

Incorporar Classe

Transcrição

A próxima técnica de refatoração que veremos a seguir é chamada de **Inline Class**, ou *Incorporar Classe*. Essa técnica é o inverso da anterior: **Extrair Classe**.

Por que incorporamos uma classe? Fazemos essa refatoração quando percebemos que a classe não está fazendo praticamente nada, uma classe supérflua ou "preguiçosa". No Visual Studio, temos o projeto `refatoracao` que contém o arquivo `Empresa.cs`, localizado na pasta "Aula06 > R13.InlineClass > depois".

A classe `Empresa` utilizará outras três classes: `Endereco`, `CNPJ`, e `RazaoSocial`. Como vimos no vídeo anterior, a classe `Endereco` tem a sua importância porque ela agrupa os endereços de cobrança e de entrega, e também serve para eliminar duplicação de código. Entretanto, a classe `CNPJ` e a classe `RazaoSocial` não estão fazendo praticamente nada, como podemos ver a seguir:

```
class CNPJ
{
    public string Codigo { get; private set; }

    public CNPJ(string codigo)
    {
        this.Codigo = codigo;
    }
}

class RazaoSocial
{
    public string Nome { get; private set; }

    public RazaoSocial(string nome)
    {
        Nome = nome;
    }
}
```

Vamos incorporar o conteúdo dessas classes em `Empresa`.

Primeiro, vamos substituir a classe `RazaoSocial` pelas informações que ela armazena. Copiaremos a propriedade única `public string Nome { get; private set; }` e colaremos no começo da classe `Empresa`:

```
class Empresa
{
    public string Nome { get; private set; }
    public RazaoSocial RazaoSocial { get; private set; }
    public CNPJ CNPJ { get; private set; }
    public Endereco EnderecoEntrega { get; private set; }
    public Endereco EnderecoCobranca { get; private set; }
}
```

Onde tínhamos somente `RazaoSocial` `RazaoSocial { get; private set; }`, agora teremos uma `string`. Vamos chamá-la de `RazaoSocial`, e então deixaremos a primeira linha referente a `RazaoSocial`:

```
class Empresa
{
    public string RazaoSocial { get; private set; }
    public CNPJ CNPJ { get; private set; }
    public Endereco EnderecoEntrega { get; private set; }
    public Endereco EnderecoCobranca { get; private set; }
}
```

Como podemos ver, a `RazaoSocial` foi adicionada como uma `string`. No construtor da `Empresa`, recebemos `razaoSocial` como objeto, a seguir, mudaremos para o tipo `string`:

```
public Empresa(string razaoSocial, CNPJ cnpj, Endereco endereco....)
```

Trocaremos de `CNPJ` para `string` também, não nos esquecendo de trocar no construtor:

```
class Empresa
{
    public string RazaoSocial { get; private set; }
    public string CNPJ { get; private set; }
    public Endereco EnderecoEntrega { get; private set; }
    public Endereco EnderecoCobranca { get; private set; }
}

public Empresa(string razaoSocial, string cnpj, Endereco endereco....)
```

Desta forma, eliminaremos as duas classes: a `CNPJ` e a `RazaoSocial`, assim deixaremos o código mais enxuto e funcionando como anteriormente.