



High performance. Delivered.

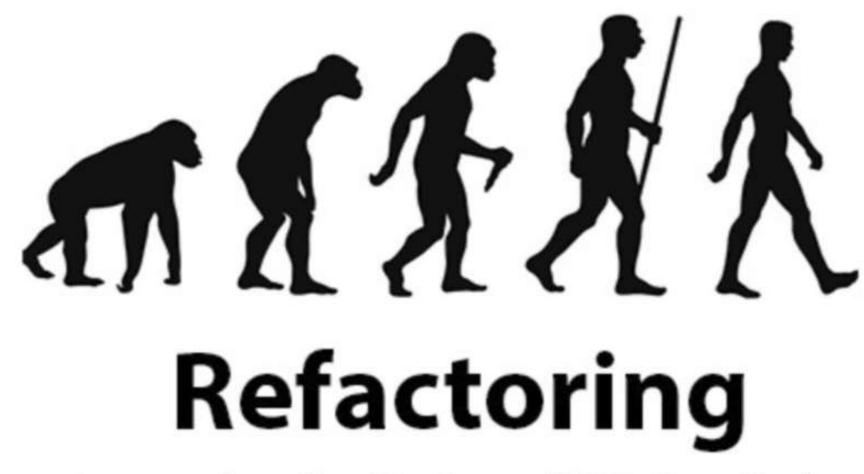
Application Delivery Fundamentals: Java

Module 14: Refatoração

•

2f4869676820706572666f726d616c63652c2044656c6976657265642c2748696768207065726667726667726667726667726667726667726667726667726667726667726667726667726667726667726772677

### "REFATORAÇÃO"

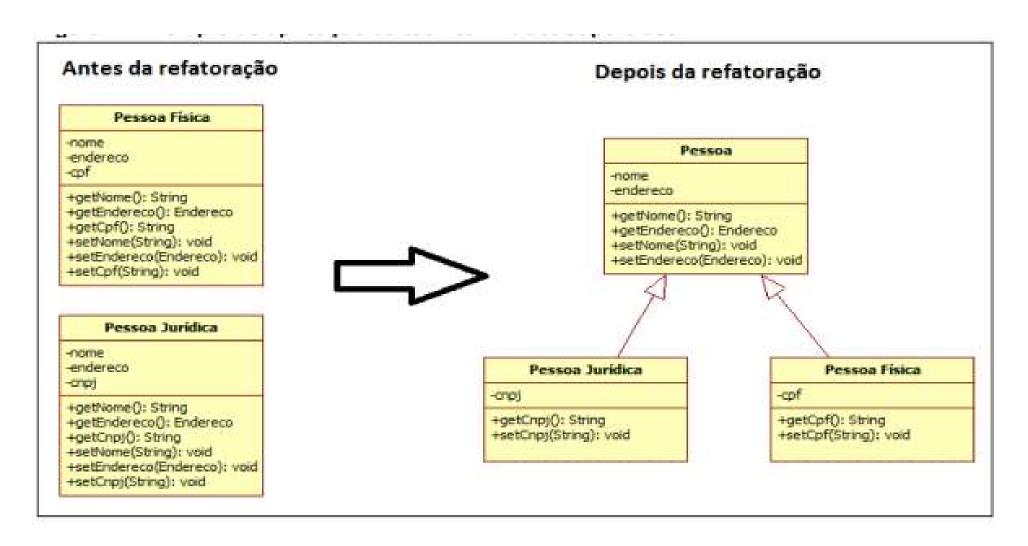


Improving the Design of Existing Code

È uma forma disciplinada de reestruturar o código quando pequenas mudanças são feitas nele para melhorar o design sem mudar o comportamento da aplicação;

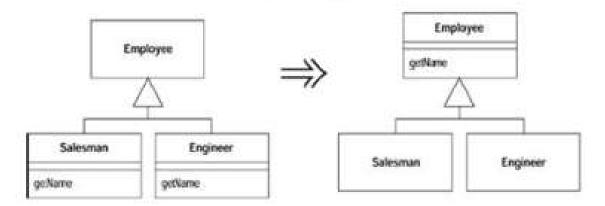
- A refatoração melhora o design do software;
- A refatoração torna o software mais fácil de manter;
- A refatoração lhe ajuda a encontrar defeitos;
- A refatoração lhe ajuda a programar mais rápido;

### Extract SuperClass



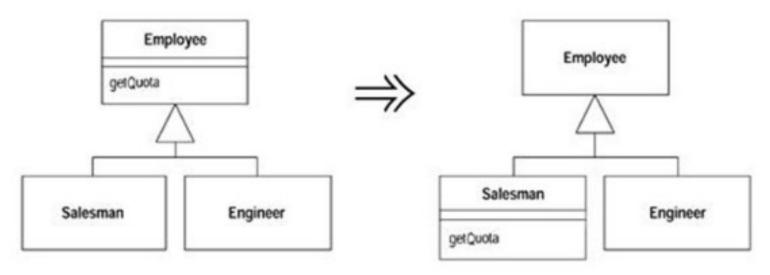
### Pull Up Método

- Funcionalidade:
  - Move métodos que são idênticos de subclasses para a superclasse.
- Motivação:
  - Remover código duplicado;
  - · Prevenir bugs quando apenas um método é alterado.



### Push Down Método

- Funcionalidade:
  - Move o método da superclasse para uma subclasse.
- Motivação:
  - Um método de uma superclasse é relevante apenas para parte das subclasses.



