Algoritmos e Programação em C/C++

Ano/Semestre: 2018/1 Horários: 23 e 43 (Segundas e quartas à noite)

Prof. Dr. Daniel Stefani Marcon - danielstefani@unisinos.br
Prof. Ms. Márcio Miguel Gomes - marciomg@unisinos.br

Trabalho do Grau B

Sistema para o Gerenciamento de uma Confeitaria

Apresentação: Os trabalhos serão apresentados pelos grupos diretamente ao professor na aula do dia **25/06/2018**. Não haverá prorrogação de data. O tempo máximo para apresentação por grupo será de 20 minutos. A ordem de apresentação será definida no dia da apresentação.

Instruções para envio do trabalho: Enviar somente os arquivos-fonte do projeto para a atividade aberta no **Moodle até às 19h30min do dia 25/06/2018**. Envios após esse horário terão desconto de 1 ponto. Apenas um integrante do grupo deve enviar os arquivos informando o nome de todos os integrantes do grupo.

Grupos: no máximo 3 integrantes.

Objetivo: Desenvolver um programa para gerenciamento básico de uma confeitaria.

Escopo do projeto

Uma confeitaria está precisando implementar um sistema de controle de estoque dos seus produtos e insumos. Cada insumo catalogado deve conter um número identificador, nome, valor, unidade de medida, quantidade atual em estoque e quantidade mínima em estoque. Os alimentos produzidos por essa confeitaria também devem ser registrados no sistema, contendo um número de identificação, nome do produto, valor de venda, quantidade atual em estoque e quantidade mínima em estoque.

A cada turno, o administrador da confeitaria gera um relatório dos produtos cuja quantidade em estoque seja inferior à quantidade mínima definida, e passa a quantidade faltante de cada produto para o confeiteiro produzi-los. Ao produzir os alimentos, automaticamente é registrada a baixa nos respectivos insumos e atualizada a quantidade em estoque dos produtos. O administrador também gera um relatório listando todos os insumos cuja quantidade em estoque seja menor que a quantidade mínima definida, e realiza a compra das quantidades faltantes. Após a realização destes procedimentos, nenhum produto ou insumo deve possuir quantidade inferior à quantidade mínima definida.

No caixa, o atendente registra as vendas dos produtos realizando a baixa em estoque de acordo com as quantidades vendidas de cada produto. Caso não haja produto suficiente em estoque, a venda não deve ser realizada.

Persistência de dados:

Ao abrir o sistema, automaticamente a lista de insumos e produtos deve ser carregada para a memória a partir de arquivos-texto tabulados. A qualquer momento, deve ser possível salvar nos mesmos arquivos os dados que estão em memória. A estrutura dos arquivos representa os atributos das classes Produto e Insumo. Arquivos de produtos e insumos para serem utilizados como base estão disponíveis no Moodle.

Algoritmos e Programação em C/C++

Ano/Semestre: 2018/1 Horários: 23 e 43 (Segundas e quartas à noite)

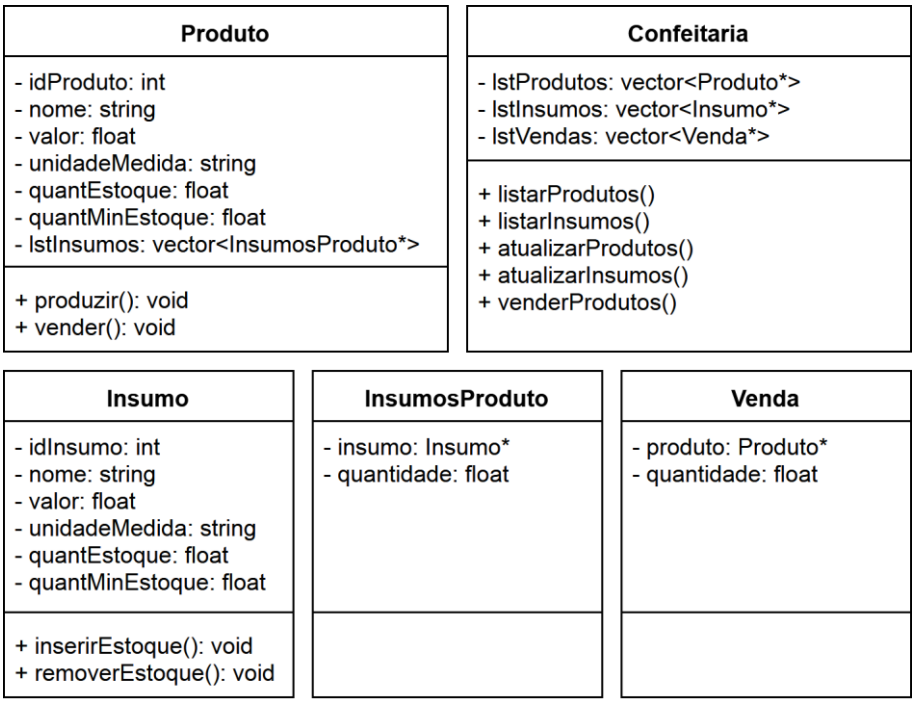
Prof. Dr. Daniel Stefani Marcon - danielstefani@unisinos.br
Prof. Ms. Márcio Miguel Gomes - marciomg@unisinos.br

