PROVAS DE TI
TUDO PARA VOCÊ PASSAR

Refactoring

Rodrigo Macedo -
http://www.itnerante.com.br/profile/RodrigoMacedo

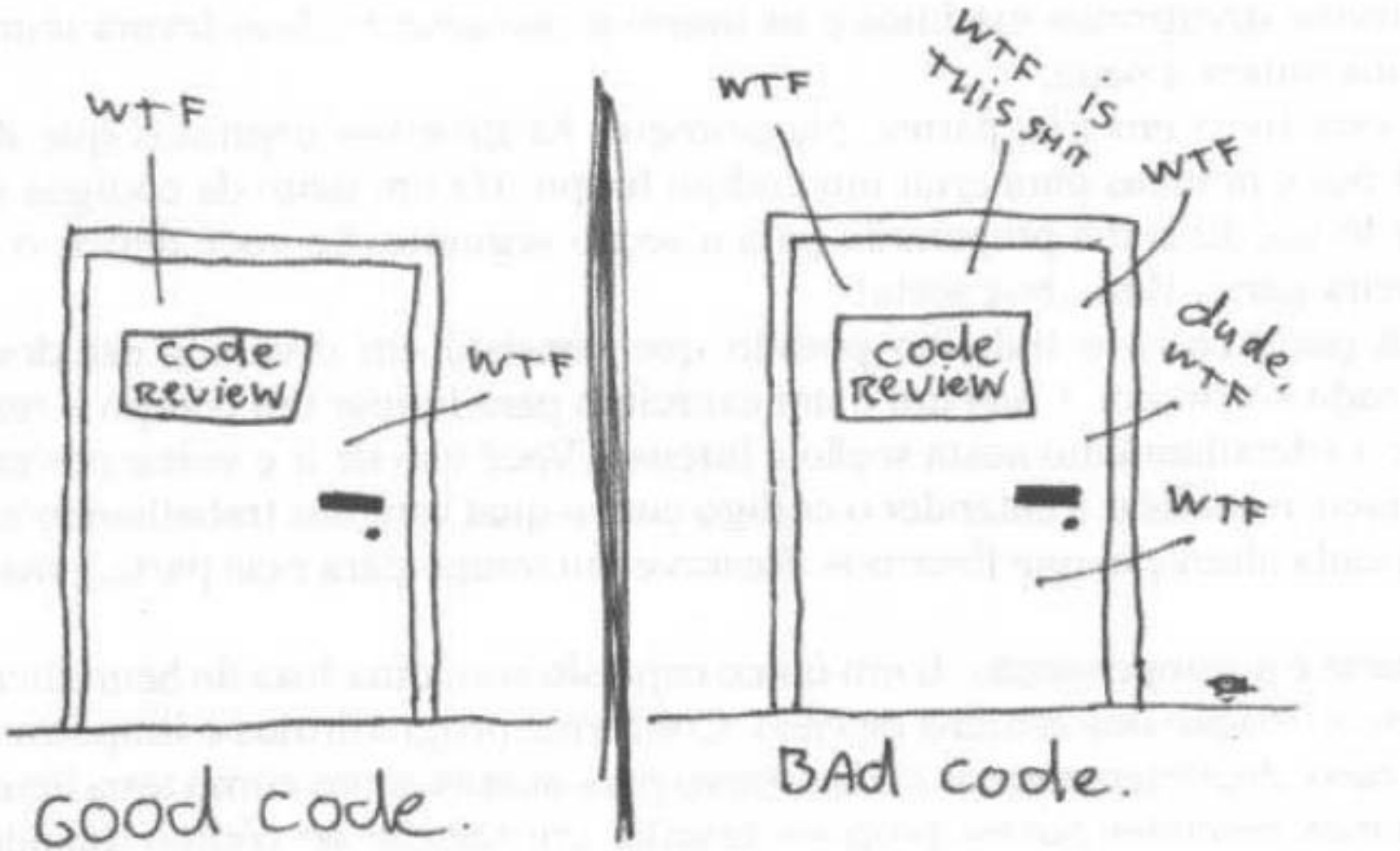
Escopo do Curso

- Conceituação Geral.
- Técnicas de Refatoração.
- Questões de Concursos.

Refactoring - Definições

Refactoring - Definições

Porta x Código



Refatoração

- Quais os passos para refatorar?
- Procure os “bad smells” (maus cheiros)

Exemplos de Bad Smell

- Duplicated Code (Código Duplicado)
- Long Method (Método Longo)
- Large Class (Classe Grande)
- Comments (Comentários)
- Long Parameter List (Lista de Parâmetros Longa)

Regra do Escoteiro

- Além de escrever código limpo, prezar para sempre **mantê-lo** limpo.
- “Deixe a área do acampamento mais limpa do que como você a encontrou”.

