

## Рубежный контроль № 2

**Задание**

Что необходимо сделать, чтобы методы работы с множествами в LINQ работали с объектами классов?

**Ответ**

Операции над множествами в LINQ — это операции запросов, результирующие наборы которых основываются на наличии или отсутствии эквивалентных элементов в одной или другой коллекции (или наборе).

Операции над множествами в LINQ возможны при использовании методов:

Имя метода	Описание метода
Distinct	Удаляет повторяющиеся значения из коллекции
Except	Исключение. Возвращает разность множеств, т. е. элементы одной коллекции, которые отсутствуют во второй.
Intersect	Перечесение. Возвращает пересечение множеств, т. е. элементы, присутствующие в каждой из двух коллекций.
Union	Объединение. Возвращает объединение множеств, т. е. уникальные элементы, присутствующие в одной из двух коллекций.

Для использования вышеперечисленных методов с пользовательскими объектам классов, необходимо поместить объекты в последовательности, реализующие интерфейс `IEnumerable<T>`. Приведём пример, демонстрирующий работу методов с последовательностями пользовательских объектов.

```

class Program
{
    Ссылка: 6
    static void OutputCollection(IEnumerable<Rectangle> col)
    {
        foreach (Rectangle r in col)
        {
            Console.WriteLine(r);
        }
    }

    Ссылка: 0
    static void Main(string[] args)
    {
        Rectangle rt1 = new Rectangle(2, 2);
        Rectangle rt2 = new Rectangle(3, 3);
        Rectangle rt3 = new Rectangle(4, 4);
        Rectangle rt4 = new Rectangle(5, 5);

        List<Rectangle> lst1 = new List<Rectangle> { rt1, rt2, rt3, rt1 };
        List<Rectangle> lst2 = new List<Rectangle> { rt3, rt4, rt1, rt4 };

        Console.WriteLine("Последовательность №1: ");
        OutputCollection(lst1);

        Console.WriteLine("\nПоследовательность №2: ");
        OutputCollection(lst2);

        Console.WriteLine("\nРезультат работы метода DISTINCT для последовательности №1:");
        IEnumerable<Rectangle> lst3 = from l:Rectangle in lst1.Distinct() select l;
        OutputCollection(lst3);

        Console.WriteLine("\nИсключение (except) из последовательности №1 последовательности №2:");
        lst3 = from l:Rectangle in lst1.Except(lst2) select l;
        OutputCollection(lst3);

        Console.WriteLine("\nПересечение (intersect) последовательностей №1 и №2:");
        lst3 = from l:Rectangle in lst1.Intersect(lst2) select l;
        OutputCollection(lst3);

        Console.WriteLine("\nОбъединение (union) последовательностей №1 и №2:");
        lst3 = from l:Rectangle in lst1.Union(lst2) select l;
        OutputCollection(lst3);
    }
}

```

Результат работы:

Последовательность №1:

Прямоугольник с площадью: 4  
Прямоугольник с площадью: 9  
Прямоугольник с площадью: 16  
Прямоугольник с площадью: 4

Последовательность №2:

Прямоугольник с площадью: 16  
Прямоугольник с площадью: 25  
Прямоугольник с площадью: 4  
Прямоугольник с площадью: 25

Результат работы метода DISTINCT для последовательности №1:

Прямоугольник с площадью: 4  
Прямоугольник с площадью: 9  
Прямоугольник с площадью: 16

Исключение (except) из последовательности №1 последовательности №2:

Прямоугольник с площадью: 9

Пересечение (intersect) последовательностей №1 и №2:

Прямоугольник с площадью: 4  
Прямоугольник с площадью: 16

Объединение (union) последовательностей №1 и №2:

Прямоугольник с площадью: 4  
Прямоугольник с площадью: 9  
Прямоугольник с площадью: 16  
Прямоугольник с площадью: 25