

alura

Cursos Online de Tecnologia

C#: Refatorando Código

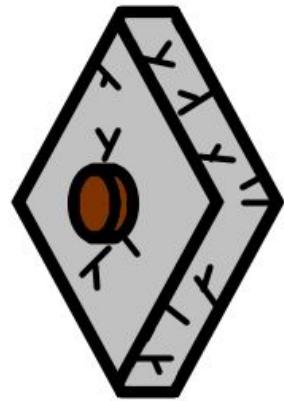
Parte 1

Código Limpo

- sem duplicação
- possui clareza
- menos classes
- fácil de manter
- passa em todos os testes

Débito Técnico

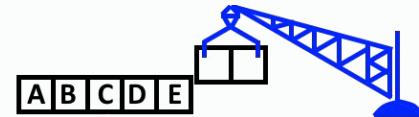
"Não tenho tempo de trocar
o meu modelo de roda"



VS



Técnicas de Composição



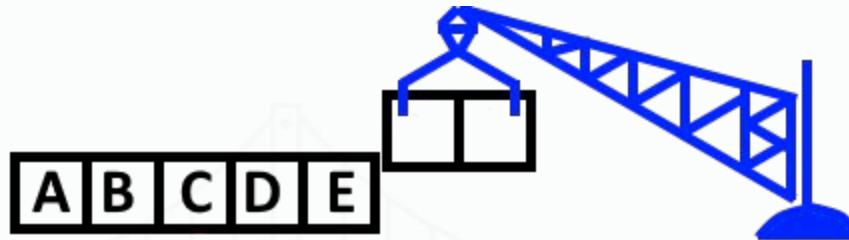
- 1) Extrair Método
- 2) Incorporar Método
- 3) Extrair Variável
- 4) Incorporar Variável Temporária
- 5) Substituir Variável por Consulta
- 6) Quebrar Variável em 2 ou Mais
- 7) Remover Atribuições a Parâmetros
- 8) Substituir Método por Método-Objeto
- 9) Substituir Algoritmo

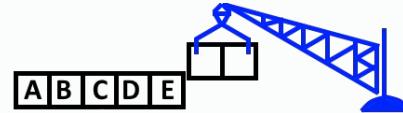
Mover Itens Entre Objetos



- 10) Mover Método
- 11) Mover Campo
- 12) Extrair Classe
- 13) Incorporar Classe
- 14) Ocultar Delegado
- 15) Remover Intermediário
- 16) Introduzir Método Estrangeiro
- 17) Introduzir Extensão Local

Técnicas de Composição





1) Extract Method (extrair método)

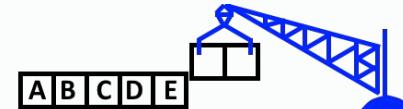
Você tem um trecho de código que pode ser agrupado

Quando refatorar?

- Código duplicado
- Método muito grande
- Comentários
- Cadeias de mensagens

Quando NÃO refatorar?

- Corpo do método vai ficar tão óbvio quanto seu nome



2) Inline Method (incorporar método)

Você tem um método cujo corpo é tão óbvio quanto seu nome.

Quando refatorar?

- Corpo do método é tão óbvio quanto seu nome, e o método se torna desnecessário

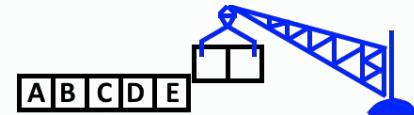
Quando NÃO refatorar?

- O corpo do método não é auto-explicativo:

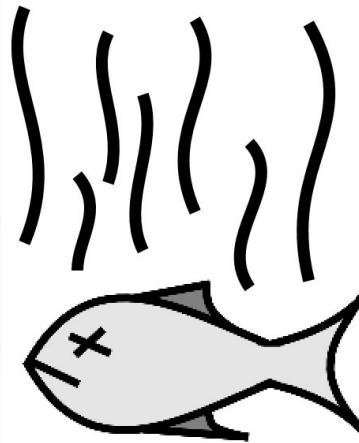
int GetNextLevel() { return level + 1; }

é óbvio, mas

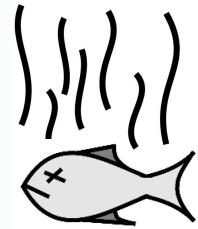
*int GetValue() { return (valorBase + acrescimo - desconto) * promocao; }* não deveria ser incorporado



