**Что такое перечисление и как оно реализуется в языке C#?**

Перечисление (enum) – это множество целых чисел, за каждым из которых закреплена строковая константа. Перечисление позволяет перечислить множество значений какого-либо свойства.

В классическом языке С аналогом перечислений были объявления констант с помощью директивы препроцессора #define. Достаточно быстро программисты на языке С стали объявлять константы с «иерархическими» именами, в имя константы входили и наименование перечисления и наименование конкретного значения. В дальнейшем это привело к появлению перечислений, использующихся в языках C++, Java и C#.  
 Перечисления представляют набор логически связанных констант. Объявление перечисления происходит с помощью оператора **enum**. Далее идет название перечисления, после которого указывается тип перечисления - он обязательно должен представлять целочисленный тип (byte, int, short, long). Если тип явным образом не указан, то по умолчанию используется тип int. Затем идет список элементов перечисления через запятую:

|  |  |
| --- | --- |
|  | enum Days  {      Monday,      Tuesday,      Wednesday,      Thursday,      Friday,      Saturday,      Sunday  }    enum Time : byte  {      Morning,      Afternoon,      Evening,      Night  } |

В этих примерах каждому элементу перечисления присваивается целочисленное значение, причем первый элемент будет иметь значение 0, второй - 1 и так далее. Мы можем также явным образом указать значения элементов, либо указав значение первого элемента:

|  |  |
| --- | --- |
|  | enum Operation  {      Add = 1,   // каждый следующий элемент по умолчанию увеличивается на единицу      Subtract, // этот элемент равен 2      Multiply, // равен 3      Divide    // равен 4  } |

Но можно и для всех элементов явным образом указать значения:

|  |  |
| --- | --- |
|  | enum Operation  {      Add = 2,      Subtract = 4,      Multiply = 8,      Divide = 16  } |

Каждое перечисление фактически определяет новый тип данных. Затем в программе мы можем определить переменную этого типа и использовать ее:

|  |  |
| --- | --- |
|  | enum Operation      {          Add = 1,          Subtract,          Multiply,          Divide      }  class Program  {      static void Main(string[] args)      {          Operation op;          op = Operation.Add;          Console.WriteLine(op); // Add            Console.ReadLine();      }  } |

В программе мы можем присвоить значение этой переменной. При этом в качестве ее значения должна выступать одна из констант, определенных в данном перечислении. То есть несмотря на то, что каждая константа сопоставляется с определенным числом, мы не можем присвоить ей числовое значение, например, Operation op = 1;. И также если мы будем выводить на консоль значение этой переменной, то мы получим им константы, а не числовое значение. Если же необходимо получить числовое значение, то следует выполнить приведение к числовому типу:

|  |  |
| --- | --- |
|  | Operation op;  op = Operation.Multiply;  Console.WriteLine((int)op); // 3 |

Зачастую переменная перечисления выступает в качестве хранилища состояния, в зависимости от которого производятся некоторые действия. Например, для использования case можно использовать enum