Técnica Extract Method

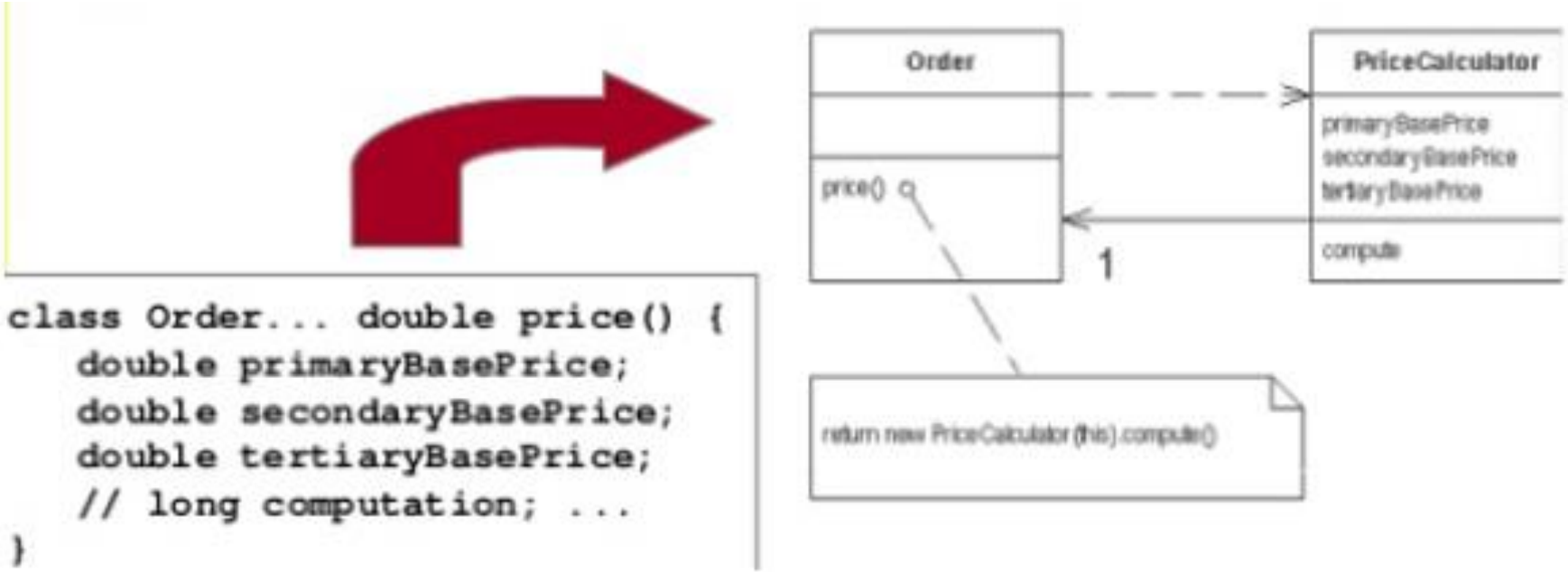
- Problema: Você tem um fragmento de código que pode ser agrupado em um método diferente.
- Solução: Transforme o fragmento em um método cujo nome explique sua funcionalidade.

```
void printOwing(int amount) {
   printBanner();
   System.out.println ("name: " + _name);
   System.out.println ("amount " + amount);
}
```

```
void printOwing() {
   printBanner();
   printDetails(amount);
}
void printDetails(double amount) {
   System.out.println ("name: " + _name);
   System.out.println ("amount " + amount);
}
```

Técnica Replace Method with Method Object

- Problema: Você tem um método longo que usa variáveis locais de tal maneira que não é possível aplicar Extract Method.
- Solução: Transforme o método em objeto para que todas as variáveis locais virem atributos.



Técnica Substitute Algorithm

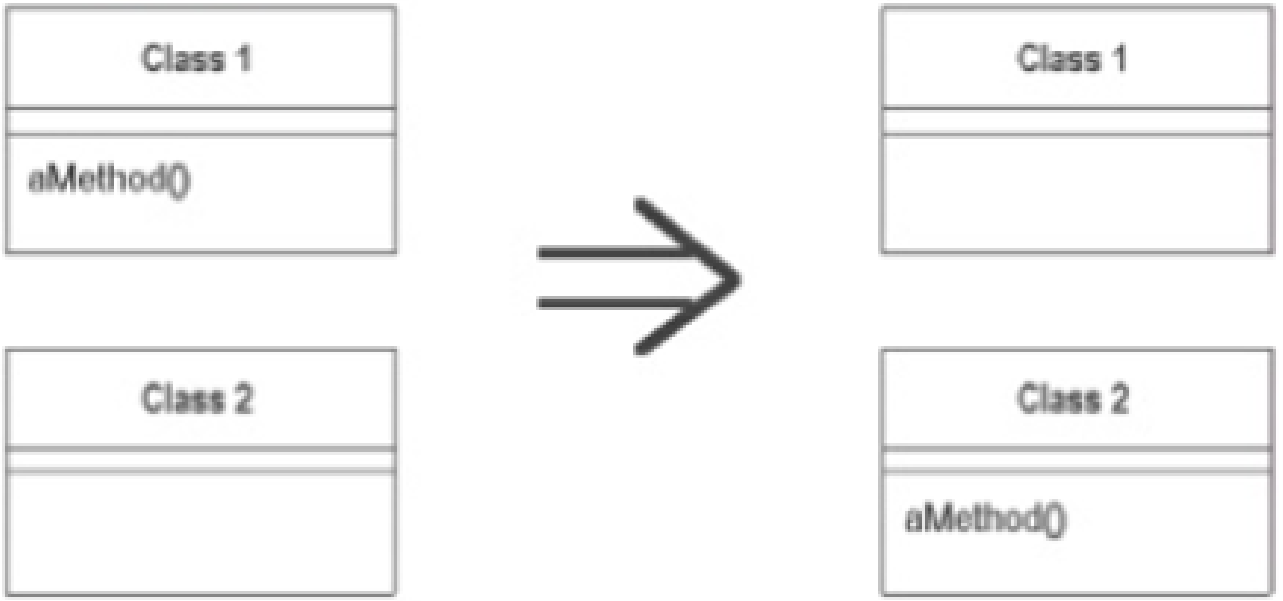
- Problema: Algoritmo que pode ser melhorado.
- Solução: Substitua o conteúdo do método com um algoritmo que faça a mesma coisa, mas seja melhor.

```
String foundPerson(String[] people){
  for (int i = 0; i < people.length; i++) {
    if (people[i].equals ("Don")){ return "Don"; }
    if (people[i].equals ("John")){ return "John"; }
    if (people[i].equals ("Kent")){ return "Kent"; }
  } return "";
}
```

```
String foundPerson(String[] people){
  List candidates =
    Arrays.asList(new String[] ("Don", "John", "Kent"));
    for (int i=0; i<people.length; i++)
      if (candidates.contains(people[i]))
        return people[i]; return "";
}
```

