01

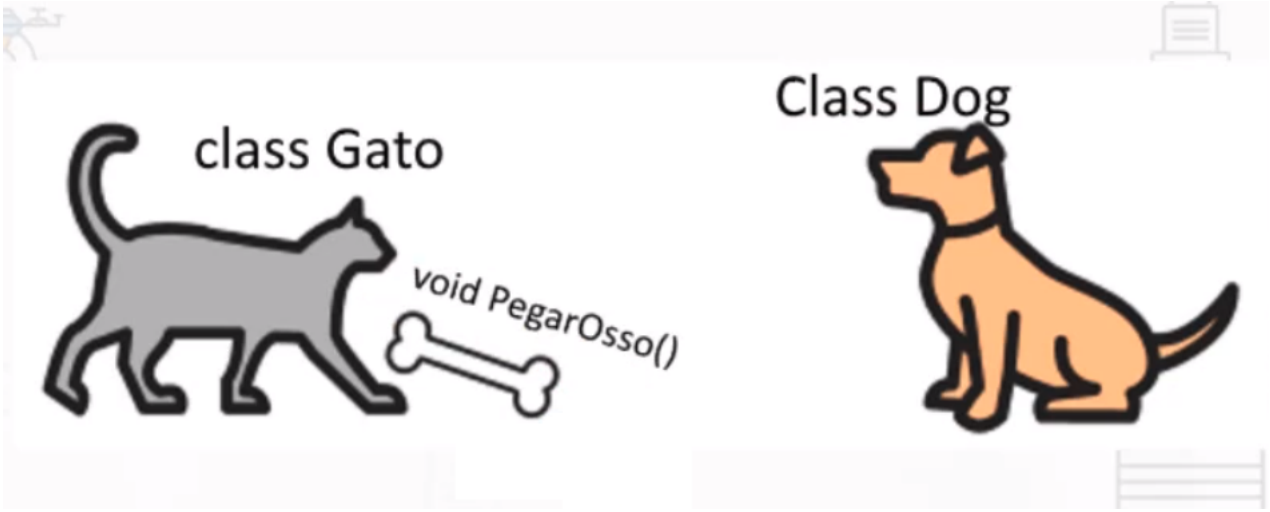
Mover Método

Transcrição

Começando deste ponto? Você pode fazer o DOWNLOAD (https://github.com/alura-cursos/csharp-refatorando-codigo/archive/5775056c110502f2b5620632f0af0f208fc84696.zip) completo do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo.

Terminamos a parte de **Técnicas de Composição**. Agora veremos a respeito das **Técnicas para Mover Itens entre Objetos**, e esses objetos são *classes,métodos, campos, propriedades*. A primeira tarefa desse novo grupo de técnicas para mover itens entre objetos é a de **mover método**.

Vamos imaginar o seguinte cenário: Temos duas classes, `Gato` e `Dog`. A classe `Gato` possui um método *void* chamado `PegarOsso()`.



Para acessibilidade: Um gato e um cachorro, ambos representando classes. A classe `Gato` possui o método `PegarOsso`.

Será que esse método é adequado para a classe `Gato`? É claro que não. Seria mais natural que esse método estivesse na classe `Dog`. Então, vamos refatorar essas classes e mover o método `PegarOsso()` para a classe `Dog`.

Para retratar essa tarefa em nosso escopo, no Visual Studio, usaremos o projeto `refatoracao`. O arquivo utilizado é o `ContaCorrente.cs`, que está localizado em "Aula05 > R10.MoveMethod > depois". Esse arquivo comporta duas classes, a `ContaCorrente` e a classe `TipoConta`.

Dentro da `ContaCorrente`, temos uma referência para os tipos de conta:

```
class ContaCorrente
{
    public decimal GetTaxaChequeEspecial()...

    public ContaCorrente(TipoConta type)...

    private readonly TipoConta tipo;
```

```
    public TipoConta Tipo { get {return tipo; } }

    private int diasEmDescoberto;
    public int DiasEmDescoberto...
}
```

Então toda conta é de um tipo. Como podemos observar, há o método `GetTaxaChequeEspecial()` para obter a taxa do cheque especial. Cada vez que um cliente fica com o saldo negativo por um certo tempo, o banco cobrará uma taxa.

Em nosso algoritmo, essa taxa será calculada com base no *tipo* da conta bancária.

```
    public decimal GetTaxaChequeEspecial()
    {
        if (tipo.EhPremium)
        {
            var result = 10.0M;
            if (this.diasEmDescoberto > 7)
            {
                result += (diasEmDescoberto - 7) * 0.01M;
            }
            return result;
        }
        return diasEmDescoberto * 1.75M;
    }
```

O cálculo da taxa do cheque especial, é algo que depende principalmente do tipo da conta. Mas como já temos uma classe para os tipos de conta, seria mais adequado mover o método `GetTaxaChequeEspecial()` para a classe `TipoConta`. Começaremos a refatoração "Mover Método" *copiando* o método `GetTaxaChequeEspecial()` para a classe de destino.

```
class TipoConta
{
    public bool EhPremium { get; set }

    public decimal GetTaxaChequeEspecial()
    {
        if (tipo.EhPremium)
        {
            var result = 10.0M;
            if (this.diasEmDescoberto > 7)
            {
                result += (diasEmDescoberto - 7) * 0.01M;
            }
            return result;
        }
        return diasEmDescoberto * 1.75M;
    }
}
```

Encontraremos alguns problemas após fazermos a cópia do método para a classe `TipoConta`, por exemplo, não temos mais a referência para o `tipo`. Ela seria a própria instância na qual o método `GetTaxaChequeEspecial()` está sendo executado. Por isso, trocaremos o `tipo` pelo `this`:

```
    public decimal GetTaxaChequeEspecial()
    {
        if (this.EhPremium)
        {
```

Outro problema que temos é o *antigo* `this`. Essa referência estava relacionada com a `ContaCorrente`, mas não temos a `ContaCorrente` dentro da classe `TipoConta`. Por isso, é necessário transformar a `ContaCorrente` em um **parâmetro** do novo método movido para a `TipoConta`. Depois, trocaremos a referência antiga `this` pela variável passada via parâmetro, lembrando de referenciar a propriedade `DiasEmDescoberto`:

```
    public decimal GetTaxaChequeEspecial(ContaCorrente conta)
    {
        if (tipo.EhPremium)
        {
            var result = 10.0M;
            if (conta.DiasEmDescoberto > 7)
            {
                result += (conta.DiasEmDescoberto - 7) * 0.01M;
            }
            return result;
        }
        return conta.DiasEmDescoberto * 1.75M;
    }
```

Em seguida, removeremos o corpo desse método `GetTaxaChequeEspecial()`, e passar uma referência para o `tipo` chamando o `GetTaxaChequeEspecial()` passando a referência `this`.

```
class ContaCorrente
{
    public decimal GetTaxaChequeEspecial()
    {
        return tipo.GetTaxaChequeEspecial(this);
    }
```

Com isso, conseguimos fazer a refatoração "Mover Método" com sucesso.