

# Qualidade e Testes De Software

Professor Gilmar Luiz de Borba

## Prática

Refatoração de extração de constantes, variáveis locais e métodos.

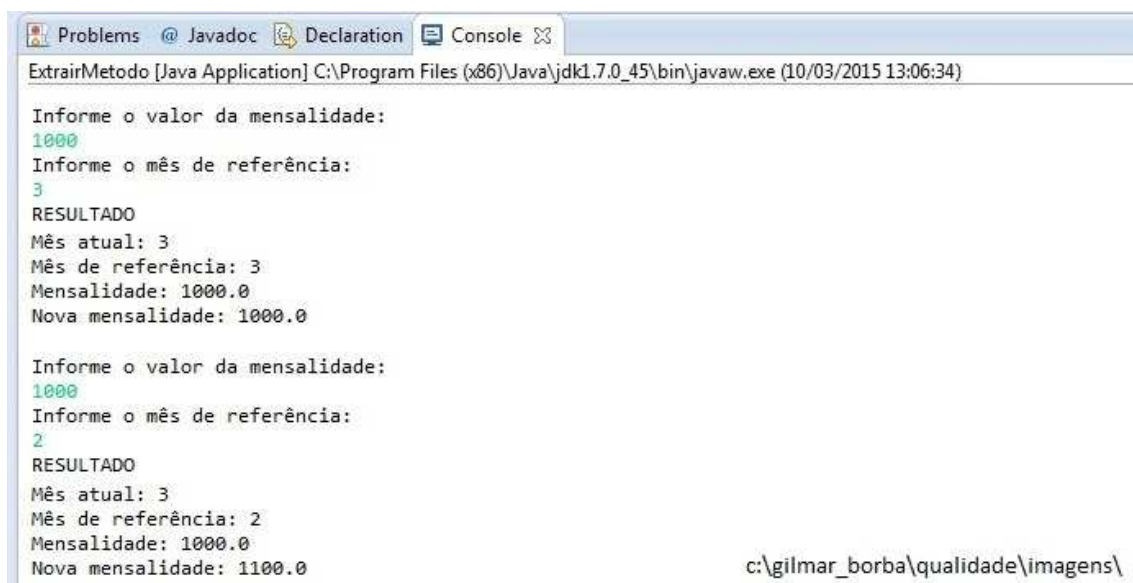
### Objetivos:

- Praticar a partir de um exemplo simples, 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)



```
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** (Crie o método main() para esse projeto)
- (2) Crie um pacote: pkgRefatoracaoParte1
- (3) Crie a classe **Refatoracao**. (com o método main()). Incluir o pacote Java.util. Veja a imagem:
- (4) Incluir o pacote Java.util (iremos usar as classes Scanner e Calendar)
- (5) Criar as seguintes variáveis:  
  
mensalidade (double)  
novaMensalidade (double)  
mes (inteiro)  
mesAtual (inteiro)

# Qualidade e Testes De Software

Professor Gilmar Luiz de Borba

Podem ser variáveis de instância (campos) da classe ou variáveis locais, a sua escolha.

(6) Criar também um campo (ou variável local) de nome “calendário” do tipo (classe) *Calendar*. A classe *Calendar* encontra-se no pacote *java.util*.

```
package pkgRefatoracaoPartel;
import java.util.Calendar;
import java.util.Scanner;
public class Refatoracao {

    static Calendar calendario;
    static Scanner entrada;

    static public void main(String[] arguments) {
        double mensalidade=0;
        double novaMensalidade=0;
        int mes=0;
        int mesAtual=0;
        entrada = new Scanner(System.in);

        } // main
    } // class
```

(7) No método *main()*, criar o mecanismo para permitir a entrada de dados, use um objeto da classe *Scanner* para isso. A entrada de dados deverá ser inserida no contexto de uma iteração “*while*” que ocorrerá 3 vezes, a partir de uma variável local “*contador*”.

```
entrada = new Scanner(System.in);
int contador = 1;
while(contador<=3)
{
    System.out.println("\nValor da mensalidade: ");
    mensalidade = entrada.nextDouble();
    System.out.println("Mês de referência: ");
    mes = entrada.nextInt();
    calendario = Calendar.getInstance();
    mesAtual = calendario.get(Calendar.MONTH)+1;
    contador++;
} // while
```

(8) Após a entrada de dados, implementar a codificação para receber o mês atual. Veja o código a seguir:

```
Calendar calendario = Calendar.getInstance();
mesAtual = calendario.get(Calendar.MONTH) + 1;
```

(9) após a entrada de dados, implementar a codificação necessária para verificar a nova mensalidade e imprimir os resultados. Veja código a seguir:

# Qualidade e Testes De Software

Professor Gilmar Luiz de Borba

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

(10) Implementar a codificação necessária para imprimir os resultados. Veja código a seguir:

```
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);
```

## REFATORAÇÕES

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

(2) Refatorar o código de modo a extrair uma variável local para o trecho de código:  
*mensalidade + mensalidade \* 1.10;*

na linha: *novaMensalidade = mensalidade + mensalidade \* 1.10;*

Criar a variável local: **valorCalculadoMensalidade**

(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**

(4) 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**

::continuação