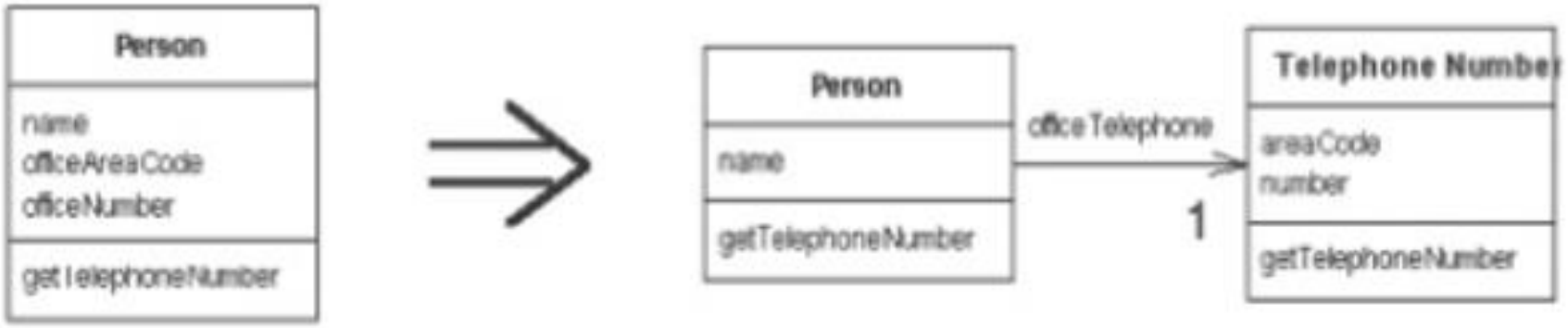
Técnica Move Method

- Problema: Um método é usado por mais recursos de outra classe que na classe em que foi definido.
- Solução: Crie um novo método com corpo similar na classe em que é usado. Transforme o antigo em simples delegação, ou elimine-o.



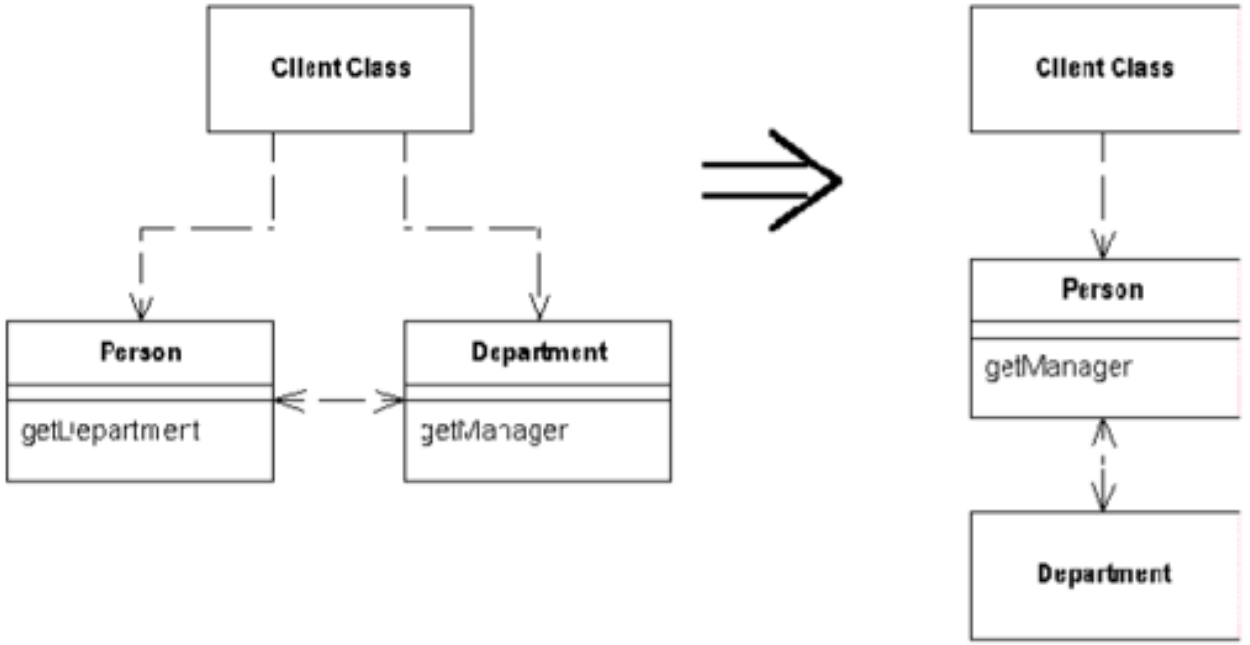
Técnica Extract Class

- Problema: Você tem uma classe fazendo trabalho que deveria ser feito por duas classes.
- Solução: Crie uma nova classe e transfira métodos e atributos relevantes para ela.



