

Conclusão

Transcrição

Chegamos ao fim do curso de refatoração de código com C# da Alura. Vamos revisar os assuntos que vimos até aqui.

Técnicas de Composição

1 - Extrair método: serve para **eliminar código duplicado, quebrar um método** muito grande em métodos menores, eliminar comentários e também cadeias de mensagens.

2 - A técnica de Incorporar Método é usada quando o corpo do método é tão obvio quanto seu nome.

Code Smells - "mau cheiro"

a - Código duplicado: viola o princípio **DRY** (*Don't Repeat Yourself*).

b - Método longo: esse método faz geralmente mais do que deveria. Viola o princípio **SRP** (*Simple Responsibility Principle*).

c - Comentários: podem acabar deixando o código confuso, estar desatualizados, enganar o desenvolvedor, e inibir a refatoração.

Voltando às **Técnicas de Composição**, temos a seguinte:

3 - Extrair variável: podemos quebrar uma expressão complexa que é atribuída a uma variável, em diversas variáveis a fim de simplificar o entendimento da aplicação.

4 - Incorporar variável: é o inverso da técnica *extrair variável*, que serve para incorporar uma expressão que é tão simples quanto o nome da variável.

5 - Substituir variável por consulta a método: com essa técnica, facilitamos a utilização da expressão para que ela seja consumida por outras partes da aplicação.

6 - Quebrar uma variável em duas ou mais: permitimos o entendimento de cada uma das variáveis.

7 - Remover atribuição a parâmetros: aprendemos a remover atribuições que são feitas a parâmetros e métodos, pois isso acaba prejudicando o entendimento de um método, e eliminando o valor inicial que foi recebido pelo método.

8 - Substituir um método com um objeto-método: quando a refatoração é difícil, e é necessário mover tanto os dados como também o comportamento do método para um outro objeto.

9 - Substituir o algoritmo: quando o código atual funciona, mas tivermos uma ideia bem melhor, ou quando o substituímos para que ele tenha uma qualidade melhor.

Técnicas para Mover Itens entre Objetos

10 - Mover o método: percebemos que um método ficaria mais adequado em uma outra classe.

11 - Mover campo: assim como a técnica *Mover método*, ela permite que um campo seja localizado na classe onde ele é mais utilizado.

12 - Extrair uma classe: extraímos uma classe quando ela é maior do que deveria, possui dados aglomerados, duplicação de código e quando existe o uso de muitos tipos primitivos.

13 - Incorporar uma classe: mover os membros (campos, métodos) de uma classe para outra, já que a mesma é supérflua e não faz nada.

14 - Esconder o delegado: quando acessamos um objeto a partir de uma cadeia de objetos, para chegar no método final, e com essa técnica, encapsulamos a delegação no método aplicando a refatoração.

15 - Remover o intermediário: aprendemos a remover uma classe intermediária quando ela não faz nada, ou quando ela não acrescenta nenhum valor à aplicação.

16 - Introduzir método externo: quando não podemos modificar uma classe para adicionar funcionalidade, criamos um método externo para essa nova funcionalidade.

17 - Introduzir extensão local: uma classe-servidora que estamos usando precisa de vários métodos adicionais, mas não podemos modificá-la.

Com isso, terminamos o curso de *C# Refatorando Código*. Espero que você tenha gostado! [Participe da comunidade Alura \(https://cursos.alura.com.br/forum/curso-csharp-refatorando-codigo/todos/\)](https://cursos.alura.com.br/forum/curso-csharp-refatorando-codigo/todos/) tirando suas dúvidas ou deixando seu comentário no fórum.

Até a próxima!