

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 4

1 Objetivo

O objetivo desta atividade será o aprendizado de relacionamentos de associação 1-.*.

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-*

3 Atividade

A atividade do laboratório consiste em implementar 3 classes, Cliente, Data e Pedido, conforme descrito na Figura 1.

- Classe Cliente, contendo:
 - as propriedades nome, cpf e endereço;
 - os métodos de acesso, um construtor que recebe a nome, o cpf e o endereço e um método toString() que retorna os dados do cliente.
- Classe Data, contendo:
 - as propriedades dia, mês e ano;
 - os métodos de acesso, um construtor que recebe o dia, mês e ano e um método toString() que retorna a data no forma dd/mm/aa.

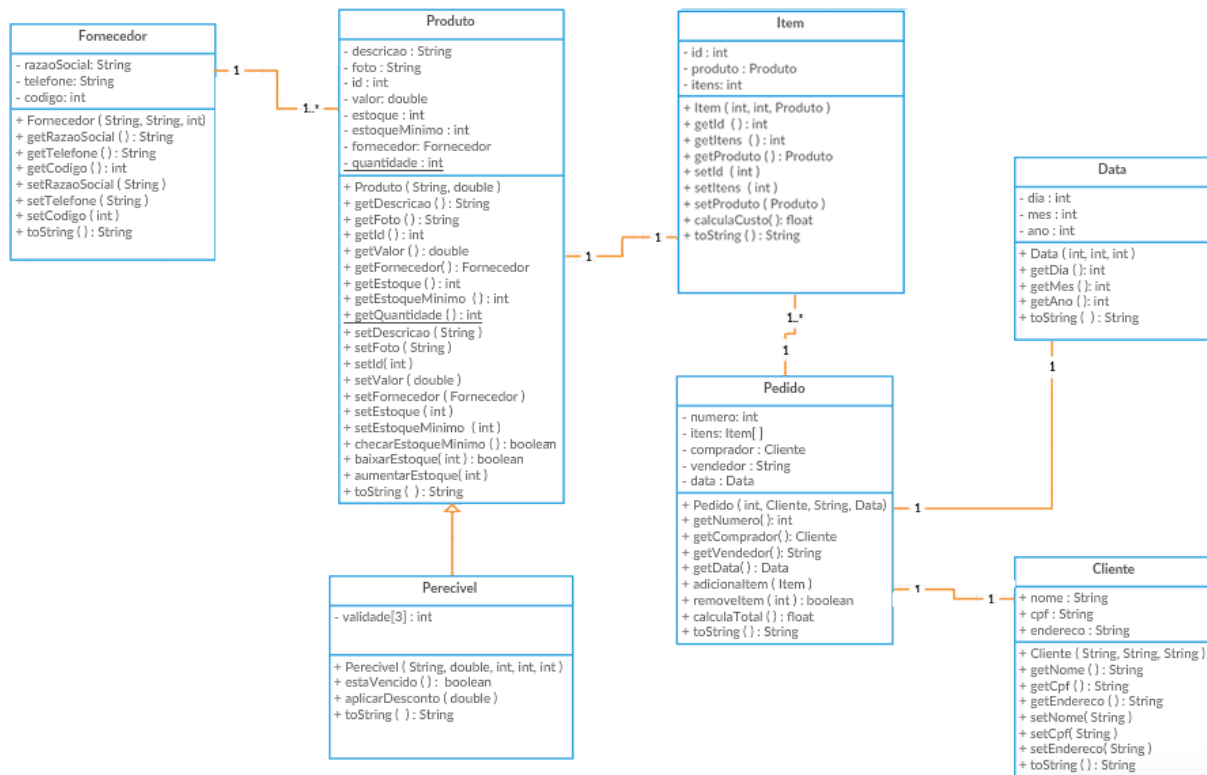


Figura 1: Diagrama de classes a ser implementado.

- Classe Pedido, contendo:

- as propriedades número (código do pedido), comprador, vendedor e data, além de um ArrayList de Item para armazenar os itens do pedido;
- o método construtor que recebe o código do pedido, o cliente que está comprando, o vendedor e a data do pedido. Além disso, a classe deve conter os métodos de acesso e o método toString(), que retorna uma String com as informações do pedido e de todos os elementos a ele associados;
- o método adicionaItem que recebe um item e o adiciona no pedido. Este método deve, além de adicionar o elemento no carrinho, realizar a baixa no estoque do produto associado ao item. Ou seja, se estamos adicionando um item que tem 2 unidades de um determinado produto, o estoque referente aquele produto deve ser diminuído em 2. Caso não haja estoque suficiente, o produto não pode ser adicionado;
- o método removeItem que recebe a posição do item no ArrayList e o remove do pedido. Este método deve, além de remover o elemento no carrinho, adicionar ao estoque do produto associado ao item o número de produtos removido. Ou seja, se estamos removendo um item que tem 2 unidades de um determinado produto, o estoque referente aquele produto deve ser aumentado em 2;
- um método calculaTotal que retorna o custo total do pedido.

Além destas duas classes, as seguintes modificações na classe Produto, implementada na atividade referente ao Laboratório 3, devem ser realizadas:

- adicionar o método `baixarEstoque(int)` que recebe um valor que deve ser diminuído do estoque. Este método retorna `true` se a quantidade a ser baixada for menor ou igual ao parâmetro recebido e `false` caso contrário.
- adicionar o método `aumentarEstoque(int)` que recebe um valor que deve ser acrescentado ao estoque.

Também será necessário adicionar na classe `Item`, implementada na atividade referente ao Laboratório 3, o método `calculaCusto()` que retorna o custo do item (`itens * preço do produto`).

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

- 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;
- Verifique se a quantidade mínima de estoque dos produtos foi ou não atingida. Imprima esta condição na tela;
- 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 1 objeto do tipo `Pedido`, associando ao mesmo os objetos necessários. Insira itens no pedido.
- Imprima os dados do pedido da tela.
- Remova um item do pedido.
- Imprima os dados do pedido da 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 03/10/2016** e submeta o arquivo zipado contendo as classes `Produto.java`, `Perecivel.java`, `Fornecedor.java`, `Item.java`, `Pedido.java`, `Cliente.java`, `Data.java` e `TestaPedido.java` com a sua implementação das classes. A atividade deverá ser submetida até as 23:55h do dia 07/10/2016. Apenas alunos presentes no laboratório podem submeter.