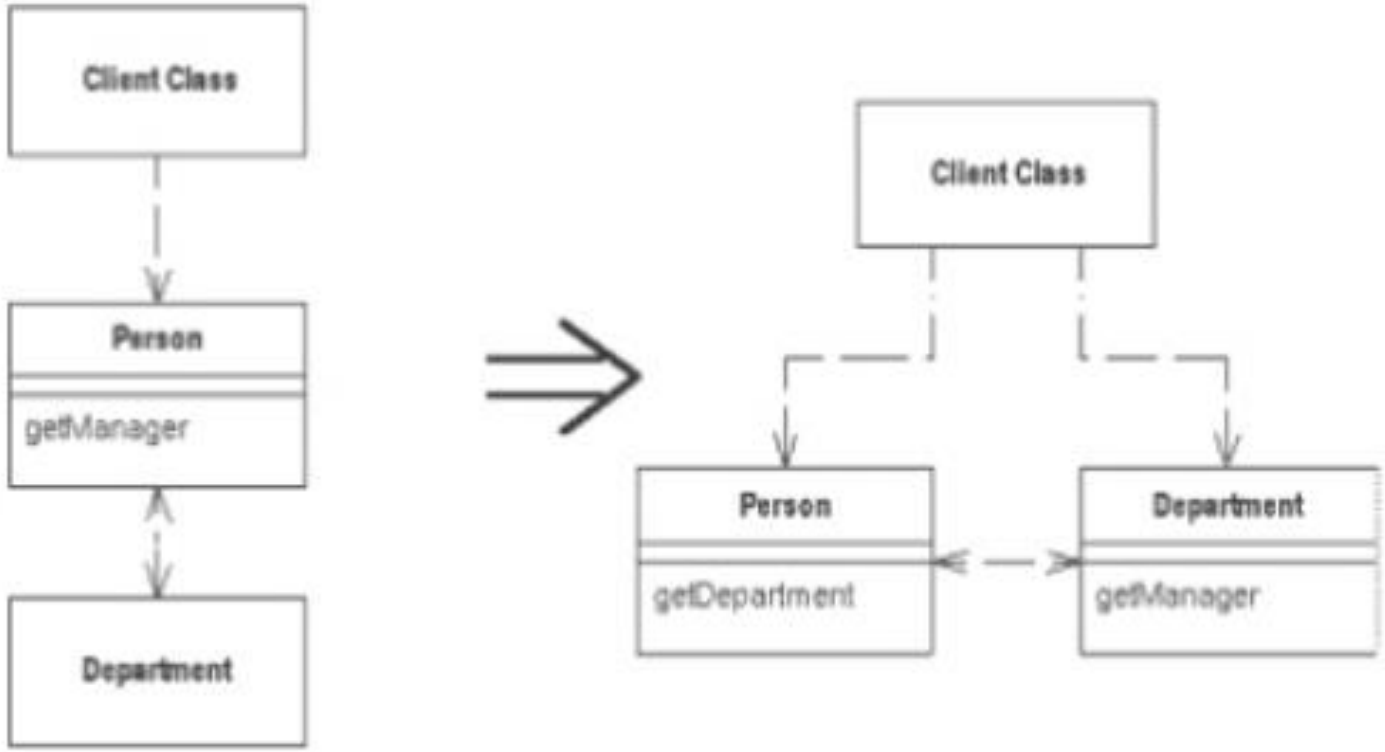
Técnica Hide Delegate

- Problema: Um cliente tem acesso e está chamando uma classe que é delegada pelo objeto que utiliza.
- Solução: Crie métodos no servidor para ocultar o objeto do acesso pelo cliente.



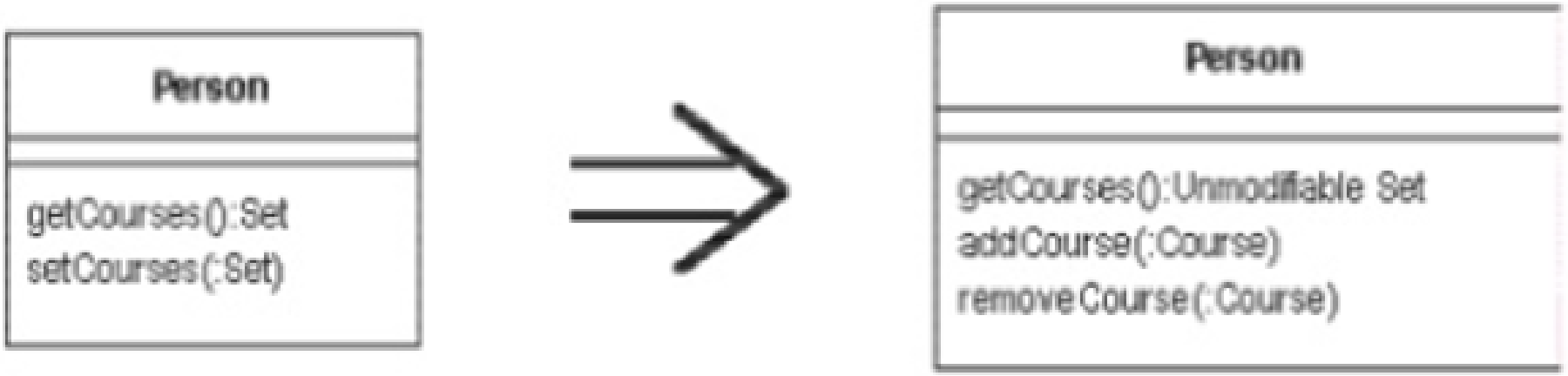
Técnica Remove Middle Man

- Problema: Uma classe está fazendo delegações simples em excesso.
- Solução: Faça o cliente delegar diretamente



