Кроме методов выборки LINQ имеет несколько методов для работы с множествами: разность, объединение и пересечение.

**Разность множеств**

С помощью метода **Except** можно получить разность двух множеств:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | string[] soft = { "Microsoft", "Google", "Apple"};  string[] hard = { "Apple", "IBM", "Samsung"};    // разность множеств  var result = soft.Except(hard);    foreach (string s in result)      Console.WriteLine(s); |

В данном случае из массива soft убираются все элементы, которые есть в массиве hard. Результатом операции будут два элемента:

Microsoft

Google

**Пересечение множеств**

Для получения пересечения множеств, то есть общих для обоих наборов элементов, применяется метод **Intersect**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | string[] soft = { "Microsoft", "Google", "Apple"};  string[] hard = { "Apple", "IBM", "Samsung"};    // пересечение множеств  var result = soft.Intersect(hard);    foreach (string s in result)      Console.WriteLine(s); |

Так как оба набора имеют только один общий элемент, то соответственно только он и попадет в результирующую выборку:

Apple

**Объединение множеств**

Для объединения двух множеств используется метод **Union**. Его результатом является новый набор, в котором имеются элементы, как из одного, так и из второго множества. Повторяющиеся элементы добавляются в результат только один раз:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | string[] soft = { "Microsoft", "Google", "Apple"};  string[] hard = { "Apple", "IBM", "Samsung"};    // объединение множеств  var result = soft.Union(hard);    foreach (string s in result)      Console.WriteLine(s); |

Результатом операции будет следующий набор:

Microsoft

Google

Apple

IBM

Samsung

Если же нам нужно простое объединение двух наборов, то мы можем использовать метод **Concat**:

|  |  |
| --- | --- |
| 1 | var result = soft.Concat(hard); |

Те элементы, которые встречаются в обоих наборах, дублируются.

**Удаление дубликатов**

Для удаления дублей в наборе используется метод **Distinct**:

|  |  |
| --- | --- |
| 1 | var result = soft.Concat(hard).Distinct(); |

Последовательное применение методов Concat и Distinct будет подобно действию метода Union.