01

Quebrar Variável Temporária

Transcrição

Começando deste ponto? Você pode fazer o DOWNLOAD (https://github.com/alura-cursos/csharp-refatorando-codigo/archive/8ee0d4c8a90082bd9dbb4940d22b051e7f2de536.zip) completo do projeto do capítulo anterior.

A próxima técnica de refatoração que abordaremos é **Quebrar uma variável temporária em 2 ou mais**. Para isso, usaremos o projeto `refatoracao` no Visual Studio. Dentro desse projeto, temos um arquivo chamado de `Retangulo.cs`, que se encontra na pasta "Aula03 > R06.SplitTemporaryVariable > depois".

Dentro da classe `Retangulo`, temos um construtor que realiza o cálculo e imprime tanto o perímetro de um retângulo, como a sua área.

```
class Retangulo
{
    public Retangulo(double altura, double largura)
    {
        double temp = 2 * (altura + largura);
        System.Console.WriteLine($"Perímetro: {temp}");

        temp = altura * largura;
        System.Console.WriteLine($"Área: {temp}");
    }
}
```

O perímetro é calculado pela soma dos lados de uma figura.

A área é calculada pela multiplicação da altura pela largura.

Como vemos, temos uma declaração de uma variável temporária `temp`, que recebe o valor do cálculo do perímetro do retângulo. Em seguida, a mesma variável `temp` é usada para armazenar o calculo da área. Qual é o problema disso?

1) A variável `temp` **não explica o que faz**. Para saber isso, precisamos analisar a sua declaração e ver o que ela está armazenando. 2) A variável `temp` está sendo usada em dois propósitos, ou seja, ela está acumulando tarefas, quebrando o princípio da responsabilidade única. O terceiro problema é que temos dois trechos de código que deveriam ser independentes.

Se quisermos realizar o cálculo da área *antes* do perímetro, será que mover o trecho referente à área para antes do perímetro resolveria o problema?

```
class Retangulo
{
    public Retangulo(double altura, double largura)
    {
        temp = altura * largura;
```

```
            System.Console.WriteLine($"Área: {temp}");

            double temp = 2 * (altura + largura);
            System.Console.WriteLine($"Perímetro: {temp}");
        }
    }
```

Isso não seria possível. Agora, a variável `temp` está sendo usada **antes** da declaração e o código não compilará. Como vimos, são dois trechos independentes. Deixe o método como estava, caso você tenha mudado os trechos de posição.

Hora de refatorar

A primeira tarefa é criar outras variáveis nas quais `temp` será usada.

```
    public Retangulo(double altura, double largura)
    {
        double temp = 2 * (altura + largura);
        System.Console.WriteLine($"Perímetro: {temp}");

        double temp2 = altura * largura;
        System.Console.WriteLine($"Área: {temp2}");
    }
```

Agora temos as variáveis `temp` e `temp2`, mas ainda não explicamos o que elas fazem. É necessário renomeá-las com nome expressivo, e para isso utilizaremos o atalho de refatoração para renomeá-las, o "Ctrl + RR". Selecionamos a primeira variável, utilizamos o atalho, e demos o nome de `perimetro` - o nome também foi mudado na linha que imprime essa variável:

```
        double perimetro = 2 * (altura + largura);
        System.Console.WriteLine($"Perímetro: {perimetro}");
```

Faremos o mesmo com a variável `temp2`, e daremos o nome de `area`, que representa bem a expressão que está armazenando.

```
    public Retangulo(double altura, double largura)
    {
        double perimetro = 2 * (altura + largura);
        System.Console.WriteLine($"Perímetro: {perimetro}");

        double area = altura * largura;
        System.Console.WriteLine($"Área: {area}");
    }
```

A refatoração quebrou a variável em duas ou mais.