

IFRN

PROGRAMAÇÃO ORIENTADA A OBJETOS EM C#

Datas, Intervalos e Enumerações

Prof. Gilbert Azevedo

Objetivos

- Conhecer as estruturas DateTime e TimeSpan utilizadas para controlar datas e intervalos de tempo
- Utilizar enumerações no modelo de entidades

Tipos de Objetos

- Tipos por Valor
 - A variável contém uma instância do tipo
 - Estruturas – struct: int, double, char, bool, DateTime, TimeSpan
 - Enumerações – enum
- Tipos por Referência
 - A variável é uma referência para uma instância do tipo
 - class, interface, delegate
- Tipos por Referência Internos
 - dynamic, object, string

struct DateTime

- DateTime – Tipo por valor que representa um instante no tempo, expressando uma data e uma hora do dia
- Principais propriedades
 - Day, Month, Year – Retorna dia, mês ano da data/hora (int)
 - DayOfWeek – Dia da semana (DayOfWeek)
 - DayOfYear – Dia do ano (int)
 - Hour, Minute, Second – Retorna hora, minuto, segundo da data/hora (int)
- Propriedades estáticas
 - Now – Retorna o dia e hora atual
 - Today – Retorna o dia atual (zero hora)

Métodos de DateTime

- DateTime - Alguns Métodos
 - Add – Adiciona um período de tempo a data
 - AddDays, AddMonths, AddYears – Adiciona dias, meses e anos a uma data
 - AddHours, AddMinutes, AddSeconds – Adiciona hora, minutos e segundos a uma data
 - Parse – Converte uma string com informação de data e hora para seu equivalente de DateTime
 - ToString – Converte o DateTime em string

Exemplo de DateTime

- Iniciando com a data atual, com *Parse* e com *new*
 - `DateTime dt = DateTime.Now;`
 - `DateTime dt = DateTime.Parse("2014-10-30");`
 - `DateTime dt = new DateTime(2014, 10, 30);`
- Lendo uma data do usuário
 - `dt = DateTime.Parse(Console.ReadLine());`
- Adicionando um dia e mostrando a data
 - `dt = dt.AddDays(1);`
 - `Console.WriteLine(dt.ToString());`

struct TimeSpan

- TimeSpan – Tipo por valor que representa um intervalo de tempo, expresso em dia, hora, minuto, segundo, milissegundo
- Algumas Propriedades
 - Days – Número de dias do intervalo
 - Hours – Número de horas
 - Minutes – Número de minutos
 - Seconds – Número de segundos
 - Milliseconds – Número de milissegundos do intervalo

Exemplo de TimeSpan

- Iniciando uma data em “30/10/2014”
 - `DateTime dt = DateTime.Parse("2014-10-30");`
- Adicionando um dia, nove horas e cinquenta minutos
 - `TimeSpan ts = new TimeSpan(1, 9, 50, 0);`
 - `dt += ts;`
 - `Console.WriteLine(dt); // 31/10/2014 09:50:00`

Enumerações

- Tipo por valor definido por um conjunto de constantes nomeadas
- Enumera uma lista de valores que uma variável pode armazenar
 - Estado civil, escolaridade, dia da semana, mês, estação, informações de pagamento, etc

```
enum Estação {  
    Primavera, Verão, Outono, Inverno  
}  
  
enum Escolaridade {  
    Fundamental, Médio, Superior,  
    Pós-Graduação  
}
```

Exemplo de Enumeração

- Uma variável da enumeração pode receber um dos valores listados no tipo
- No exemplo, o texto “Primavera” é mostrado na saída

```
enum Estação {  
    Primavera, Verão, Outono, Inverno  
}  
  
class MainClass {  
    public static void Main (string[] args) {  
        Estação x = Estação.Primavera;  
        Console.WriteLine(x);           // Primavera  
    }  
}
```

Tipos Adjacentes

- A enumeração pode definir um tipo adjacente e um valor para cada constante. O tipo padrão é int e o valor inicial é zero.
- É possível converter o tipo adjacente em enumeração e vice-versa

```
enum Estação : byte {  
    Primavera = 1, Verão = 2, Outono = 3, Inverno = 4  
}  
  
class MainClass {  
    public static void Main (string[] args) {  
        Estação x = Estação.Primavera;  
        Console.WriteLine ((byte) x);           // 1  
        x = (Estação) 2;  
        Console.WriteLine (x);                 // Verão  
    }  
}
```

Enumerações com Sinalizadores

- Enumerações podem também definir um conjunto de opções
- Neste caso, deve utilizar o atributo Flags
- E os valores devem ser potências de dois para utilização de operadores binários | e &

```
[Flags]
public enum Dias
{
    Nenhum = 0, Segunda = 1,
    Terça = 2, Quarta = 4,
    Quinta = 8, Sexta = 16,
    Sábado = 32, Domingo = 64
}
```

Operadores | e &

- O operador | faz a união de dois ou mais elementos
- O operador & testa se um elemento está no conjunto

[Flags]

```
public enum Dias
{
    Nenhum = 0, Segunda = 1,
    Terça = 2, Quarta = 4,
    Quinta = 8, Sexta = 16,
    Sábado = 32, Domingo = 64
}
```

```
class MainClass {
    public static void Main (string[] args) {
        Dias curso = Dias.Segunda | Dias.Quarta;
        Dias fds = Dias.Sábado | Dias.Domingo;
        Console.WriteLine (curso);
        Console.WriteLine (fds);
        if ((curso & Dias.Segunda) != 0)
            Console.WriteLine("Segunda tem aula");
    }
}
```

Referências

- Microsoft Visual C# 2010 – Passo a passo, John Sharp, Bookman, 2010
- DateTime Estrutura
 - <https://docs.microsoft.com/pt-br/dotnet/api/system.datetime>
- TimeSpan Estrutura
 - <https://docs.microsoft.com/pt-br/dotnet/api/system.timespan>
- Tipos de Enumeração
 - <https://docs.microsoft.com/pt-br/dotnet/csharp/language-reference/builtin-types/enum>