

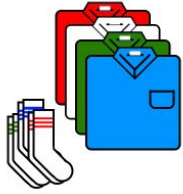


alura

Cursos Online de Tecnologia

C#: Refatorando 2 Conclusão

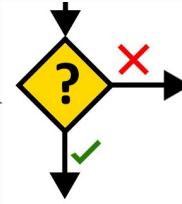
Parte 2



Organização de Dados



1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64



Simplificar Condições



Técnicas para Organização de Dados



18) Substituir Número Mágico



Problema

Número ou string solta no código

Solução

Utilizar uma constante

A menos que....

O número seja óbvio, como 0 ou 1

19) Substituir Array por Objeto



Problema

Tipos diferentes



Solução

Uma classe

Refatorações Similares

- Substituir Valor por Objeto

Elimina os odores:

- Obsessão por Tipos Primitivos

20) Substituir Dados por Objeto



Problema

Dados e comportamentos estão soltos

Solução

Criar nova classe

Refatorações Similares

Extraí Classe

Introduzir Objeto-Parâmetro

Substituir Array por Objeto

Substituir Método por Objeto-Método

Elimina os odores:

Código Duplicado

21) Encapsular Campo



Por que refatorar?

Encapsulamento
Funcionamento
Consistência

Elimina os odores:
Classe de Dados



22) Encapsular Coleção



Problema

Coleção desprotegida

Solução

Proteger coleção, criar add e remove

23) Mudar de Valor para Referência



Quando refatorar?

Instâncias idênticas

Você quer usar 1 único objeto

Refatoração inversa:

Mudar de Referência para Valor

24) Mudar de Referência para Valor



Objeto imutável como referência

Por que refatorar?

Refatoração inversa:

Mudar de Valor para Referência

25) Mudar Associação de Uni para Bidirecional



Problema

A referência B, mas B não referência A

Solução

Nova associação no sentido contrário

Refatoração Oposta

Mudar Associação de Bi para Unidirecional

26) Mudar Associação de Bi para Unidirecional



Problema

Podemos remover a referência em um dos sentidos

Refatoração Oposta

Mudar Associação de Uni para Bidirecional

27) Duplicar Dados Observados



Problema

Lógica do domínio na GUI

Solução

Crie classes para o domínio

Implementa o Design Pattern

Observer

Elimina os odores:

Classe Muito Grande

28) Substituir Código de Tipo por Classe



Problema

Tipo é identificado por números

Solução

Criar nova classe

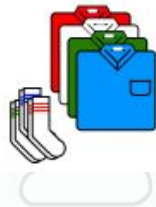
Refatorações Similares

Encapsular Campo

Elimina os odores:

Obsessão por Tipos Primitivos

29) Substituir Código de Tipo por Subclasses



Problema

Você tem um tipo de código imutável que afeta o comportamento da classe

Solução

Criar 1 Subclasse por Tipo

Elimina os odores:

Obsessão por Tipos Primitivos

Refatorações Similares

Substituir Código de Tipo por Classe

Substituir Código de Tipo por State/Strategy

Refatoração Oposta

Substituir Subclasse por Campos

30) Substituir Código de Tipo por State ou Strategy



Problema

Tipos MUTÁVEIS identificados por números. Cada tipo tem um comportamento diferente.

Solução

Substituir Código de Tipo por State/Strategy

Refatorações Similares

Substituir Código de Tipo por Classe

Substituir Código de Tipo por Subclasses

Implementa os Design Patterns

State

Strategy

Elimina os odores:

