



MC322A - Programação Orientada a Objetos
Instituto de Computação - Unicamp
Segundo Semestre de 2016
Profa. Esther Colombini esther@ic.unicamp.br
PED: Vinicius Viali viniciusviali@gmail.com

<http://www.ic.unicamp.br/~esther/teaching/2016s2/mc322>

Laboratório 6

1 Objetivo

O objetivo desta atividade será o aprendizado do uso de Interfaces em Java.

2 Tópicos Abordados

Os tópicos a serem abordados neste laboratório incluem:

- Herança simples
- Entrada e saída
- Relacionamento de Associação 1-1
- Classe ArrayList
- Relacionamento de Associação 1-*
- Classes Abstratas
- Polimorfismo
- Interface

3 Atividade

A atividade do laboratório consiste em construir a Interface Calculavel e ajustar as classes do laboratório 5 para comportar a implementação desta Interface, conforme o diagrama apresentado na Figura 1.

A interface Calculavel contém apenas a definição do método calcularCusto(). As seguintes classes devem sofrer modificação:

- Classe Item:
 - a classe deve implementar a interface Calculavel. O método getTotal() deve ser substituído pelo método definido na interface. Ou seja, o método da classe Item que calcula o custo do Item agora deve ser o método calcularCusto(). Altere todos os locais que chamavam o método getTotal() desta classe para que invoquem o método calcularCusto().

da venda a prazo, incluindo a aplicação dos juros.

Além da interface `Calculavel`, vamos trabalhar com a interface `Comparable`. Apenas as classes `Pedido` e `Venda` devem implementar esta interface.

- Classe `Pedido`: o método `compareTo()` deve comparar o custo do pedido atual com aquele recebido como parâmetro pelo método, conforme exemplo dado em sala. Como agora a classe `Pedido` suporta o método `compareTo()`, altere o método `toString()` da classe `Venda` para que os pedidos sejam apresentados em ordem crescente de custo. Utilize o método `Collections.sort(parâmetro)` para tal.
- Classe `Venda`: o método `compareTo()` deve comparar o custo da venda atual com o daquela recebida como parâmetro pelo método.

Para testar as classes construídas, será necessário implementar uma classe contendo o método `main` (`TestaVenda.java`). Nesta classe, faça:

- Declare e Instancie um `ArrayList` para armazenar as vendas;
- Declare e Instancie ao menos 1 objeto do tipo `Fornecedor`. Os dados do fornecedor devem ser solicitados ao usuário;
- Declare e Instancie ao menos 2 objetos do tipo `Produto` ou `Produto Perecível` associando a eles o fornecedor do produto;
- Declare e Instancie ao menos 2 objetos do tipo `Item`. Associe estes objetos aos produtos criados anteriormente;
- Declare e Instancie ao menos 1 objeto do tipo `Cliente`;
- Declare e Instancie ao menos 1 objeto do tipo `Data`;
- Declare e Instancie ao menos 3 objetos do tipo `Pedido`, associando ao mesmo os objetos necessários. Insira itens no pedido.
- Solicite ao usuário se o mesmo quer (1) Criar uma venda (2) Exibir relatório de venda (3) Sair.
 - Para a opção (1), verifique se o usuário quer criar uma venda à vista (1) ou à prazo (2). Para cada caso, crie uma venda pertinente e a adicione no `ArrayList`. Não esqueça de associar os pedidos que quiser a cada venda.
 - Para a opção (2), imprima os dados de todas as vendas armazenadas no `ArrayList` até o momento;
 - Para a opção (3), encerre o programa. Antes de encerrar o programa, ordene as vendas por custo e apresente o resultado na tela.

Para realizar a entrada de dados, utilize objetos de `InputStreamReader` e `BufferedReader`, utilizados no laboratório anterior.

3.1 Submissão

A atividade da aula deve ser submetida pelo sistema Moodle (<https://www.ggte.unicamp.br/ea/>) na área correspondente à disciplina. Para isso, procure a atividade **Atividade 21/11/2016** e submeta o arquivo zipado contendo as classes Produto.java, Percivel.java, Fornecedor.java, Item.java, Pedido.java, Cliente.java, Data.java, Venda.java, VendaVista.java, VendaPrazo.java, Calculavel.java e TestaVenda.java com a sua implementação das classes. A atividade deverá ser submetida até as 23:55h do dia 25/11/2016. Apenas alunos presentes no laboratório podem submeter.