Турсунов Баходурхон

#22. .NET Functional Techniques. Discards. Deconstructing types.

**1. Как обозначить, что переменная является "Discard" в C#? В чем смысл этой конвенции?**

Символ подчеркивания ( \_ ) может использоваться для обозначения переменной как Discard(проигнорировать). Это означает, что значение переменной не будет использовано, и компилятор не будет выдавать предупреждения о неиспользуемой переменной. Такая конструкция используется, когда вы обязаны указать переменную в сигнатуре метода или в цикле, но не намерены использовать ее значение.

**2. Каков общий синтаксис "Деконструкции" кортежей в C#?**

Деконструкция – это процесс извлечения отдельных значений из кортежа и присвоение их отдельным переменным. Общий синтаксис деконструкции кортежей выглядит следующим образом:

// Создание кортежа  
**var** myTuple = (1, "Hello", 3.14);  
  
// Деконструкция кортежа  
**var** (firstValue, secondValue, thirdValue) = myTuple;  
  
// Использование извлеченных значений  
Console.WriteLine($"First: {firstValue}, Second: {secondValue}, Third: {thirdValue}");

Деконструкция кортежей упрощает работу с несколькими значениями, предоставляя удобный синтаксис для извлечения и использования отдельных компонентов кортежа.

**3. Как разрешить неоднозначность, если у вас есть несколько методов «Deconstruct» с одинаковым количеством параметров?**

Если у нас есть несколько методов Deconstruct с одинаковым количеством параметров в одном классе, необходимо явно указать, какой метод использовать в конкретном контексту. Это можно сделать с использованием атрибута **System.Runtime.CompilerServices.TupleElementNamesAttribute**.

**class** **MyClass**  
{  
 **private** **int** value1;  
 **private** **string** value2;  
  
 // Первый метод Deconstruct  
 **public** **void** **Deconstruct**(**out** **int** val1, **out** **string** val2)  
 {  
 val1 = value1;  
 val2 = value2;  
 }  
  
 // Второй метод Deconstruct с атрибутом TupleElementNamesAttribute  
 [System.Runtime.CompilerServices.TupleElementNamesAttribute("Number", "Text")]  
 **public** **void** **Deconstruct**(**out** **int** number, **out** **string** text)  
 {  
 number = value1;  
 text = value2;  
 }  
  
 **public** **MyClass**(**int** number, **string** text)  
 {  
 value1 = number;  
 value2 = text;  
 }  
}  
  
**class** **Program**  
{  
 **static** **void** **Main**()  
 {  
 **var** myObject = **new** MyClass(42, "Hello");  
  
 // Использование первого метода Deconstruct  
 (**int** val1, **string** val2) = myObject;  
 Console.WriteLine($"Number: {val1}, Text: {val2}");  
  
 // Использование второго метода Deconstruct с атрибутом TupleElementNamesAttribute  
 (**int** number, **string** text) = myObject;  
 Console.WriteLine($"Number: {number}, Text: {text}");  
 }  
}

Practice:

1. Напишите функцию на C#, которая возвращает кортеж с двумя значениями (например, имя и возраст человека). Затем используйте "Деконструкцию" для получения и вывода этих значений.

2. Создайте класс "Книга" с полями для названия и автора книги. Реализуйте метод "Деконструкция" для одновременного получения названия книги и автора. Затем создайте объект "Book" и используйте метод "Deconstructing" для получения и вывода информации о книге.