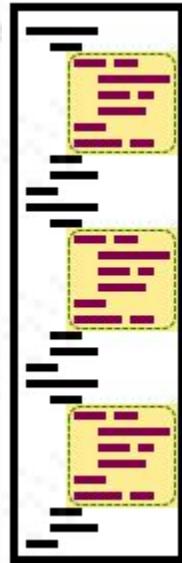
Code Smells



Code Smells

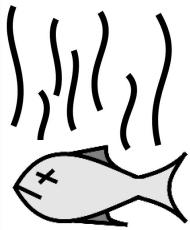


Código
Duplicado

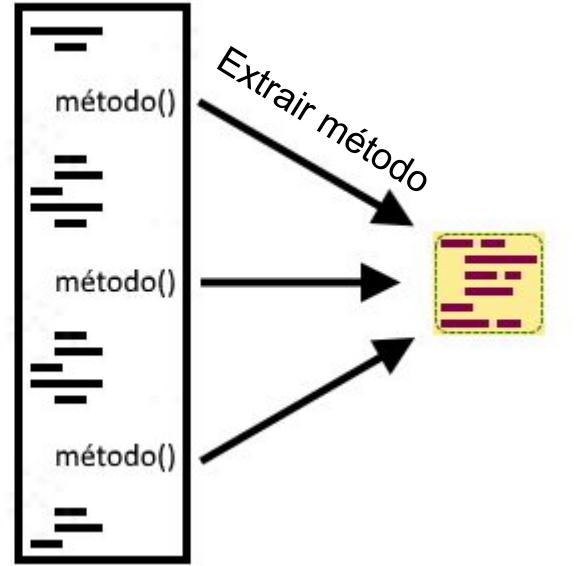
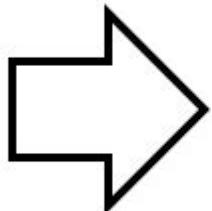
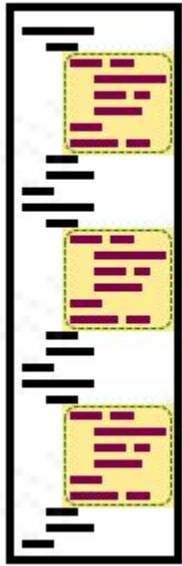


Viola o princípio **DRY**:
Don't Repeat Yourself, "Não se Repita".

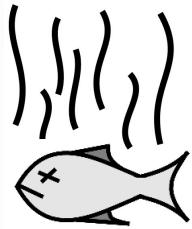
Code Smells



Código
Duplicado



Code Smells

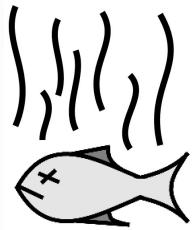


Método Longo

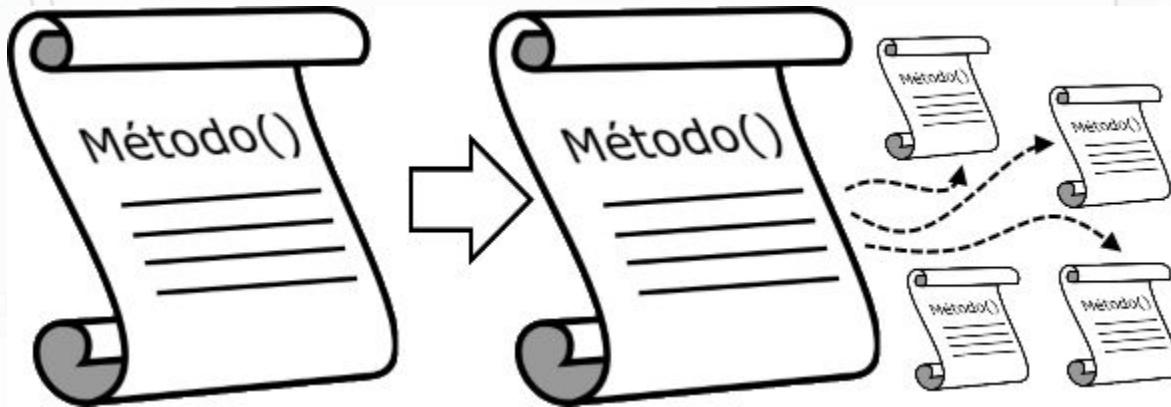


- Viola Princípio de Responsabilidade Única (SRP, Simple Responsibility Principle)
- Métodos longos são mais difíceis de ler
- Métodos curtos são mais fáceis de comparar
- Métodos curtos são mais fáceis de entender