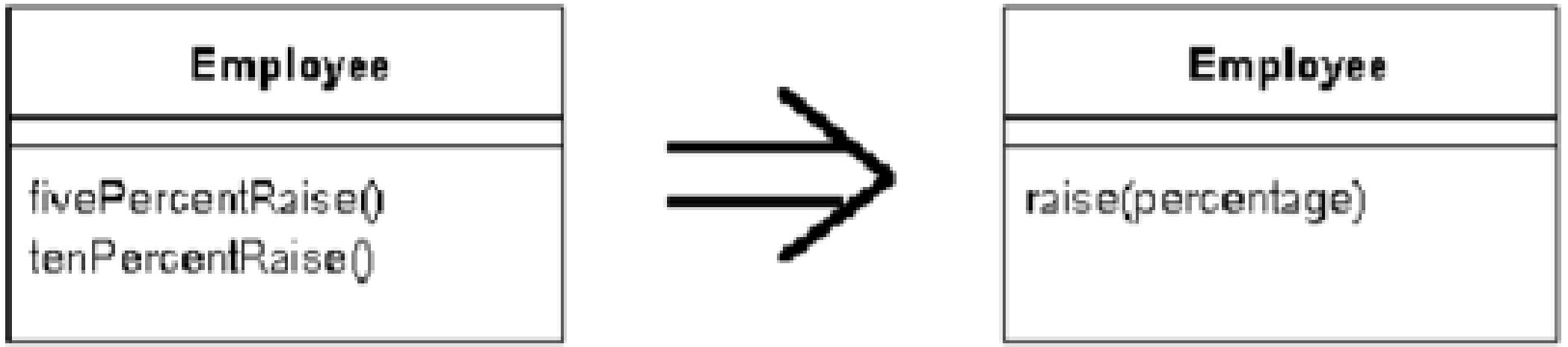
Técnica Encapsulate Collection

- Problema: Um método retorna uma coleção (List, Set) – acesso total e ausência de controle sobre tipos de dados.
- Solução: Faça-o retornar uma visão read-only dos dados e forneça métodos de adição e remoção.



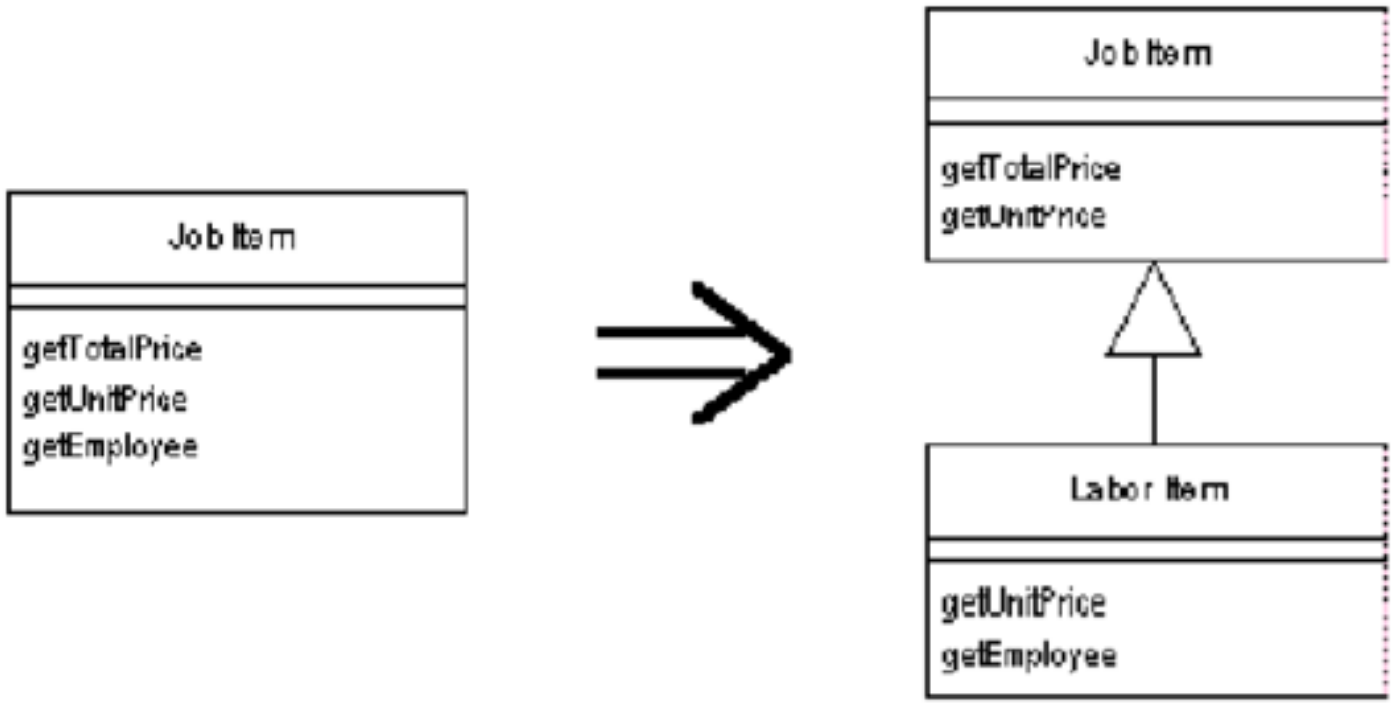
Técnica Parametrize Method

- Problema: Vários métodos fazem coisas similares mas com diferentes valores contidos no corpo do objeto.
- Solução: Crie um método que usa um parâmetro para os valores diferentes.



Técnica Extract Subclass

- Problema: Uma classe tem recursos que só são utilizados em algumas instâncias.
- Solução: Crie uma subclasse para esse conjunto de recursos.



Q1 – [CESPE TRE PE 2017] Refactoring é o processo que

a)implementa todas as funcionalidades da camada de *model* para depois implementar as camadas de *controller* e de *viewer*, nos casos em que a arquitetura M V C é utilizada.

b) efetua mudanças em um código existente e funcional sem alterar seu comportamento externo, com o objetivo de aprimorar a estrutura interna do código.

c) inclui funcionalidades extras no código, com o intuito de aprimorá-lo (r*ich source-code).*

d) aprimora a extração e o refinamento iterativo dos requisitos do produto ainda na fase de planejamento do *software,* sendo considerado um valor na X P *(extreme programming).*

e) estabelece os métodos, um após o outro, para depois definir as classes e suas abstrações e implementar as interfaces.

