



Коллекции

Dictionary<T, V>

Еще один распространенный тип коллекции представляют словари. Словарь хранит объекты, которые представляют пару ключ-значение. Каждый такой объект является объектом структуры **KeyValuePair<TKey, TValue>**. Благодаря свойствам `Key` и `Value`, которые есть у данной структуры, мы можем получить ключ и значение элемента в словаре.

Рассмотрим на примере использование словарей:

```
Dictionary<int, string> countries = new Dictionary<int, string>(5);
countries.Add(1, "Russia");
countries.Add(3, "Great Britain");
countries.Add(2, "USA");
countries.Add(4, "France");
countries.Add(5, "China");

foreach (KeyValuePair<int, string> keyValue in countries)
{
    Console.WriteLine(keyValue.Key + " - " + keyValue.Value);
}

// получение элемента по ключу
string country = countries[4];
// изменение объекта
countries[4] = "Spain";
// удаление по ключу
countries.Remove(2);
```

Класс словарей также, как и другие коллекции, предоставляет методы `Add` и `Remove` для добавления и удаления элементов. Только в случае словарей в метод `Add` передаются два параметра: ключ и значение. А метод `Remove` удаляет не по индексу, а по ключу.

Так как в нашем примере ключами являются объекты типа `int`, а значениями - объекты типа `string`, то словарь в нашем случае будет хранить объекты `KeyValuePair<int, string>`. В цикле `foreach` мы их можем получить и извлечь из них ключ и значение.



Коллекции

Dictionary<T, V>

Кроме того, мы можем получить отдельно коллекции ключей и значений словаря:

```
Dictionary<char, Person> people = new Dictionary<char, Person>();
people.Add('b', new Person() { Name = "Bill" });
people.Add('t', new Person() { Name = "Tom" });
people.Add('j', new Person() { Name = "John" });

foreach (KeyValuePair<char, Person> keyValue in people)
{
    // keyValue.Value представляет класс Person
    Console.WriteLine(keyValue.Key + " - " + keyValue.Value.Name);
}

// перебор ключей
foreach (char c in people.Keys)
{
    Console.WriteLine(c);
}

// перебор по значениям
foreach (Person p in people.Values)
{
    Console.WriteLine(p.Name);
}
```

Здесь в качестве ключей выступают объекты типа `char`, а значениями - объекты `Person`. Используя свойство `Keys`, мы можем получить ключи словаря, а свойство `Values` соответственно хранит все значения в словаре.

Для добавления необязательно применять метод `Add()`, можно использовать сокращенный вариант:

```
Dictionary<char, Person> people = new Dictionary<char, Person>();
people.Add('b', new Person() { Name = "Bill" });
people['a'] = new Person() { Name = "Alice" };
```

Несмотря на то, что изначально в словаре нет ключа 'a' и соответствующего ему элемента, то он все равно будет установлен. Если же он есть, то элемент по ключу 'a' будет заменен на новый объект `new Person() { Name = "Alice" }`



Коллекции

Dictionary<T, V>

Инициализация словарей

В С# 5.0 мы могли инициализировать словари следующим образом:

```
Dictionary<string, string> countries = new Dictionary<string, string>
{
    {"Франция", "Париж"},
    {"Германия", "Берлин"},
    {"Великобритания", "Лондон"}
};

foreach(var pair in countries)
    Console.WriteLine("{0} - {1}", pair.Key, pair.Value);
```

То начиная с С# 6.0 доступен также еще один способ инициализации:

```
Dictionary<string, string> countries = new Dictionary<string, string>
{
    ["Франция"] = "Париж",
    ["Германия"] = "Берлин",
    ["Великобритания"] = "Лондон"
};
```