Code Smells

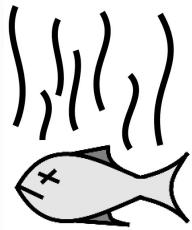


Método Longo

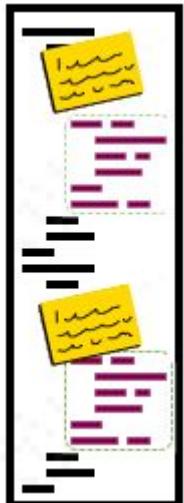


"Code never lies,
comments sometimes do."
-Anonymous

Code Smells



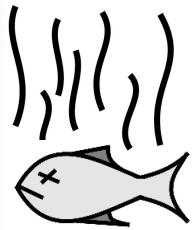
Comentários



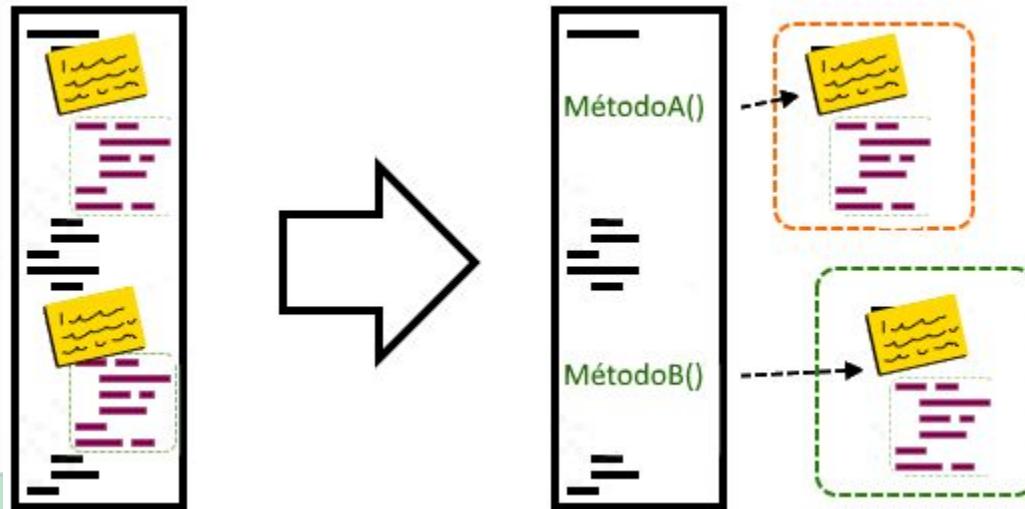
- Podem não explicar direito o código
- Podem estar mais desatualizados que o código
- Podem enganar o desenvolvedor
- Podem inibir a refatoração

"Code never lies,
comments sometimes do."
-Anonymous

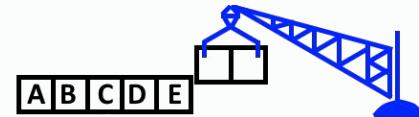
Code Smells



Comentários

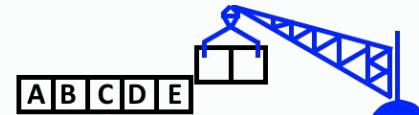


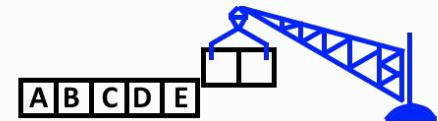
Técnicas de Composição



- 1) ~~Extrair Método~~
- 2) ~~Incorporar Método~~
- 3) Extrair Variável
- 4) Incorporar Variável Temporária
- 5) Substituir Variável por Consulta
- 6) Quebrar Variável em 2 ou Mais
- 7) Remover Atribuições a Parâmetros
- 8) Substituir Método por Método-Objeto
- 9) Substituir Algoritmo

Técnicas de Composição





3) Extract Variable (extrair variável)

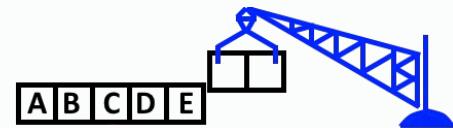
Partes de uma expressão complexa são armazenadas em variáveis explicativas

Quando refatorar?

- Uma expressão complexa pode ser quebrada em partes (variáveis) mais simples

Porém...

- Você terá mais variáveis espalhadas no código



4) Inline Temp (incorporar variável)

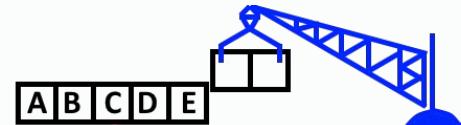
Você tem uma variável temporária que só recebe valor uma vez, e ela recebe o valor de uma expressão simples

Quando refatorar?

- Expressão da variável é tão óvia quanto seu nome, e a variável se torna desnecessária

Quando NÃO refatorar?