Q1 – [CESPE TRE PE 2017] Refactoring é o processo que

a) implementa todas as funcionalidades da camada de *model* para depois implementar as camadas de *controller* e de *viewer*, nos casos em que a arquitetura M V C é utilizada.

b) efetua mudanças em um código existente e funcional sem alterar seu comportamento externo, com o objetivo de aprimorar a estrutura interna do código.

c) inclui funcionalidades extras no código, com o intuito de aprimorá-lo (r*ich source-code).*

d) aprimora a extração e o refinamento iterativo dos requisitos do produto ainda na fase de planejamento do *software,* sendo considerado um valor na X P *(extreme programming).*

e) estabelece os métodos, um após o outro, para depois definir as classes e suas abstrações e implementar as interfaces.

Q2 –[CESPE ANAC 2009] A técnica conhecida como refactoring é constantemente aplicada no desenvolvimento baseado no método ágil extreme programming.

Q3 –[VUNESP TJM-SP 2017] A IDE NetBeans possui diversas funções de refatoração do código (*refactoring*). Esses recursos têm o objetivo de:

A)migrar o código desenvolvido para uma arquitetura específica para outra arquitetura suportada pela linguagem.

B)manter o histórico de alterações realizadas no código, permitindo que sejam revertidas caso necessário.

C)testar o funcionamento do código, evitando que a implantação de novas funcionalidades reintroduza erros já corrigidos.

D)evitar erros de sintaxe durante a digitação do código, oferecendo autocomplemento de palavras reservadas e nomes de classes à medida que são digitadas.

E)modificar a estrutura do código sem alterar o seu comportamento.

Q2 –[CESPE ANAC 2009] A técnica conhecida como refactoring é constantemente aplicada no desenvolvimento baseado no método ágil extreme programming. CERTO.

Q3 –[VUNESP TJM-SP 2017] A IDE NetBeans possui diversas funções de refatoração do código (*refactoring*). Esses recursos têm o objetivo de:

A)migrar o código desenvolvido para uma arquitetura específica para outra arquitetura suportada pela linguagem.

B)manter o histórico de alterações realizadas no código, permitindo que sejam revertidas caso necessário.

C)testar o funcionamento do código, evitando que a implantação de novas funcionalidades reintroduza erros já corrigidos.

D)evitar erros de sintaxe durante a digitação do código, oferecendo autocomplemento de palavras reservadas e nomes de classes à medida que são digitadas.

E)modificar a estrutura do código sem alterar o seu comportamento.

Q4 –[FGV AL-MT 2013] Segundo M. Fowler et al, na mecânica de aplicação da técnica de *Refactoring* chamada "*ExtractClass*", outras técnicas de *Refactoring* também precisarão ser usadas. Assinale a alternativa que as indica.

A)"*Hide Delegate*" e "*Remove Middle Man*"

B)"*Move Method*" e "*Move Field*"

C)"*Encapsulate Field*" e "*Encapsulate Downcast*"

D)"*Extract Subclass*" e "*Replace Inheritance with Delegation*"

E)"*Extract Subclass*" e "*Tease Apart Inheritance*"

Q5 –[CESPE STJ 2018] Acerca de TomCat, desenvolvimento seguro de *software, refactoring* e integração contínua, julgue o próximo item.

A refatoração de um código escrito em Delphi pode levar um método a ser separado e transformado em alguns outros métodos.

Q4 –[FGV AL-MT 2013] Segundo M. Fowler et al, na mecânica de aplicação da técnica de *Refactoring* chamada "*ExtractClass*", outras técnicas de *Refactoring* também precisarão ser usadas. Assinale a alternativa que as indica.

A)"*Hide Delegate*" e "*Remove Middle Man*"

B)"*Move Method*" e "*Move Field*"

C)"*Encapsulate Field*" e "*Encapsulate Downcast*"

D)"*Extract Subclass*" e "*Replace Inheritance with Delegation*"

E)"*Extract Subclass*" e "*Tease Apart Inheritance*"

Q5 –[CESPE STJ 2018] Acerca de TomCat, desenvolvimento seguro de *software, refactoring* e integração contínua, julgue o próximo item.

A refatoração de um código escrito em Delphi pode levar um método a ser separado e transformado em alguns outros métodos. CERTO.

Q6 –[CESPE TER-MT 2015] A técnica de *refactoring*

A) resulta em melhoramento do código, pois promove, a cada iteração, uma limpeza no código-fonte e alteração semântica da lógica.

B) apresenta a desvantagem de exigir uma nova alteração no código existente a cada novo requisito criado, mesmo na parte do código que tenha relação com esse requisito.

C) é usada para aumentar a complexidade do código a cada iteração, com o intuito de garantir e aumentar a segurança do código.

D) tem o objetivo de permitir grandes alterações no código a cada iteração no sentido de melhorar a qualidade do código.

E) permite a reestruturação de códigos de maneira disciplinada e compatível com os métodos ágeis e é uma prática fundamental da XP (*eXtreme Programming*).

Q6 –[CESPE TER-MT 2015] A técnica de *refactoring*

A) resulta em melhoramento do código, pois promove, a cada iteração, uma limpeza no código-fonte e alteração semântica da lógica.

B) apresenta a desvantagem de exigir uma nova alteração no código existente a cada novo requisito criado, mesmo na parte do código que tenha relação com esse requisito.

C) é usada para aumentar a complexidade do código a cada iteração, com o intuito de garantir e aumentar a segurança do código.

D) tem o objetivo de permitir grandes alterações no código a cada iteração no sentido de melhorar a qualidade do código.

E) permite a reestruturação de códigos de maneira disciplinada e compatível com os métodos ágeis e é uma prática fundamental da XP (*eXtreme Programming*).

Q7 –[CESPE STF 2013] A respeito da técnica de refactoring, julgue o item seguinte.
O refactoring aprimora o design de um software, reduz a complexidade da aplicação, remove redundâncias desnecessárias, reutiliza código, otimiza o desempenho e evita a deterioração durante o ciclo de vida de um código.

