

Universidade Federal de Campina Grande – UFCG

Centro de Engenharia Elétrica e Informática – CEEI

Unidade Acadêmica de Sistemas e Computação – UASC

Disciplina: Laboratório de Programação 2

Laboratório 02

O objetivo do Lab 2 é:

- 1) Usar o Eclipse.
- 2) Exercitar a sintaxe da linguagem Java por meio de implementação procedural.

Descrição da Implementação:

Será implementado o cadastro de produtos de uma loja de R\$ 1.99 chamada “Lojao P2”. Cada produto tem **um nome e uma quantidade** que deve ser comprada. O usuário define **quantos produtos diferentes serão comprados**, e então para cada um desses produtos diferentes é especificado o nome e uma quantidade que será comprada. Por exemplo:

```
Bem vindo ao Lojão LP2!
Digite a quantidade de produtos diferentes que será comprada: 4
-- Cadastro de Produtos --
Nome do Produto 1: Prato
Quantidade de Prato: 5
Nome do Produto 2: Carrinho
Quantidade de Carrinho: 9
Nome do Produto 3: Boneco de X-Men
Quantidade de Boneco de X-men: 6
Nome do Produto 4: Cabide
Quantidade de Cabide: 2
-- Cadastro Concluído --
A sua compra foi:
5 Prato
9 Carrinho
6 Boneco de X-Men
2 Cabide
Total de produtos: 22
Total gasto: R$ 43.78
-- Fim da execução --
```

Para implementar o código siga os seguintes passos:

Passo 1:

Crie a classe **CompraLojaOP2** e nela escreva um main com o código para **ler um valor inteiro** da entrada padrão (teclado). Use o **Scanner** e o **System.in**. Lembre-se de imprimir as mensagens informando ao usuário qual a informação que deve ser digitada. Use esse valor inteiro para **determinar** a quantidade de **produtos diferentes** que será comprada.

OBS: Cuidado ao usar apenas o nextInt(). Lembre-se que o usuário termina sempre com um fim de linha.

Passo 2:

Está na hora de começar a ler da **entrada os dados dos produtos**. Utilize uma estrutura de controle de **repetição** adequada para ler os "n" produtos que o usuário especificou como entrada.

Além disso, você precisará **armazenar todas** essas informações em **dois arrays**. Um contendo a informação dos **nomes** dos produtos, e o outro contendo as informações **das quantidades** do respectivo produto. Note que a **manutenção do índice é importante** pois, ele garante que você consiga resgatar a quantidade de cada produto. Considerando o nosso exemplo acima, temos:

```
nomes[0] é "Prato".  
quantidade[0] é 5.
```

Cuidado também para não sobrescrever os dados existentes ao ler informações de um produto novo.

Passo 3:

Agora que os produtos foram lidos e armazenados, está na hora de **imprimir a fatura** da sua compra, que deve incluir:

- 1) **Todos os produtos** comprados e suas respectivas quantidades.
- 2) O **total** de produtos **comprados**.
- 3) O **total gasto**, considerando que cada produto custa **R\$1.99**.

Utilize como referência o nosso exemplo apresentado no início deste documento.

```
-- Cadastro Concluído --  
A sua compra foi:  
5 Prato  
9 Carrinho  
6 Boneco de X-Men  
2 Cabide  
Total de produtos: 22  
Total gasto: R$ 43.78
```

```
-- Fim da execução --
```

Passo 4:

Chegou a hora de **expandir o Lojão P2**. Melhore a sua **impressão da fatura**. Para cada produto, indique qual a porcentagem daquele produto que seu cliente comprou. Para o nosso exemplo, temos:

```
-- Cadastro Concluído --  
A sua compra foi:  
5 Prato (23%)  
9 Carrinho (41%)  
6 Boneco de X-Men (27%)  
2 Cabide (9%)  
Total de produtos: 22  
Total gasto: R$ 43.78  
-- Fim da execução --
```

Agora **crie um outro menu** de opções com 3 escolhas: *Fazer uma compra, estabelecer um novo preço, fechar o sistema*. O seu menu deve ser da seguinte forma:

```
Bem vindo ao Lojão LP2!  
Menu Inicial: Digite uma das opções abaixo:  
  
1 - Realizar compra.  
2 - Mudar preço base de produtos.  
3 - Sair  
  
Opção:
```

Para isso, crie **uma nova classe (MenuPrincipal)** que irá conter esse seu menu principal no *main*. Note que ao escolher a opção 1, a sua classe **deve chamar** o método da sua classe **CompraLojaoP2**. Realize os *refatoramentos* adequados para suportar essas mudanças.

Passo 5:

Implemente a funcionalidade da **Opção 2**, em que o usuário (dono da loja) irá **digitar um novo preço base**. Com o aumento dos impostos e das contas de gasolina e energia elétrica, a loja deve vender seus produtos de forma flexível. Portanto, ao alterar o preço na opção 2, a classe CompraLojaoP2 deve considerar o novo preço para realizar sua funcionalidade.

A essa altura, tente **modularizar seu código** o máximo que conseguir, criando métodos para partes do código em que haja um padrão repetitivo de comportamento. Além disso, pense em partes pequenas do seu código que depois poderiam ser reutilizadas (por exemplo, para a implementação de uma lanchonete).

Para facilitar o código da Opção 2, utilize *variáveis fora de métodos*, ou seja, os **atributos das classes**. A pergunta é *quem deve possuir esse atributo? MenuPrincipal ou CompraLojaoP2? Justifique sua resposta como comentário no código*.