- Expressão da variável não é auto-explicativa: **var proximoNivel = level + 1;** é óvia, mas **var vlrPromocional = (valorBase + acrescimo - desconto) * promocao;** não deveria ser incorporado



5) Replace Temp With Query

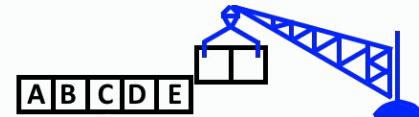
(Substituir variável por consulta a método)

Você está armazenando o resultado de uma expressão em uma variável temporária

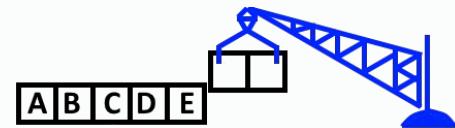
Por que refatorar?

- O método pode ser reutilizado em outros lugares, mas a variável não.

Técnicas de Composição



- 1) ~~Extrair Método~~
- 2) ~~Incorporar Método~~
- 3) ~~Extrair Variável~~
- 4) ~~Incorporar Variável Temporária~~
- 5) Substituir Variável por Consulta
- 6) Quebrar Variável em 2 ou Mais
- 7) Remover Atribuições a Parâmetros
- 8) Substituir Método por Método-Objeto
- 9) Substituir Algoritmo



6) Split Temporary Variable (Quebrar uma variável em duas ou mais)

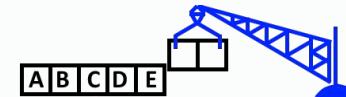
Você tem uma variável temporária que recebe valor mais de uma vez

Por que refatorar?

- Uma variável com mais de uma responsabilidade
- Nome da variável não explica o que ela armazena
- Todo o código fica “amarrado” com a linha onde a variável é declarada
- Código pouco legível

Quando NÃO refatorar?

- Se a variável só é usada uma vez



7) Remove Assignments to Parameters

(Remover atribuição a parâmetros)

Você modificou um parâmetro de entrada dentro de um método

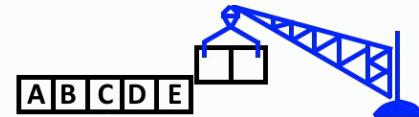
Quando refatorar?

- Você está
“economizando” variável ao
modificar um parâmetro de
entrada.

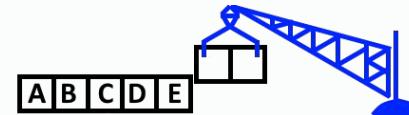
Quando NÃO refatorar?

- Em métodos recursivos
ou não, quando o
parâmetro funciona como
acumulador

Técnicas de Composição



- 1) ~~Extrair Método~~
- 2) ~~Incorporar Método~~
- 3) ~~Extrair Variável~~
- 4) ~~Incorporar Variável Temporária~~
- 5) ~~Substituir Variável por Consulta~~
- 6) ~~Quebrar Variável em 2 ou Mais~~
- 7) ~~Remover Atribuições a Parâmetros~~
- 8) ~~Substituir Método por Método-Objeto~~
- 9) ~~Substituir Algoritmo~~



8) Replace Method With Method Object

(Substitua um método com um objeto-método)

Um método é tão complexo que você não consegue extrair um método

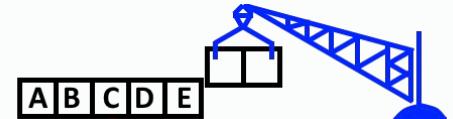
Quando refatorar?

- Um método é longo e complexo, mesmo assim não dá pra extrair métodos menores a partir dele.

Porém...

- Você vai precisar de mais uma classe, o que aumenta a complexidade do programa

9) Substitute Algorithm (Substituir o algoritmo)



O algoritmo atual funciona, mas você teve uma ideia bem melhor

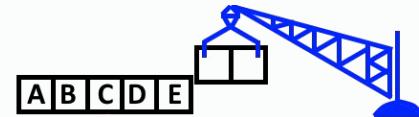
Quando refatorar?

- Quando você vê um algoritmo e sente que pode fazer melhor, simplificando ou esclarecendo o código.

Quando NÃO refatorar?

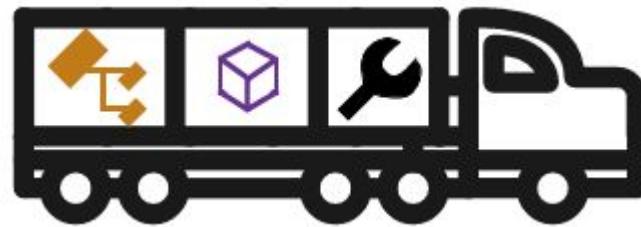
- Você não entendeu direito o código original, ou não tem testes unitários suficientes para garantir que o programa vai funcionar como antes.