Q8 – [VUNESP TJM-SP 2017] A IDE NetBeans possui diversas funções de refatoração do código (refactoring). Esses recursos têm o objetivo de

a) migrar o código desenvolvido para uma arquitetura específica para outra arquitetura suportada pela linguagem.
b) manter o histórico de alterações realizadas no código, permitindo que sejam revertidas caso necessário.
c) testar o funcionamento do código, evitando que a implantação de novas funcionalidades reintroduza erros já corrigidos.
d) evitar erros de sintaxe durante a digitação do código, oferecendo autocomplemento de palavras reservadas e nomes de classes à medida que são digitadas.
e) modificar a estrutura do código sem alterar o seu comportamento.

Q7 –[CESPE STF 2013] A respeito da técnica de refactoring, julgue o item seguinte.

O refactoring aprimora o design de um software, reduz a complexidade da aplicação, remove redundâncias desnecessárias, reutiliza código, otimiza o desempenho e evita a deterioração durante o ciclo de vida de um código. CERTO.

Q8 – [VUNESP TJM-SP 2017] A IDE NetBeans possui diversas funções de refatoração do código (refactoring). Esses recursos têm o objetivo de

a) migrar o código desenvolvido para uma arquitetura específica para outra arquitetura suportada pela linguagem.

b) manter o histórico de alterações realizadas no código, permitindo que sejam revertidas caso necessário.

c) testar o funcionamento do código, evitando que a implantação de novas funcionalidades reintroduza erros já corrigidos.

d) evitar erros de sintaxe durante a digitação do código, oferecendo autocomplemento de palavras reservadas e nomes de classes à medida que são digitadas.

e) modificar a estrutura do código sem alterar o seu comportamento.

Q9–[CESPE ANATEL 2014] No que se refere a práticas ágeis, julgue o item.
Para se evitar dispêndio de muito tempo na alteração do código e pouco tempo na programação de novas funcionalidades, a prática de *refactoring* deve restringir-se aos casos em que haja grandes porções de código a ser refatorado.

Q10 –[COMPERVE UFRN 2018] Considere a situação em que uma classe A é superclasse das classes B e C e que, tanto B quanto C possuem um método M com a mesma assinatura e código. Nessa situação, a operação de refatoração (*refactoring*) de código mais apropriada a ser aplicada é

A)Extract module.

B)Pull up method.

C)Extract method.

D)Inline method.

Q9–[CESPE ANATEL 2014] No que se refere a práticas ágeis, julgue o item.

Para se evitar dispêndio de muito tempo na alteração do código e pouco tempo na programação de novas funcionalidades, a prática de *refactoring* deve restringir-se aos casos em que haja grandes porções de código a ser refatorado. ERRADO.

Q10 –[COMPERVE UFRN 2018] Considere a situação em que uma classe A é superclasse das classes B e C e que, tanto B quanto C possuem um método M com a mesma assinatura e código. Nessa situação, a operação de refatoração (*refactoring*) de código mais apropriada a ser aplicada é

A)Extract module.

B)Pull up method.

C)Extract method.

D)Inline method.

Q11 –[CESPE SLU-DF 2019] Com relação a desenvolvimento de *software*, julgue o item a seguir.
Refactoring (refatoração) é o processo utilizado para reescrever aplicações desatualizadas, com a finalidade de incrementar e melhorar suas funcionalidades; o uso dessa técnica normalmente aprimora aplicações para disponibilizá-las na Internet.

Q12 –[CESPE ANATEL 2014] No que se refere a práticas ágeis, julgue o item.
O fato de cada código ter sua própria estrutura não constitui impedimento para se estabelecerem padrões para a prática de *refactoring*, sendo possível, por exemplo, utilizar padrões de projeto, como o *factory method*, para remover duplicações no código.

Q11 –[CESPE SLU-DF 2019] Com relação a desenvolvimento de *software*, julgue o item a seguir.

Refactoring (refatoração) é o processo utilizado para reescrever aplicações desatualizadas, com a finalidade de incrementar e melhorar suas funcionalidades; o uso dessa técnica normalmente aprimora aplicações para disponibilizá-las na Internet. ERRADO.

Q12 –[CESPE ANATEL 2014] No que se refere a práticas ágeis, julgue o item.

O fato de cada código ter sua própria estrutura não constitui impedimento para se estabelecerem padrões para a prática de *refactoring*, sendo possível, por exemplo, utilizar padrões de projeto, como o *factory method*, para remover duplicações no código. CERTO.

Q13 –[IBFC EMBASA 2017] Quanto à refatoração (*refactoring*), analise as afirmativas abaixo e assinale a alternativa correta.

I. Quando o prazo de entrega está próximo do final, deve-se fazer a refatoração.

II. Quando o código está instável, e não funciona, é o melhor momento para fazer a refatoração.

III. *EXtreme Programming* tem refatoração como uma de suas práticas.

IV. É melhorar a estrutura do código de um sistema preservando as suas funcionalidades.

Estão corretas as afirmativas:

A)Somente a I e II

B)Somente a II e III

C)Somente a III e IV

D)Somente a I e III

Q13 –[IBFC EMBASA 2017] Quanto à refatoração (*refactoring*), analise as afirmativas abaixo e assinale a alternativa correta.

I. Quando o prazo de entrega está próximo do final, deve-se fazer a refatoração.

II. Quando o código está instável, e não funciona, é o melhor momento para fazer a refatoração.

III. *EXtreme Programming* tem refatoração como uma de suas práticas.

IV. É melhorar a estrutura do código de um sistema preservando as suas funcionalidades.

Estão corretas as afirmativas:

A)Somente a I e II

B)Somente a II e III

C)Somente a III e IV

D)Somente a I e III

Q14–[COPERVE UFSC 2018] Considere os seguintes exemplos de procedimentos de manutenção, no contexto da necessidade de alteração de um programa hipotético de controle acadêmico de cursos de graduação da UFSC:

I. fazer com que o resultado da matrícula passe a ter a opção de gerar o resultado em formato PDF, além da atual possibilidade de informar na tela;

II. incluir funcionalidade para permitir que o trancamento de matrícula possa ser feito on-line;

III. reorganização da hierarquia de herança das classes do programa;

IV. criar classes no programa;

V. remover classes do programa;

Assinale a alternativa que relaciona apenas procedimentos de manutenção que podem ser classificados como ações de refatoração (refactoring).

