

1 Especificação do Trabalho

1. **(peso 2)** Elabore um Plano de Teste para o programa especificado na Seção 3. O plano deve considerar a aplicação do critério de teste Teste Funcional Sistemático.
2. **(peso 2)** Defina os requisitos de teste de acordo com o Plano de Teste elaborado. Para isso, considere as restrições definidas para os dados de entrada. Além disso, siga a sua intuição para definir outras condições de teste relevantes.
3. **(peso 1)** Elabore um programa em Java que atenda a especificação apresentada na Seção 3.
4. **(peso 1)** Implemente os casos de teste em JUnit para cobrir os requisitos de teste definidos no item 2.
5. **(peso 4)** Escreva um relatório de teste, após a execução dos testes JUnit, o qual deve incluir:
 - (25%) Um sumário dos resultados obtidos
 - (25%) A lista de falhas observadas durante a execução dos testes.
 - (50%) Informações sobre a cobertura estrutural do código, de acordo com os critérios Todos-Nós e Todas-Arestas

2 Critérios de avaliação

- Ver pesos associados a cada item solicitado.
- Avaliações adicionais
 - **(peso 10)**: Programa passar nos casos de teste definidos pelo professor. Para isso, é mandatório seguir as especificações das classes definidas em sala de aula.
 - Interface criada: bônus de até 10% na nota final do trabalho, considerando a interação clara e correta entre o usuário e o sistema.

Instruções Gerais

- O trabalho deverá ser realizado por grupos de 5 alunos. Exceções devem ser previamente autorizadas pelo professor.

- O trabalho deve ser entregue em arquivo único no Moodle. O arquivo deverá conter todos os artefatos criados pelo grupo. O projeto Eclipse que contém o programa fonte (Java) e os casos de teste (JUnit) devem estar contidos no arquivo entregue. O mesmo programa enviado ao Moodle será utilizado para a apresentação ao professor.
- A data de entrega é 17/06/2015 (horário limite: 13h55).

3 Especificação do Programa

O programa deve gerenciar a venda de produtos a clientes, oferecendo as seguintes funcionalidades:

- Gerenciamento de produtos: inclusão e alteração. Para cada produto, os seguintes dados são armazenados: código (único), descrição, preço de compra, preço de venda, unidade, quantidade em estoque e status (ativo ou inativo).

Restrições:

– Entrada:

- * Código: número inteiro positivo (o programa não deve permitir códigos repetidos para produtos).
- * Descrição: alfanumérico, tamanho mínimo 3 caracteres, tamanho máximo 64 caracteres
- * Preço de compra: número real positivo
- * Preço de venda: número real positivo, maior ou igual ao preço de compra
- * Unidade: alfabético, tamanho 2 caracteres
- * Estoque: número real positivo (incluindo zero)
- * Status: campo numérico, sendo 1 = ativo e 0 = inativo
- * Um produto não pode ser desativado se tiver itens em estoque.

– Saída:

- * RuntimeException caso a inclusão (ou alteração) não ocorra com sucesso.

- Gerenciamento de clientes: inclusão e alteração. Para cada cliente, os seguintes dados são armazenados: nome, CPF (único), endereço, telefone, e-mail e status (ativo ou inativo).

Restrições:

– Entrada:

- * Nome: alfabético, com tamanho máximo 128 caracteres, composto por no mínimo duas palavras com tamanho mínimo de 2 caracteres cada
- * CPF: ver regras para validação de CPF em http://www.geradorcpf.com/algoritmo_do_cpf.htm. (o programa não deve permitir CPFs repetidos para clientes).
- * Endereço: alfanumérico, tamanho máximo 256 caracteres
- * Telefone: deve conter apenas números e os símbolos “(” e “)”, seguindo o template “(XX)XXXXXXXXXX”, no qual os dois dígitos entre parênteses representam o DDD.
- * E-mail¹: deve seguir o template “USUARIO@DOMINIO.COMPLEMENTO”, no qual USUARIO e DOMINIO são sequências de caracteres iniciadas por uma letra e seguidas por letras e números, e COMPLEMENTO é uma das seguintes strings: “com” ou “com.br”
- * Status: campo numérico, sendo 1 = ativo e 0 = inativo

– Saída:

- * RuntimeException caso a inclusão (ou alteração) não ocorra com sucesso.
- Alteração da quantidade em estoque dos produtos, de acordo com o código do produto informado.