Técnicas de Composição



- 1) ~~Extrair Método~~
- 2) ~~Incorporar Método~~
- 3) ~~Extrair Variável~~
- 4) ~~Incorporar Variável Temporária~~
- 5) ~~Substituir Variável por Consulta~~
- 6) ~~Quebrar Variável em 2 ou Mais~~
- 7) ~~Remover Atribuições a Parâmetros~~
- 8) ~~Substituir Método por Método-Objeto~~
- 9) ~~Substituir Algoritmo~~

Técnicas Para Mover Itens Entre Objetos



Mover Itens Entre Objetos

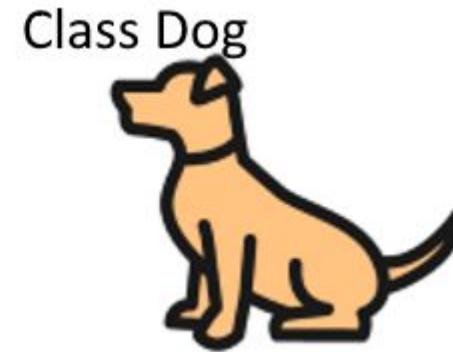
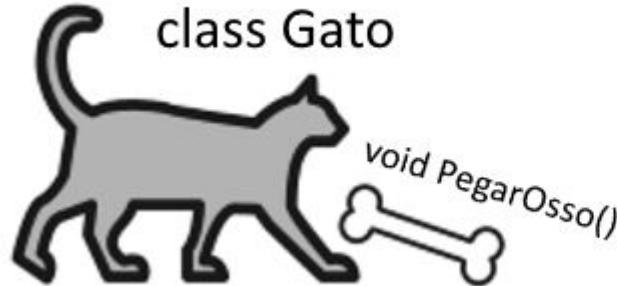


- 10) Mover Método
- 11) Mover Campo
- 12) Extrair Classe
- 13) Incorporar Classe
- 14) Ocultar Delegado
- 15) Remover Intermediário
- 16) Introduzir Método Estrangeiro
- 17) Introduzir Extensão Local

Mover Itens Entre Objetos



10) Move Method (Mover o método)





10) Move Method (Mover o método)

Um método é movido de uma classe para outra

Quando refatorar?

- Um método é mais usado por outra classe do que pela própria classe onde ele se encontra

Quando NÃO refatorar?

- O método é estático e faz parte de uma “biblioteca de métodos”

Mover Itens Entre Objetos

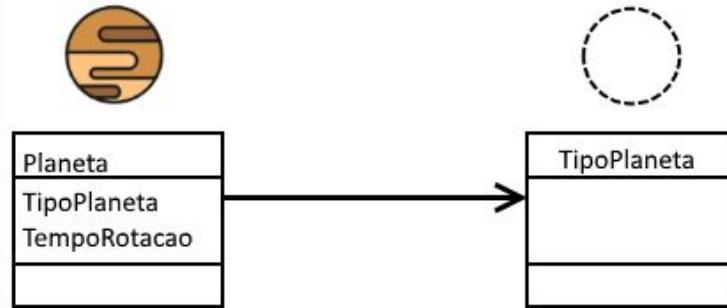


- 10) Mover Método
- 11) Mover Campo
- 12) Extrair Classe
- 13) Incorporar Classe
- 14) Ocultar Delegado
- 15) Remover Intermediário
- 16) Introduzir Método Estrangeiro
- 17) Introduzir Extensão Local