Passo 6:

Hora de **debugar** o seu código. Iremos aprender a usar o **Debug do Eclipse**.

O debug utiliza **breakpoints** para determinar os trechos de código que serão debugados. Ou seja, definir os trechos de código que você deseja verificar a execução passo-a-passo. Isso significa ver os estados das variáveis, sequência de chamadas, referências, etc. Quando executamos como usuário, vemos o código como uma **caixa preta**. Ao debugar, abrimos essa caixa preta para saber o que está acontecendo dentro dela.

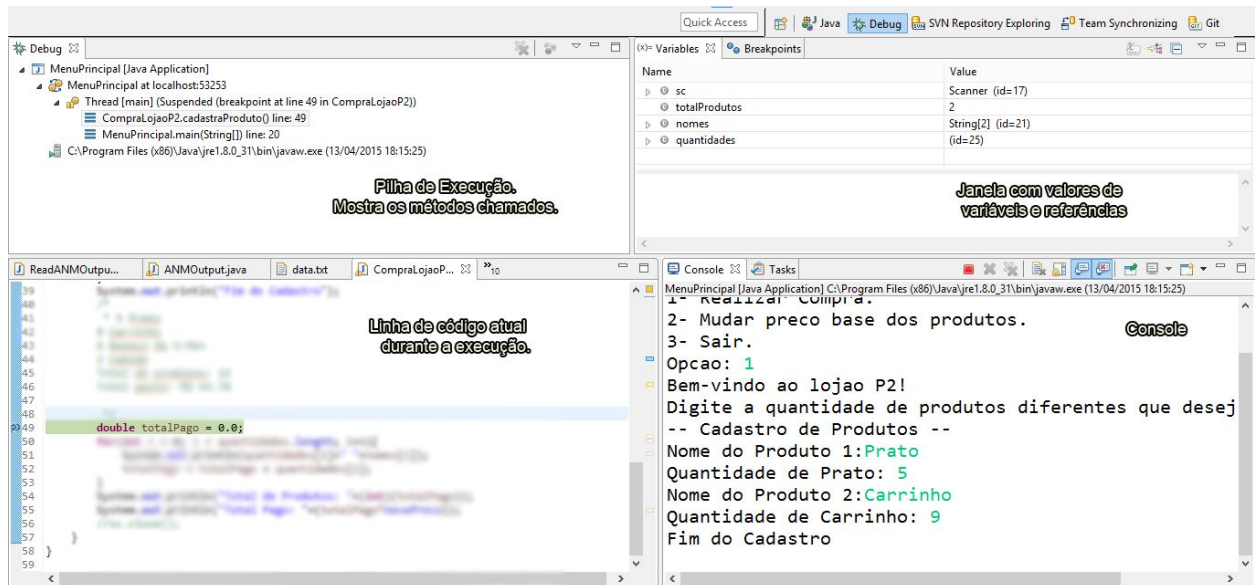
Vamos lá: Comece definindo um breakpoint no seu código de CompraLojaoP2. Digamos que você deseja debugar o seu código de **“gerar a fatura”**. Faça o seguinte:

Vá até a **linha de código** que deseja iniciar o debug. Depois vá em **Run>ToggleBreakPoint**. Você também pode usar o atalho: **Ctrl+Shift+B**. Depois disso vá em **Run>Debug As> Java Application**. Note que sua classe deve ter um main para permitir o debug.

Para nosso programa siga as seguintes instruções.

- 1) Abra a classe `CompraLojaoP2.java` e vá até a primeira linha de código da impressão de fatura.
- 2) Adicione um BreakPoint.
- 3) Volte para a classe MenuPrincipal;
- 4) Selecione **Run>Debug as>Java Application**.

Você irá executar seu programa normalmente até atingir o **breakpoint** estabelecido. Daí aparecerá um aviso solicitando que você mude para a tela de debug. Escolha “Sim”. Você verá uma tela parecida com a abaixo.



No canto superior existe um painel com as opções de **controle** do debug. São elas:

- **Resume (f8):** Interrompe o debug e retorna à execução normal.
- **Suspend:** Suspende a execução do programa para retornar ao debug.
- **Terminate:** Encerra a execução do programa ou do debug.
- **Step into (f5):** Continua a execução dentro do comando atual, caso esse seja uma chamada de método.
- **Step over (f6):** Executa o comando atual e prossegue para o próximo. Ao contrário do step into, ele não entra nos métodos da linha atual.
- **Step return (f7):** Sai do método atual (caso haja um step into) e prossegue a execução.

Tente adicionar **vários breakpoints** no seu código, e em **diferentes classes** para verificar como a **execução** é realizada e como as **variáveis** são modificadas.

Exercício Extra:

- 1) **Crie mais funcionalidades** para seu menu de CompraLojaP2. Por exemplo, crie um menu em CompraLojaProduto aonde o usuário possa escolher adicionar produtos (um por um) ou então remover produtos adicionados (use uma busca pelo nome).
- 2) **Adicione variáveis na classe MenuPrincipal** para controlar o total que foi vendido pela loja. Utilize métodos de retorno no(s) método(s) da classe CompraLojaP2.
- 3) Adicione uma **nova característica aos seus produtos**. Por exemplo, faça com que cada um tenha **seu próprio preço**, utilizando um array de double.