Restrições:

– Entrada:

- * Código: número inteiro positivo.
- * Estoque: número real positivo (incluindo zero)

– Saída:

- * RuntimeException caso a alteração não ocorra com sucesso.

- Consulta de produtos por descrição.

Restrições:

– Entrada:

¹Regras simplificadas

- * Descrição: alfanumérico, tamanho mínimo 3 caracteres, tamanho máximo 64 caracteres
- Saída:
 - * *array* de números inteiros positivos, possivelmente vazio, contendo todos os códigos dos produtos cujas descrições se iniciam com o valor fornecido como entrada.
 - * *array* cujo primeiro elemento é o valor -1, caso o valor de entrada não respeite a respectiva restrição
- Registro de vendas de produtos para clientes, com os seguintes dados: número do registro de venda, CPF do cliente, data da venda, lista de produtos adquiridos e a quantidade vendida de cada item. Uma venda pode incluir mais de um produto, sendo que cada item vendido pode ter uma quantidade diferente. Somente produtos cadastrados podem compor os registros de vendas.

Restrições:

- Entrada:
 - * Número do registro de venda: número inteiro positivo (o programa não deve permitir números repetidos para registros de vendas).
 - * Data da venda: no formato DD/MM/AAAA, respeitando as regras para datas válidas do calendário
 - * Quantidade de itens vendidos: número inteiro positivo
 - * Quantidades vendidas de cada item: *array* de números reais positivos, cujo tamanho deve corresponder à quantidade de itens vendidos.
 - * Demais entradas: observar regras definidas para produtos e clientes.
 - * A entrada do item deve ser uma string única, contendo todos os campos separados por ponto e vírgula. A ordem dos campos na string deve ser a seguinte: número do registro de venda, CPF do cliente, data da venda, código do produto 1, quantidade do produto 1, código do produto 2, quantidade do produto 2, ..., código do produto n, quantidade do produto n.
 - * Importante: a quantidade de cada produto a ser vendido deve estar disponível no estoque; caso contrário, a venda não deve ser efetuada. Se a venda for efetuada, os estoques dos produtos devem ser atualizados. A baixa do estoque de produtos deve ser efetuada somente após o encerramento do registro de vendas.

– Saída:

* RuntimeException caso a venda não ocorra com sucesso.

- Cálculo do faturamento em um determinado mês. Para isso, o usuário deve informar o mês desejado.

Restrições:

– Entrada:

* Mês: número inteiro positivo, representando um mês válido do calendário

– Saída:

* Faturamento calculado, tipo real, caso o mês seja válido; valor -1.0, caso o mês seja inválido.

Programa Principal e Definição das Classes

- A implementação das classes deverá seguir as especificações definidas em sala de aula. Ver diagrama de classes no Moodle.
- A classe central do sistema (Mercado) deve ***obrigatoriamente*** conter as operações `calculaFaturamento` e `consultaProdutos`. Outras operações necessárias para gerenciar as demais funcionalidades do sistema devem ser definidas a critério do grupo de alunos.
- Crie métodos acessores (*getters*) para facilitar a criação de casos de teste e as respectivas assertivas.
- (item não obrigatório) A interface (programa principal) deve possibilitar a escolha da funcionalidade desejada pelo usuário. Deve permitir, *por exemplo*, a inclusão de um novo produto, a entrada de registros de vendas e o cálculo do faturamento mensal.