Mover Itens Entre Objetos



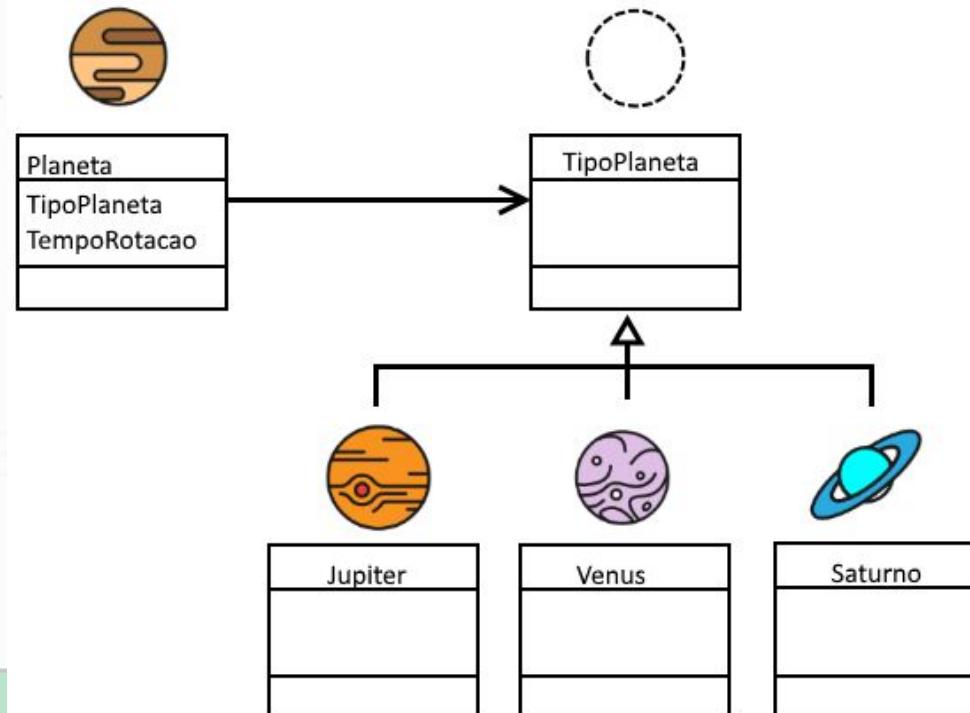
11) Move Field (Mover campo)



Mover Itens Entre Objetos



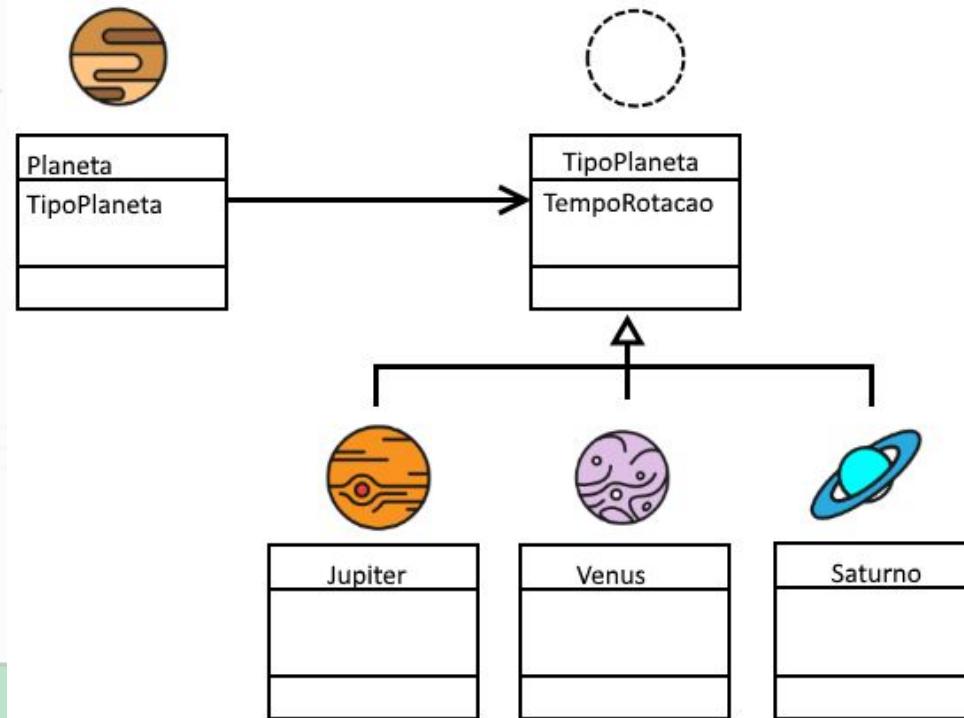
11) Move Field (Mover campo)



Mover Itens Entre Objetos



11) Move Field (Mover campo)





11) Move Field (Mover campo)

Um campo é movido de uma classe para outra

Quando refatorar?

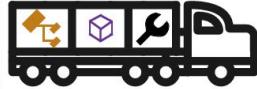
- um campo é mais usado por outra classe do que na própria classe onde ele está

Mover Itens Entre Objetos



- ~~10) Mover Método~~
- ~~11) Mover Campo~~
- 12) Extrair Classe
- 13) Incorporar Classe
- 14) Ocultar Delegado
- 15) Remover Intermediário
- 16) Introduzir Método Estrangeiro
- 17) Introduzir Extensão Local

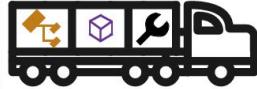
Mover Itens Entre Objetos



12) Extract Class (Extrair uma classe)



Mover Itens Entre Objetos



12) Extract Class (Extrair uma classe)



Mover Itens Entre Objetos



12) Extract Class (Extrair uma classe)

CAMPO CAMPO
CAMPO CAMPO

**NOVA
CLASSE**



12) Extract Class (Extrair uma classe)

Partes de uma classe podem formar uma nova classe

Quando refatorar?

