Основы программирования

Особые классы

Перечисления

Определение

- Перечисление – набор связанных между собой именованных констант.
  Конкретные значения констант не важны -- важны сами имена констант и их принадлежность перечислению.
- Конкретное значение каждой константы можно задать из диапазона значений того целочисленного типа, на котором базируется описываемое перечисление.

Синтаксис

```
<атрибуты> <спецификаторы> enum <Имя> :
  <базовый тип>
{
  <константы перечисления>
}
```

- атрибуты и спецификаторы перечисления аналогичны атрибутам и спецификаторам класса;
- переменная типа перечисление в качестве значения может принимать константы перечисления.

Примеры

```
enum Time : byte
{ Morning, Afternoon, Evening, Night  }

enum Operation
{  Add = 1, Sub, Mult, Div }

enum Operation
{  Add = 2, Sub = 4, Mult = 8, Div = 16 }

enum Color
{  White = 1, Black = 2, Blue = White  }
```

Тип перечисления

- Базовым типом может быть один из целочисленных типов кроме `char`.
- Если базовый тип не указывается в описании перечисления, то по умолчанию базовым типом считается тип `int`.
- Каждая константа в перечислении имеет свое базовое целочисленное значение.
- Значения константам в перечислении задаются упорядоченно, начиная с 0. Эти значения можно задать явно.

Тело перечисления

- Перечисление констант с возможным явным указанием их целочисленных значений.
- Если значение константы не указано явно, то оно по умолчанию равно увеличенному на 1 значению предыдущей константы.
- В перечислении не может быть констант с одинаковыми целочисленными значениями.
- Также по умолчанию все константы в перечислении публичны

Пример

```
enum Color {Red, Green, Blue, Black}

class Figure
{
  public double area; //площадь фигуры
  public Color color; //цвет фигуры
…
}
```

Пример

```
…
Figure circle = new Figure();
circle.area = 2.54;
circle.color = Color.Red;
  //для фигуры задается красный цвет
…
```

Операции с переменными типа перечислений

- возможно выполнение арифметических операций: «+», «-», «++», «--», «>», «<», «>=», «<=», «==», «!=», «&», «|», «~», «^»
- операции получения размера в байтах `sizeof`. (операция выполняется над целочисленным значением константы перечисления).

Переменные типа перечислений

Могут использоваться в:

- арифметических выражениях и
- операциях присваивания.

Требуется явное преобразование к целому типу.

Состав класса Enum

Элемент	Описание	Вид
`GetName`	Получение имени константы по ее значению	Статические методы
`GetNames`	Получение массива имен констант в перечислении	
`GetValues`	Получение массива значений констант в перечислении	
`IsDefined`	Возвращает true, если строковое имя константы определено в перечислении	
`GetUnderlyingType`	Возвращает имя базового типа (по умолчанию `Int32`)	

Пример

```
Console.WriteLine(Enum.GetName(typeof(Color), 3));
if (!Enum.IsDefined(typeof(Color), ”Brown”))
 Console.WriteLine(”Константы Brown нет в
  перечислении Color”);
```

Все статические методы принимают в качестве параметра представление типа перечисления (ссылку типа `Type` на описание перечисления, которую можно получить с помощью операции `typeof`

Преимущества перечислений

- гарантия того, что переменным будут назначаться допустимые значения из указанного набора;
- код становится читабельнее, когда в нем присутствуют понятные имена, а не числа.

Широко используются в самой библиотеке классов `.NET`. Например, при создании файлового потока (`FileStream`) используется перечисление `FileAccess`, при помощи которого мы указываем с каким режимом доступа открыть файл (чтение/запись).

Полезные ссылки

C# и .NET | Перечисления enum (metanit.com)

Перечисления (enum) (programming-lessons.xyz)

enum в C#: что это и как его использовать (skillbox.ru)

Перечисления (enum) в Си-шарп (mycsharp.ru)