Obsessão por Tipos Primitivos

Comando Switch

31) Substituir Subclasses por Campo



Problema

Subclasses variam somente para retornar dados constantes

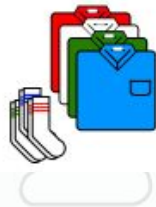
Solução

Substituir Subclasse por Campos

Refatoração Oposta

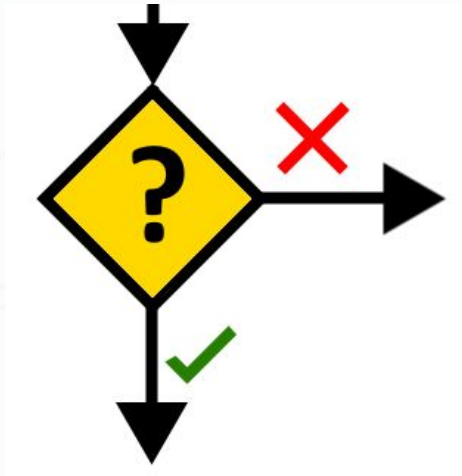
Substituir Tipo de Código por Subclasses

Qual técnica utilizar ?

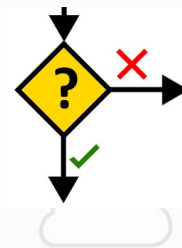


Tipos hoje são diferenciados por	Tipo é Mutável ?	Tipo afeta o comportamento da classe?	Aplicar técnica
int, string, etc	tanto faz	não	Substituir Código de Tipo por Classe
int, string, etc	não	sim	Substituir Código de Tipo por Subclasses
int, string, etc	sim	sim	Substituir Código de Tipo por State ou Strategy
subclasses	não	não	Substituir Subclasse por Campos

Técnicas para Simplificar Condições



32) Decompor Condição



Problema

Condição complexa

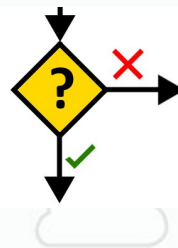
Solução

Cada bloco if, then, else, switch em um método separado

Elimina os odores:

Método Longo

33) Consolidar Expressão Condiciona



Problema

Condições diferentes com mesmo resultado

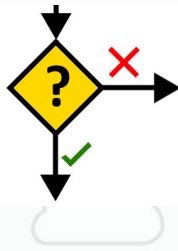
Solução

Extrair condições para um método

Elimina os odores:

Código Duplicado

34) Consolidar Fragmentos Condicionais Duplicados



Problema

Árvore condicional com código duplicado em seus ramos

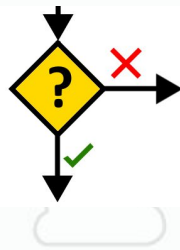
Solução

Mover o trecho duplicado para fora da árvore de condições

Elimina os odores:

Código Duplicado

35) Remover Flag de Controle



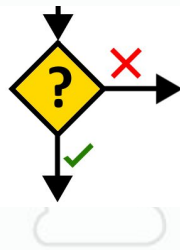
Problema

Variável como flag de controle booleana

Solução

Usar early return: return, break ou continue.

36) Substituir Condição Aninhada com Cláusulas de Guarda



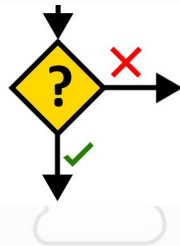
Problema

Condições dentro de condições

Solução

Condições excepcionais no início do método

37) Substituir Condição por Polimorfismo



Problema

Condições produzem resultados diferentes dependendo do tipo de código

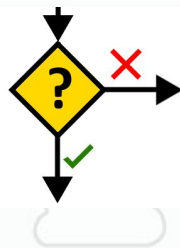
Solução

Uma subclasse encapsulando cada resultado diferente da condição

Elimina os odores:

Comando Switch

38) Introduzir Objeto Nulo



Problema

Checar objetos nulos o tempo todo

Solução

Objeto que encapsula comportamento se objeto == null

Refatorações semelhantes

Substituir Condições por Polimorfismo

Implementa o Design Pattern

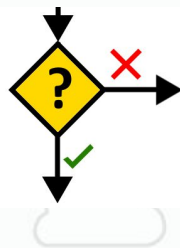
Objeto Nulo

Elimina os odores:

Comando Switch

Campo Temporário

39) Introduzir Asserção



Problema

Método pressupõe uma pré-condição que não está sendo verificada

Solução

Verificação o que foi pressuposto

Elimina os odores:

Comentários