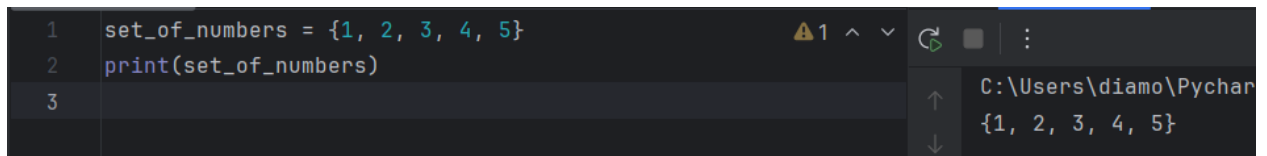


Лабораторная работа № 2. Задачи.

Комплект 1: Множества

1.1: Создание множества. Создайте множество, состоящее из любых 5ти чисел. Выведите его на экран.

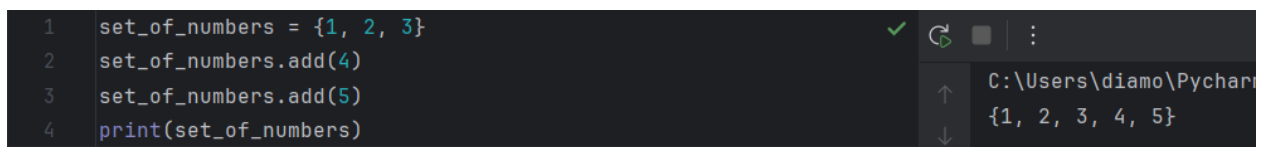
```
1 set_of_numbers = {1, 2, 3, 4, 5}
2 print(set_of_numbers)
3
```

The screenshot shows a code editor with a dark theme. The code consists of two lines: a set creation and a print statement. The output window on the right shows the set {1, 2, 3, 4, 5}.

C:\Users\diamo\Pychar
{1, 2, 3, 4, 5}

1.2: Добавление элементов в множество. Создайте множество, состоящее из 3х чисел. Добавьте в него еще два числа. Выведите конечное множество на экран.

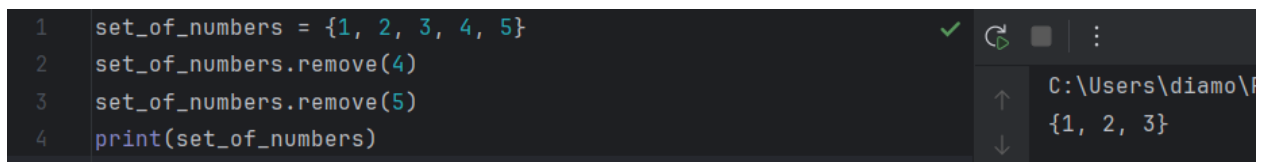
```
1 set_of_numbers = {1, 2, 3}
2 set_of_numbers.add(4)
3 set_of_numbers.add(5)
4 print(set_of_numbers)
```

The screenshot shows a code editor with four lines of code. The first line creates a set with three elements. The next two lines use the add method to add two more elements. The output window shows the resulting set {1, 2, 3, 4, 5}.

C:\Users\diamo\Pychar
{1, 2, 3, 4, 5}

1.3: Удаление элементов из множества. Создайте множество из 5 чисел. Используя метод remove, удалите два элемента. Выведите конечное множество на экран.

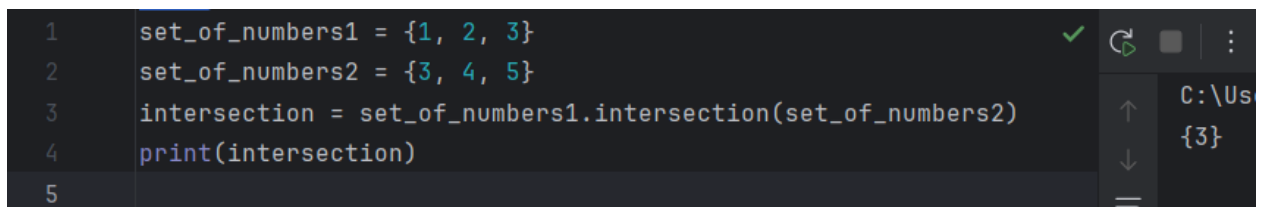
```
1 set_of_numbers = {1, 2, 3, 4, 5}
2 set_of_numbers.remove(4)
3 set_of_numbers.remove(5)
4 print(set_of_numbers)
```

The screenshot shows a code editor with four lines of code. The first line creates a set with five elements. The next two lines use the remove method to remove two elements. The output window shows the resulting set {1, 2, 3}.

C:\Users\diamo\Pychar
{1, 2, 3}

1.4: Пересечение множеств. Создайте два множества по 3 числа в каждом. Найдите их пересечение и выведите результат на экран.

