

96 Опишите работу с обобщенным словарем в языке C#?

Работа со словарем похожа на работу со списком. Только в словарь добавляются пары ключ-значение. Создание объекта словаря:

```
Dictionary <int, string> d = new Dictionary <int, string>();
```

При создании словаря указывается два обобщенных типа – тип ключа и тип значения (Dictionary<TKey, TValue>).

Добавление элементов:

```
d.Add(1, "строка 1");
```

```
d.Add(2, "строка 2");
```

```
d.Add(3, "строка 3");
```

В версии языка C# 6 появился более удобный синтаксис для инициализации словарей:

```
Dictionary <int, string> d1 = new Dictionary <int, string>
```

```
{
```

```
    [1] = "строка 1",
```

```
    [2] = "строка 2",
```

```
    [3] = "строка 3"
```

```
};
```

Вывод элементов:

При выводе элементов в цикле foreach необходимо объявить тип данных, который содержит пару ключ-значение, читаемых из словаря.

```
foreach (KeyValuePair <int, string> v in d)
```

```
{
```

```
    Console.WriteLine(v.Key.ToString() + "-" + v.Value);
```

```
}
```

У объекта класса словаря существуют коллекции ключей и значений: Keys и Values.

Пример вывода ключей и значений:

```
Console.Write("\nКлючи: ");
```

```
foreach (int i in d.Keys)
```

```
{
```

```
    Console.Write(i.ToString() + " ");
```

```
}
```

```
Console.Write("\nЗначения: ");
```

```
    foreach (string str in d.Values)
```

```
{
```

```
Console.Write(str + " ");  
}
```

Для получения значения по ключу можно использовать перегруженный индексатор, что аналогично применению индексатора для списка.

Пример получения значения по ключу:

```
int key = 3;  
string val = d[key];  
Console.WriteLine("\nДля ключа " + key.ToString() + " значение " + val + "");
```

Однако если указать ключ, отсутствующий в словаре, то это приведет к возникновению исключения `KeyNotFoundException`.

Для решения данной задачи можно также использовать метод `TryGetValue`. Он возвращает логическое значение: `true` если удалось прочесть значение по ключу и `false` если не удалось.

Пример использования метода

```
TryGetValue: string val2 = "";  
bool res = d.TryGetValue(key, out val2);  
if (res)  
{  
    Console.WriteLine("\nДля ключа " + key.ToString() + " значение " + val2 + "");  
}
```