

Usando a convenção de descoberta de tipos na propriedade Duração

Transcrição

O erro que presenciamos na última atividade é um erro de conversão (the expected type was 'System.Int32' but the actual value was of type 'System.Int16'). O programa está tentando converter um inteiro de 32 bits para um inteiro de 16 bits, o que não é possível. No C#, os tipos primitivos que estão relacionados ao inteiro de 32 bits(ou 4 bytes) e 16 bits(ou 2 bytes) são, respectivamente, o `int` e o `short` . Portanto, se no banco de dados só temos valores de 2 bytes, não faz sentido termos na classe `Filme` uma propriedade cujo tamanho é 4 bytes.

Observemos como está o código da classe `Filme` .

```
namespace Alura.Filmes.App.Negocio
{
    public class Filme
    {
        public int Id { get; set; }
        public string Titulo { get; set; }
        public string Descricao { get; set; }
        public string AnoLancamento { get; set; }
        public int Duracao { get; set; }
    }
}
```

Mudaremos o tipo da propriedade `Duracao` para um inteiro de 2 bytes, ou seja, um `short` .

```
namespace Alura.Filmes.App.Negocio
{
    public class Filme
    {
        public int Id { get; set; }
        public string Titulo { get; set; }
        public string Descricao { get; set; }
        public string AnoLancamento { get; set; }
        public short Duracao { get; set; }
    }
}
```

Com essa pequena mudança, perceberemos que ao executar o programa não ocorrerá mais nenhum erro de conversão.

```

C:\Windows\system32\cmd.exe

Executing DbCommand [Parameters=[], CommandType='Text', CommandTimeout='30']
SELECT [f].[film_id], [f].[release_year], [f].[description], [f].[length], [f].[title], [f].[last_update]
FROM [film] AS [f]

Executed DbCommand (6ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
SELECT [f].[film_id], [f].[release_year], [f].[description], [f].[length], [f].[title], [f].[last_update]
FROM [film] AS [f]

Alura.Filmes.App.Negocio.Filme
Alura.Filmes.App.Negocio.Filme
Alura.Filmes.App.Negocio.Filme
Alura.Filmes.App.Negocio.Filme
Alura.Filmes.App.Negocio.Filme
Alura.Filmes.App.Negocio.Filme
Alura.Filmes.App.Negocio.Filme
Alura.Filmes.App.Negocio.Filme
Alura.Filmes.App.Negocio.Filme
Alura.Filmes.App.Negocio.Filme
Alura.Filmes.App.Negocio.Filme
Alura.Filmes.App.Negocio.Filme
Alura.Filmes.App.Negocio.Filme
Alura.Filmes.App.Negocio.Filme
Alura.Filmes.App.Negocio.Filme

```

O programa ainda não está exibindo os valores, mas conseguimos resolver esse problema sobrescrevendo o método `ToString()` no código da classe `Filme`. Colocaremos o `Id`, `Título` e `AnoLancamento`.

```

namespace Alura.Filmes.App.Negocio
{
    public class Filme
    {
        public int Id { get; set; }
        public string Titulo { get; set; }
        public string Descricao { get; set; }
        public string AnoLancamento { get; set; }
        public short Duracao { get; set; }

        public override string ToString()
        {
            return $"Filme ({Id}): {Titulo} = {AnoLancamento}";
        }
    }
}

```

Ao executarmos novamente o programa, veremos que os valores estão sendo exibidos, e os filmes estão sendo listados no console como gostaríamos. Temos o modelo `Filme (1): ACADEMY DINOSAUR - 2006`, portanto, temos `Id`, `Título` e `AnoLancamento` em cada registro.