Menu:

Código	Opção	Descrição
1	Realizar venda	Dentro de uma estrutura de repetição, o usuário informa a quantidade desejada e o nome do produto. Caso a quantidade seja menor ou igual a zero, encerra o laço. Do contrário, registra a venda caso o produto exista e tenha quantidade suficiente em estoque.
2	Consultar um produto	O usuário informa o nome ou código identificador de um produto e o programa lista todos os seus atributos, juntamente com uma relação dos insumos necessários para produzi-lo.
3	Consultar um insumo	O usuário informa o nome ou código identificador de um insumo e o programa lista todos os seus atributos.
4	Listar produtos a produzir	Listagem de todos os atributos dos produtos com quantidade em estoque menor que a quantidade mínima definida. Caso a listagem não seja vazia, perguntar ao usuário se ele quer produzir as quantidades faltantes. Caso afirmativo, produzir as quantidades, registrar no sistema e dar baixa nos insumos consumidos.
5	Listar insumos a comprar	Listagem de todos os atributos dos insumos com quantidade em estoque menor que a quantidade mínima definida. Caso a listagem não seja vazia, perguntar ao usuário se ele quer comprar as quantidades faltantes. Caso afirmativo, comprar as quantidades e registrar no sistema.
6	Listar todos os produtos	Listagem de todos os atributos de todos os produtos.
7	Listar todos os insumos	Listagem de todos os atributos de todos os insumos.
8	Listar vendas	Listar os detalhes de todas as vendas realizadas juntamente com o saldo.
9	Salvar	Salva os dados em arquivos texto tabulados.
0	Sair	Encerra o sistema. Caso algum dado não tenha sido salvo em arquivo, salvar automaticamente.

Estrutura de Classes:

A estrutura abaixo é uma proposta que serve como ponto de partida para o desenvolvimento do sistema. Adicione novos métodos e atributos caso seja necessário.



Algoritmos e Programação em C/C++

Ano/Semestre: 2018/1 Horários: 23 e 43 (Segundas e quartas à noite)

Prof. Dr. Daniel Stefani Marcon - danielstefani@unisinos.br
Prof. Ms. Márcio Miguel Gomes - marciomg@unisinos.br

* **Observação 1:** Todas as opções do menu devem ser implementadas. A não-implementação de alguma opção acarretará em um desconto na nota final do grupo.
* **Observação 2:** Os nomes de classes, atributos e métodos especificados acima na estrutura das classes devem ser mantidos na implementação do código (ou seja, não renomeie-os). Novos métodos e atributos devem ser nomeados de acordo com a sua respectiva função.
* **Observação 3**: Implemente métodos **getters** e **setters** para os atributos das classes. Métodos setters devem verificar o valor sendo setado ao atributo quando necessário (por exemplo, a quantidade mínima de um produto não deve ser negativa). Métodos para alterar o valor de um atributo podem ser úteis (por exemplo, o método addQuantEstoque(float quant) adiciona o valor de "quant" ao valor do atributo "quantEstoque" de um objeto Insumo).

Critérios de avaliação:

- O código-fonte entregue deve ser **compilável** e **executável**;
- O programa deve ser todo **orientado a objetos**;
- O código-fonte deve estar **corretamente indentado** e **comentado**;
- Os **nomes** dos atributos, métodos, classes, parâmetros e variáveis utilizados devem ser **autoexplicativos**, utilizando a notação "**camel case**";
- Devem ser utilizados todos os comandos e conceitos especificados na definição desse trabalho (incluindo orientação a objetos, vectors, iteradores, manipulação de arquivos, strings, comandos de seleção e comandos de repetição);
- Todas as funcionalidades do programa contidas nesta definição devem ser implementadas;
- Todos os componentes do grupo devem explicar uma parte do trabalho durante a apresentação;
- Qualidade das respostas às perguntas do professor durante a apresentação.