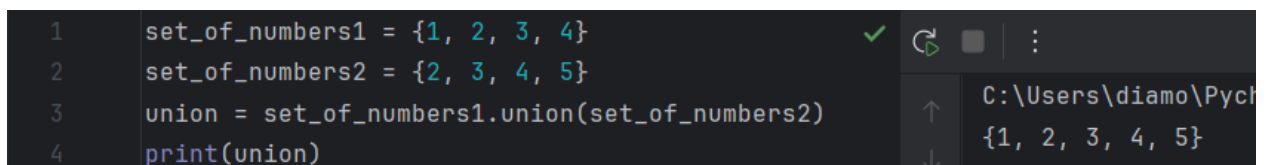
```
1 set_of_numbers1 = {1, 2, 3}
2 set_of_numbers2 = {3, 4, 5}
3 intersection = set_of_numbers1.intersection(set_of_numbers2)
4 print(intersection)
5
```

The screenshot shows a code editor with five lines of code. The first two lines create two sets. The third line uses the intersection method to find their common elements. The output window shows the result {3}.

C:\Us
{3}

1.5: Объединение множеств. Создайте два множества по 4 числа в каждом. Найдите их объединение и выведите результат на экран.

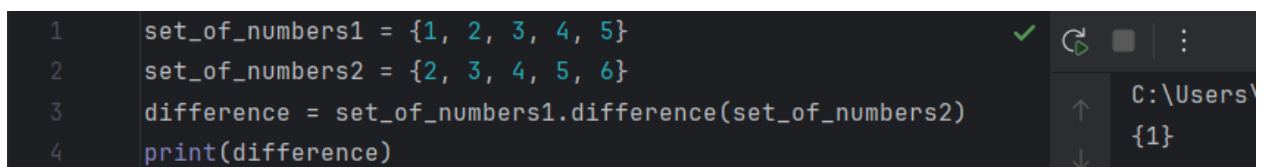
```
1 set_of_numbers1 = {1, 2, 3, 4}
2 set_of_numbers2 = {2, 3, 4, 5}
3 union = set_of_numbers1.union(set_of_numbers2)
4 print(union)
```

The screenshot shows a code editor with four lines of code. The first two lines create two sets. The third line uses the union method to combine them. The output window shows the resulting set {1, 2, 3, 4, 5}.

C:\Users\diamo\Pychar
{1, 2, 3, 4, 5}

1.6: Разность множеств. Создайте два множества по 5 чисел в каждом. Найдите разность первого и второго множества и выводите результат на экран.

```
1 set_of_numbers1 = {1, 2, 3, 4, 5}
2 set_of_numbers2 = {2, 3, 4, 5, 6}
3 difference = set_of_numbers1.difference(set_of_numbers2)
4 print(difference)
```

The screenshot shows a code editor with four lines of code. The first two lines create two sets. The third line uses the difference method to find elements in the first set that are not in the second. The output window shows the result {1}.

C:\Users\
{1}

1.7: Симметричная разность множеств. Создайте два множества по 3 числа в каждом. Определите их симметричную разность и выведите результат на экран.

```

1 set_of_numbers1 = {1, 2, 3}
2 set_of_numbers2 = {3, 4, 5}
3 symmetric_difference = set_of_numbers1.symmetric_difference(set_of_numbers2)
4 print(symmetric_difference)

```

C:\Users\diam...
{1, 2, 4, 5}

1.8: Проверка на вхождение элемента в множество. Создайте множество из 5 чисел. Проверьте, присутствует ли число 3 в вашем множестве и выводите результат на экран.

```

1 set_of_numbers = {1, 2, 3, 4, 5}
2 if 3 in set_of_numbers:
3     print('Число 3 присутствует в множестве')
4 else:
5     print('Число 3 отсутствует в множестве')

```

C:\Users\diamo\PycharmProjects\pythonf...
Число 3 присутствует в множестве
Process finished with exit code 0

```

1 set_of_numbers = {1, 2, 3, 4, 5}
2 n = int(input())
3 if n in set_of_numbers:
4     print('Число присутствует в множестве')
5 else:
6     print('Число отсутствует в множестве')
7

```

C:\Users\diamo\PycharmProjects\pyt...
3
Число присутствует в множестве
Process finished with exit code 0

1.9: Подмножества и надмножества. Создайте два множества, одно из которых будет являться подмножеством другого. Проверьте данное утверждение с использованием соответствующего метода и выведите результат на экран.

```

1 set_of_numbers1 = {2, 3, 4, 5}
2 set_of_numbers2 = {1, 2, 3, 4, 5}
3 if set_of_numbers1.issubset(set_of_numbers2):
4     print('Первое множество является подмножеством второго')
5 else:
6     print('Первое множество не является подмножеством второго')

```

C:\Users\diamo\PycharmProjects\pythonProject2\venv...
Первое множество является подмножеством второго
Process finished with exit code 0

1.10: Применение множеств в задаче. Дан список чисел с повторяющимися элементами. Необходимо с помощью множества убрать повторяющиеся элементы из списка и вывести итоговый список на экран.

```

1 list_of_numbers = [1, 2, 2, 3, 4, 5, 5]
2 unique_numbers = list(set(list_of_numbers))
3 print(unique_numbers)
4

```

C:\Users\diamo\PycharmProjects...
[1, 2, 3, 4, 5]