- Uma classe está fazendo trabalho de duas (viola princípio da resp. única)
- Code Smell: Classe Muito Longa
- Code Smell: Duplicação de Dados
- Code Smell: Dados aglomerados
- Code Smell: Obsessão por tipos primitivos

Porém...

- Em excesso, pode produzir classes supérfluas



13) Inline Class (Incorporar a classe)

Mover os membros (campos, métodos) de uma classe para outra

Quando refatorar?

- Uma classe é supérflua, “preguiçosa”, ou “anêmica”.
Ela não está fazendo praticamente nada

Quando NÃO refatorar?

- A refatoração irá deixar a classe de destino com responsabilidades demais

Mover Itens Entre Objetos



- ~~10) Mover Método~~
- ~~11) Mover Campo~~
- ~~12) Extrair Classe~~
- ~~13) Incorporar Classe~~
- 14) Ocultar Delegado
- 15) Remover Intermediário
- 16) Introduzir Método Estrangeiro
- 17) Introduzir Extensão Local

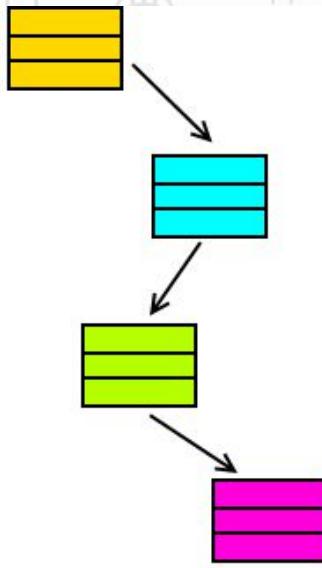
Mover Itens Entre Objetos



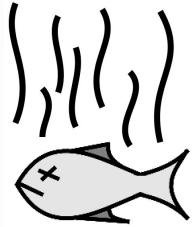
14) Hide Delegate (Esconder o delegado)



Cadeia de Mensagens

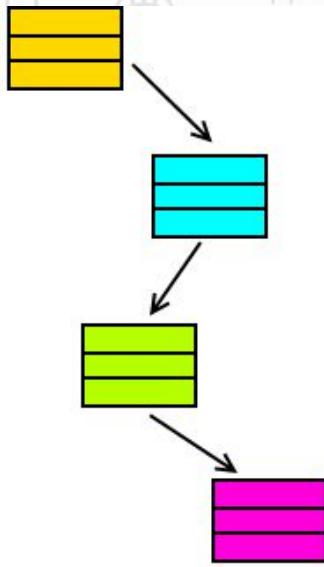


Code Smells



- **Exemplo:** var resultado =
this.GetObjetoA().GetObjetoB().GetObjetoC()
.GetObjetoD();
- Uma longa cadeia de chamadas de métodos de objetos, criando dependências desnecessárias no cliente:
- Viola a “Lei de Demeter”

Cadeia de Mensagens



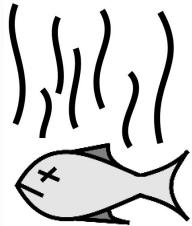
ANTES:

```
var custo =  
    this.Obra.GrupoMateriais().GrupoAtual().CotacaoPadrao.Ob  
terCusto().UltimoCusto.Valor();
```

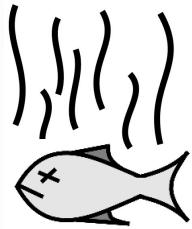
DEPOIS:

```
var custo = this.Obra.GetCusto();
```