Copyright © 202(

- Nome: Replace Temp with Query
- Resumo: Uma variável local está sendo usada para guardar o resultado de uma expressão. Troque as referências a esta expressão por um método.
- Motivação: Variáveis temporárias encorajam métodos longos (devido ao escopo). O código fica mais limpo e o método pode ser usado em outros locais.

## Replace Temp with Query

```
double getPreco() {
   int precoBase = _quantidade * _precoItem;
   double fatorDesconto;
   if (precoBase > 1000) fatorDesconto = 0.95;
   else fatorDesconto = 0.98;
   return precoBase * fatorDesconto;
}
```

# Replace Temp with Query (120) resultado final...

```
double getPreco() {
   return precoBase() * fatorDesconto();
private int fatorDesconto() {
    if (precoBase() > 1000)
        return 0.95;
   return 0.98;
private int precoBase() {
    return _quantidade * _precoItem;
```

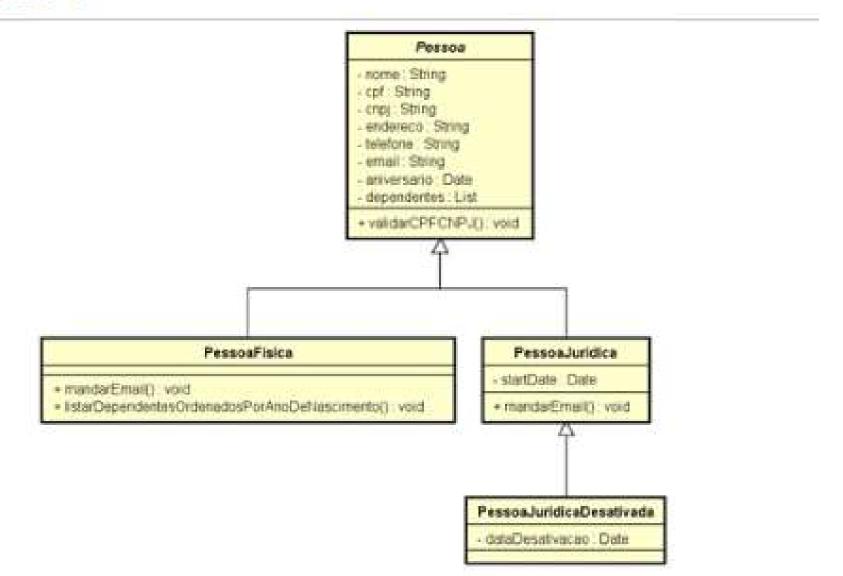
#### **Perguntas:**

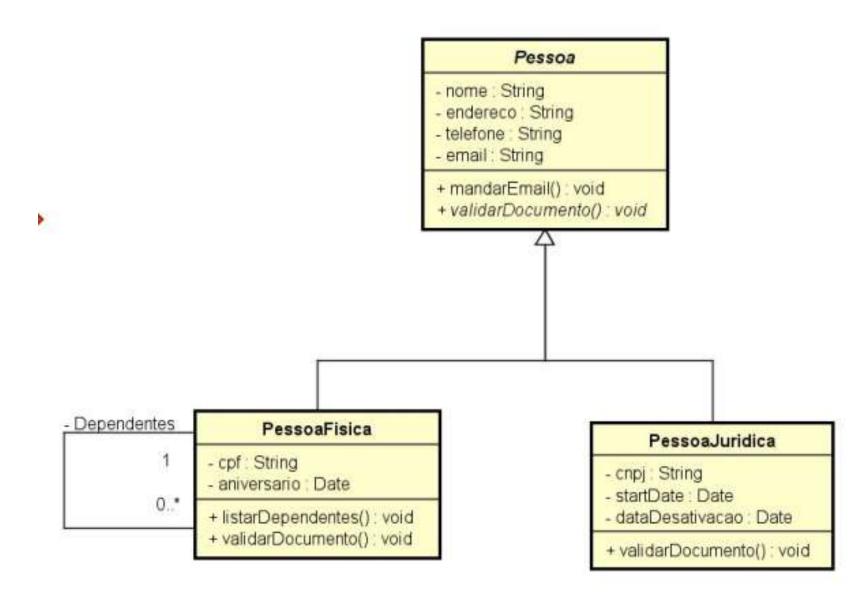
### O que é Refatoração?

São mudanças no código para melhorar o design sem mudar o comportamento da aplicação;

https://medium.com/studio-oceano/guia-definitivo-para-a-refatora%C3%A7%C3%A3o-568cc6625eaf

### Exercício





```
public static String DetermineCor(int entrada) {
       String cor = new String;
         If (entrada==0) {
                                                                  REFATORAÇÃO
           cor = "azul".
         } else if (entrada == 1) {
          cor = "verde"
         } else {
          cor = "desconhecida"
         return cor;
                                                     public static String DetermineCor(int entrada) {
                                                         String cor = new String;
                                                           If (entrada==0) return "azul";
public static String DetermineCor(int entrada) {
                                                          if (entrada == 1) return "verde";
   String cor = new String;
                                                          return "desconhecida";
     If (entrada<0) throw new ArgmentException();
     if (entrada>1) throw new ArgmentException(); }
     return entrada == 0 ? "azul" : "verde":
```





Refatorar o Jogo de Xadrez;

- Criar a classe Abstrata Peça;
  - Método MOVER;
- Criar a classe PEAO;
   Herdar o atributo MOVER;
   Implementar o movimento;

#### **Questions and Comments**

 What questions or comments do you have?