A)III, IV e V.

B)I e II.

C)I, III e IV.

D)II, III e IV.

E)I, II e V.

Q14–[COPERVE UFSC 2018] Considere os seguintes exemplos de procedimentos de manutenção, no contexto da necessidade de alteração de um programa hipotético de controle acadêmico de cursos de graduação da UFSC:

I. fazer com que o resultado da matrícula passe a ter a opção de gerar o resultado em formato PDF, além da atual possibilidade de informar na tela;

II. incluir funcionalidade para permitir que o trancamento de matrícula possa ser feito on-line;

III. reorganização da hierarquia de herança das classes do programa;

IV. criar classes no programa;

V. remover classes do programa;

Assinale a alternativa que relaciona apenas procedimentos de manutenção que podem ser classificados como ações de refatoração (refactoring).

A)III, IV e V.

B)I e II.

C)I, III e IV.

D)II, III e IV.

E)I, II e V.

Q15–[IBFC TRE-AM 2014] Processo de modificar um sistema de software para melhorar a estrutura interna do código sem alterar seu comportamento externo, evitando a deterioração tão comum durante o ciclo de vida de um código, é denominado:

A)Refatoração (Refactoring).

B)Padrões de projeto (Design Patterns).

C)Design Simples (Simple Design).

D)Padrões de Arquitetura de Aplicações Corporativas (Patterns of Enterprise Applications Architecture).

Q16 - [UFG 2014] No desenvolvimento orientado a testes, a refatoração compreende

a) a remoção de duplicação causada pela adição de novas funcionalidades.

b) a criação de um teste e fazê-lo falhar.

c) a escrita de código o suficiente para que o mesmo compile com sucesso.

d) a escrita de código de produção para fazer o teste passar.

e) a criação de código para requisitos futuros.

Q15–[IBFC TRE-AM 2014] Processo de modificar um sistema de software para melhorar a estrutura interna do código sem alterar seu comportamento externo, evitando a deterioração tão comum durante o ciclo de vida de um código, é denominado:

A)Refatoração (Refactoring).

B)Padrões de projeto (Design Patterns).

C)Design Simples (Simple Design).

D)Padrões de Arquitetura de Aplicações Corporativas (Patterns of Enterprise Applications Architecture).

Q16 - [UFG 2014] No desenvolvimento orientado a testes, a refatoração compreende

a) a remoção de duplicação causada pela adição de novas funcionalidades.

b) a criação de um teste e fazê-lo falhar.

c) a escrita de código o suficiente para que o mesmo compile com sucesso.

d) a escrita de código de produção para fazer o teste passar.

e) a criação de código para requisitos futuros.

Q17 - [FCC TRE RN 2011] Em relação à Refatoração, é INCORRETO afirmar:

a) a existência ou não de testes automatizados no software é irrelevante .

b) utiliza fortemente conceitos de orientação a objeto.

c) melhora a estrutura interna do código sem alterar seu comportamento externo.

d) evita a deterioração durante o ciclo de vida de um código.

e) incrementa melhoria no entendimento do código.

Q18 - [CESPE INPI 2013]

A refatoração objetiva tornar o código mais claro e limpo.

Q17 - [FCC TRE RN 2011] Em relação à Refatoração, é INCORRETO afirmar:

a) a existência ou não de testes automatizados no software é irrelevante .

b) utiliza fortemente conceitos de orientação a objeto.

c) melhora a estrutura interna do código sem alterar seu comportamento externo.

d) evita a deterioração durante o ciclo de vida de um código.

e) incrementa melhoria no entendimento do código.

Q18 - [CESPE INPI 2013]

A refatoração objetiva tornar o código mais claro e limpo. CERTO.

Q19 - [CESGRANRIO BACEN 2010] Um programador inexperiente gerou, em determinado sistema, código-fonte de baixa qualidade que foi analisado e reprovado por ferramentas automatizadas de auditoria de código. Considerando-se que existem muitos códigos duplicados nesse sistema, que técnicas de refatoração são

indicadas nessa situação?

a) Introduce Parameter Object e Move Method.

b) Duplicate Observed Data e Replace Inheritance with Delegation.

c) Move Method e Move Field.

d) Extract Method e Substitute Algorithm.

e) Remove Double Negative e Introduce Parameter Object.

Q19 - [CESGRANRIO BACEN 2010] Um programador inexperiente gerou, em determinado sistema, código-fonte de baixa qualidade que foi analisado e reprovado por ferramentas automatizadas de auditoria de código. Considerando-se que existem muitos códigos duplicados nesse sistema, que técnicas de refatoração são

indicadas nessa situação?

a) Introduce Parameter Object e Move Method.

b) Duplicate Observed Data e Replace Inheritance with Delegation.

c) Move Method e Move Field.

d) Extract Method e Substitute Algorithm.

e) Remove Double Negative e Introduce Parameter Object.

GABARITO

Q1 – LETRA B.
Q2 - CERTO.
Q3 – LETRA E.
Q4 - LETRA B.
Q5 - CERTO.
Q6 - LETRA E.
Q7 – CERTO.
Q8 – LETRA E.
Q9 - ERRADO.
Q10 - LETRA B.
Q11 – ERRADO.
Q12 - CERTO.
Q13 - LETRA C.
Q14 – LETRA A.
Q15 – LETRA A.
Q16 - LETRA A.
Q17 – LETRA A.
Q18 – CERTO.
Q19 - LETRA D.