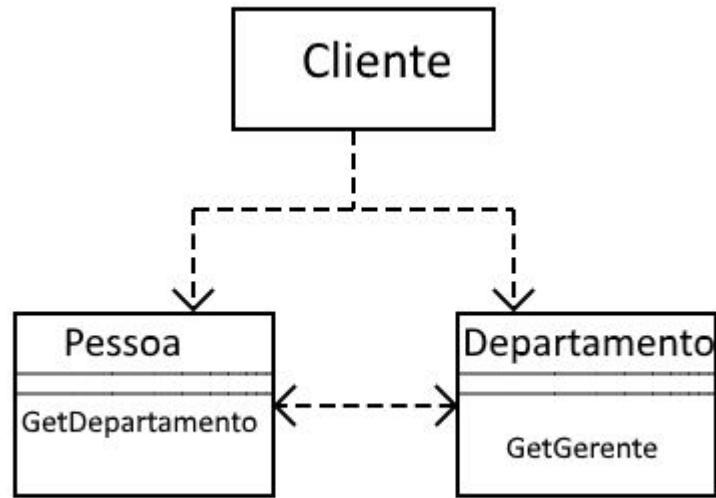
Code Smells



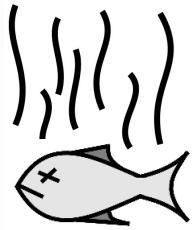
Code Smells



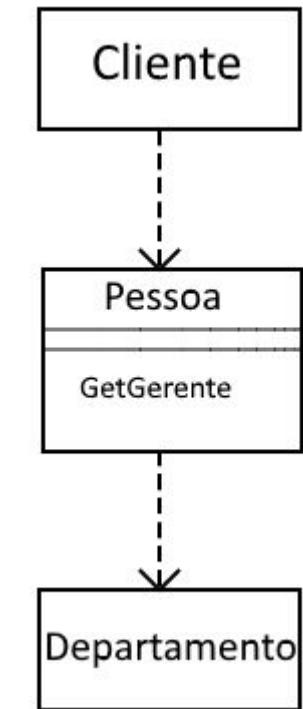
Cadeia de
Mensagens



Code Smells



Cadeia de
Mensagens





14) Hide Delegate (Esconder o delegado)

Uma classe cliente está invocando uma classe delegada de um objeto

Quando refatorar?

- Um objeto cliente "Obj1" qualquer invoca um objeto "Obj2" a partir de um outro objeto "Obj3", que por sua vez invoca um outro objeto "Obj4", e assim por diante.

Porém...

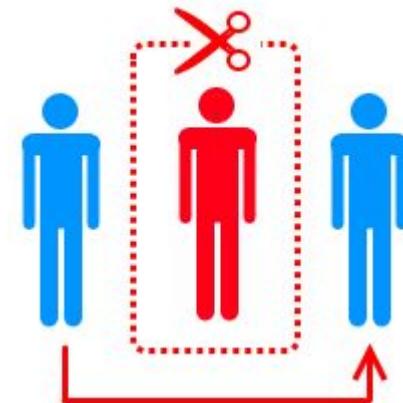
- Em excesso pode levar à criação de muitos intermediários

Mover Itens Entre Objetos



15) Remove Middle Man

(Remover o intermediário)





15) Remove Middle Man (Remover o intermediário)

A classe está fazendo uma delegação muito simples

Quando refatorar?

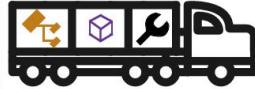
- Uma classe qualquer "C1" precisa de informações que estão numa classe "C2", mas em vez de fazer uma referência a esta, a classe "C1" obtém informações de "C2" através de chamadas a uma terceira classe intermediária "C3", e essa sim obtém as informações de C2 e por sua vez as devolve para C1.

Mover Itens Entre Objetos



- ~~10) Mover Método~~
- ~~11) Mover Campo~~
- ~~12) Extrair Classe~~
- ~~13) Incorporar Classe~~
- ~~14) Ocultar Delegado~~
- ~~15) Remover Intermediário~~
- 16) Introduzir Método Estrangeiro
- 17) Introduzir Extensão Local

Mover Itens Entre Objetos



16) Introduce Foreign Method (Introduzir método externo)





16) Introduce Foreign Method (Introduzir método externo)

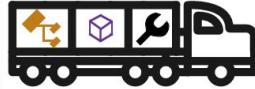
A classe-servidora que você está usando precisa de um novo método, mas você não pode modificar essa classe

Quando refatorar?

- Você tem código que consome dados e métodos de uma determinada classe (que chamamos de "classe utilitária"). Você percebe que esse código ficaria melhor dentro de um novo método dessa classe, mas você não pode fazer isso porque a classe faz parte de uma biblioteca de terceiros.

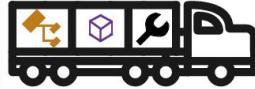
Porém...

- Se precisar criar vários métodos como esse, considere criar uma classe de extensão



17) Introduce Local Extension (Introduzir extensão local)





17) Introduce Local Extension

(Introduzir extensão local)

Uma classe-servidora que você está usando precisa de vários métodos adicionais, mas você não pode modificar essa classe

Quando refatorar?

- A classe que você está usando não possui os métodos que você precisa, e além disso você não pode adicionar esses métodos porque a classe faz parte de uma biblioteca de terceiros.

Porém...

- Se só houver um método, a classe pode ficar supérflua.