



MC302 - Programação Orientada a Objetos
Instituto de Computação - Unicamp
Primeiro Semestre de 2017
Profa. Esther Colombini esther@ic.unicamp.br
PEDs: Elisangela Santos (ra149781@students.ic.unicamp.br)
Lucas Faloni (lucasfaloni@gmail.com)
Lucas David (lucasolivdavid@gmail.com)
Wellington Moura (wellington.tylon@hotmail.com)
PAD: Igor Torrente (igortorrente@hotmail.com)
<http://www.ic.unicamp.br/~esther/teaching/2017s1/mc302>

Atividade de Avaliação Continuada

1 Objetivo

O objetivo desta atividade é avaliar o nível de absorção da turma do conteúdo apresentado no curso até o momento.

2 Tópicos Abordados

Os tópicos a serem avaliados incluem:

- Princípios de orientação a objetos: encapsulamento e modularidade
- Classes e objetos
- Variáveis e métodos (instância e atributos)
- Sobrecarga de métodos
- Entrada e saída
- Arrays

3 Atividade

Com o intuito de avaliar o nível de absorção do conhecimento do curso até o momento, a atividade deverá ser realizada em dupla, em papel e sem o auxílio do computador. Não serão atribuídas notas para esta atividade, mas os alunos receberão um retorno sobre os principais problemas encontrados na solução proposta.

A atividade proposta consiste em implementar 2 classes, Produto e Venda, conforme descrição a seguir. Para a classe Produto indique como seria, além do código da classe, a representação da mesma em UML.

Desejamos construir um sistema capaz de realizar a venda de produtos. Para tal, devemos construir uma classe capaz de armazenar as seguintes informações referentes ao estoque de uma loja, a classe Produto, que contenha as seguintes informações:

Classe Produto:

1. Atributos:

- os atributos de instância foto, id, valor, nome, quantidadeEmEstoque, estoqueMinimo e fornecedor
 - foto: local de armazenamento da foto do produto em disco;
 - id: deverá ser um valor único definido automaticamente na criação do objeto, ou seja, cada novo objeto deve ter o seu id automaticamente definido pelo sistema;
 - valor: valor unitário do produto;
 - nome: nome do produto;
 - quantidadeEmEstoque: quantidade de produtos em estoque atualmente
 - estoqueMinimo: quantidade que, quando atingida, deve disparar uma mensagem para que seja realizada a compra de mais produtos;
 - fornecedor: nome do fornecedor.
- o atributo de classe quantidadeProdutos que armazena o número de produtos criados até o momento.

2. Métodos:

- os métodos de acesso para todas as variáveis privadas;
- aos menos dois construtores sobrecarregados;
- um método booleano baixarEstoque(int) que recebe como parâmetro um valor que indica a quantidade a ser diminuída do estoque atual. Este método retorna true se a quantidade a ser baixada for menor ou igual ao parâmetro recebido e false caso contrário;
- um método sobrecarregado booleano baixarEstoque() que retira uma unidade do produto do estoque atual. Este método retorna true se a quantidade a ser baixada for menor ou igual ao parâmetro recebido e false caso contrário;
- um método aumentarEstoque(int) que recebe um valor que indica a quantidade a ser acrescentada ao estoque;
- um método checarEstoqueMínimo() que retorna true se a quantidade em estoque for menor ou igual ao estoque mínimo e falso caso contrário;
- um método toString() para retornar uma String contendo as propriedades da classe;
- um método de classe que retorna o número de objetos da classe instanciados até o momento.

Classe Venda:

A classe Venda será aquela que conterà o método main, sendo responsável pela instanciação dos objetos e interação com o usuário. Você pode dividir as atividades da classe Venda em métodos, caso entenda apropriado. Para testar o programa, deverão ser instanciados 5 objetos do tipo Produto. Os mesmos deverão ser armazenados em um Array. Os valores das propriedade de cada produto podem ser definidas pelo programador.

A seguir, a seguinte sequência deve ser seguida:

- Apresente um menu para o usuário com as opções:
 1. **Listar produtos:** caso o usuário selecione esta opção, devem ser apresentadas na tela as informações sobre todos os produtos disponíveis;

2. **Comprar produto:** caso o usuário selecione esta opção, deve-se solicitar o código e a quantidade do produto a ser comprada. O sistema deverá: 1) baixar automaticamente o estoque do produto caso exista quantidade suficiente ou avisar que a compra não pode ser realizada porque a quantidade em estoque é inferior a que se deseja comprar; 2) mostrar o valor gasto nesta compra (valor unitário * quantidade) e 3) apresentar uma mensagem de estoque baixo caso o estoque mínimo tenha sido atingido;
3. **Devolver produto:** caso o usuário selecione esta opção, deve-se solicitar o código e a quantidade do produto a ser devolvido. O sistema deverá: 1) aumentar automaticamente o estoque do produto; 2) indicar o valor a ser estornado ao comprador;
4. **Finalizar comprar:** caso o usuário selecione esta opção, o programa deve ser encerrado apresentando na tela o estado atual de todos os produtos disponíveis.

3.1 Entrega

A atividade deverá ser entregue ao final da aula e deverá conter a identificação dos integrantes da dupla.