

Практическое занятие 17

Работа с данными типа множество.

Цель:

1. Получение навыков в задании переменных типа множество и организации ввода и вывода данных типа множество.
2. Получение практических навыков в выполнении операций над множествами.

Оборудование:

- Компьютер
- Среда программирования Visual Studio.

Порядок выполнения работы

Теоретический материал

Множественный тип данных

Множество – это совокупность объектов, рассматриваемая как одно целое.

Для работы с множествами в библиотеке классов .NET Framework имеется целых два обобщённых класса `HashSet<T>` и `SortedSet<T>`, которые находятся в пространстве имён `System.Collections.Generic`. Различие между ними в том, что `SortedSet<T>` представляет упорядоченное множество.

Рассмотрим все наиболее часто встречающиеся операции над множествами:

1. `Add` — добавление элемента. Если такой элемент уже присутствует, то он не будет добавлен.

`Remove` — удаление элемента из множества.

2. `Union` — объединение множеств. Создается новое множество, включающее в себя все элементы из множества `A` и множества `B`. Если элемент содержится в обоих множествах, он будет добавлен однократно.

3. `Difference` — разность множеств. Создается новое множество, включающее в себя все элементы множества `A`, которые не входят в множество `B`.

4. `Intersection` — пересечение множеств. Создается новое множество, включающее в себя все элементы входящие одновременно и в множество `A`, и в множество `B`.

5. `Subset` — проверка на подмножество. Чтобы быть подмножеством, все элементы множества `A` должны содержаться в множестве `B`. Тогда множество `A` является подмножеством множества `B`.

Создание множества

Ниже приведён пример создания и заполнения двух множеств, состоящих из целых чисел.

```
1  HashSet<int> a = new HashSet<int>();
2  a.Add(1);
3  a.Add(15);
4  a.Add(45);
5  a.Add(7);
6  a.Add(8);
7  a.Add(22);
8  a.Add(73);
9  a.Add(10);
10 HashSet<int> b = new HashSet<int>();
11 b.Add(7);
12 b.Add(8);
13 b.Add(42);
14 b.Add(22);
15 b.Add(12);
```

Заполнение множеств можно осуществлять и непосредственно при создании. Для этого нужно просто передать в конструктор коллекцию `IEnumerable<T>`.

Задание для самостоятельной работы:

Дан текст из строчных латинских букв, за которым следует точка. Напечатать: - первые вхождения букв в текст, сохраняя их взаимный исходный порядок; - все буквы, входящие в текст не менее двух раз; - все буквы, входящие в текст по одному разу. Дана непустая последовательность слов из строчных русских букв; между соседними словами - запятая, за последним словом - точка. Напечатать в алфавитном порядке

Контрольные вопросы:

1. Что называется элементом множества?
2. Каким требованиям должны удовлетворять все элементы множества?
3. Что такое базовый тип множества? Как он задается?
4. Какие операции допустимы над множествами?
5. Как описываются переменные множеств