

Qualidade e Testes De Software

Professor Gilmar Luiz de Borba

Prática

Refatoração Parte 1

Refatoração de extração de constantes, variáveis locais e métodos.

Objetivos:

- Praticar a partir de um exemplo, a refatoração de extração de constantes, variáveis locais e métodos .

Considerações:

- A aplicação tem como objetivo calcular o valor de 3 mensalidades, considerando no exercício de um ano, baseadas no mês de vencimento da mesma. Caso tenha vencido (independente do mês), será cobrado juros de 10%.
- Dados de entrada: valor da mensalidade, mês (um número inteiro)
- Dados de saída:

IHM (Interface Homem Máquina)

```
Problems @ Javadoc Declaration Console
ExtrairMetodo [Java Application] C:\Program Files (x86)\Java\jdk1.7.0_45\bin\javaw.exe (10/03/2015 13:06:34)

Informe o valor da mensalidade:
1000
Informe o mês de referência:
3
RESULTADO
Mês atual: 3
Mês de referência: 3
Mensalidade: 1000.0
Nova mensalidade: 1000.0

Informe o valor da mensalidade:
1000
Informe o mês de referência:
2
RESULTADO
Mês atual: 3
Mês de referência: 2
Mensalidade: 1000.0
Nova mensalidade: 1100.0

c:\gilmar_borba\qualidade\imagens\
```

(1) Crie um novo projeto JAVA: **RefatoracaoParte1**

(2) Crie um pacote: meuPackage

(3) Crie a classe **Refatoracao** (com o método main())

(4) Incluir o pacote Java.util (iremos usar as classes Scanner e Calendar)

(5) Seguir o exemplo

Qualidade e Testes De Software

Professor Gilmar Luiz de Borba

Ao final da digitação, salve o projeto e execute o programa.

```
1 package br.com.borba.luiz.gilmar.prof;
2 import java.util.Calendar;
3 import java.util.*;
4 public class Refatoracao {
5     static Calendar calendario;
6     static Scanner entrada;
7     static public void main(String[] arguments) {
8         double mensalidade = 0;
9         double novaMensalidade = 0;
10        int mes = 0;
11        int mesAtual = 0;
12        entrada = new Scanner(System.in);
13        int contador = 1;
14        while(contador<=3) {
15            System.out.println("\nValor da mensalidade: ");
16            mensalidade = entrada.nextInt();
17            System.out.println("\nMês de referência: ");
18            mes = entrada.nextInt();
19            calendario = Calendar.getInstance();
20            mesAtual = calendario.get(Calendar.MONTH)+1;
21            if (mes < mesAtual) novaMensalidade = mensalidade * 1.10;
22            else if (mes >= mesAtual) novaMensalidade = mensalidade;
23            System.out.println("RESULTADO");
24            System.out.println("Mês atual: "+mesAtual);
25            System.out.println("Referência: "+mes);
26            System.out.println("Mensalidade: "+mensalidade);
27            System.out.println("Novo valor: "+novaMensalidade);
28            contador++;
29        } // while
30        System.out.println("PROGRAMA FINALIZADO!");
31    } // main
32 }
```

c:\prof_Gilmar_Borba\Programacao\imagens\

REFATORAÇÕES

(1) Faça a refatoração de endentação (identação) do código.

No Eclipse Ctrl + Shift + F, **No Netbeans** Alt + Shift + F

(2) Refatorar o código de modo a extrair uma constante para o valor máximo da iteração (3).
Criar a constante: QUANTIDADE

(3) Refatorar o código de modo a extrair um método do trecho de código:

```
if (mes < mesAtual)
    novaMensalidade = mensalidade * 1.10;
else if (mes == mesAtual )
    novaMensalidade = mensalidade;
```

Criar o método: **verificarValorMensalidade**

Qualidade e Testes De Software

Professor Gilmar Luiz de Borba

(4) Refatorar o código de modo a extrair uma variável local para o cálculo da mensalidade:

`mensalidade * 1.10`

Criar a variável local: **valorCalculadoMensalidade**

(5) Refatorar o código de modo a extrair um método do trecho de código:

```
System.out.println("RESULTADO");  
System.out.println("Mês atual: "+mesAtual);  
System.out.println("Referência: "+mes);  
System.out.println("Mensalidade: "+mensalidade);  
System.out.println("Novo valor: "+novaMensalidade);
```

Criar o método: **imprimirResultados**

(6) Execute a aplicação, confirme que após todas as alterações o programa continua com o mesmo comportamento externo.

Qualidade e Testes De Software

Professor Gilmar Luiz de Borba

QUESTÕES

(01) Explique o código: `calendario = Calendar.getInstance();`

(02) O que é um método estático?

(03) Em termos de coesão os métodos gerados após a refatoração foram devidamente bem implementados? Justifique.

(05) Quando deve ser usada a refatoração de extração de método?

(06) É possível refatorar um método existente criando outro(s) métodos(s) a partir deste? Quando isso deve ser implementado?

(07) Por que refatorar trechos de códigos de modo a criar constantes e variáveis locais?

::fim