

Introdução à Programação Orientada por Objetos

Mini-Projeto 3



Máquina de venda automática

Ano Letivo: 2015/2016

Época Normal e de Recurso

Índice

Índice	2
Introdução.....	3
Descrição Geral do Problema	3
Descrição adicional de características e funcionalidades pretendidas.....	3
Regras de codificação.....	4
Regras de Desenvolvimento e Entrega do Projeto	5
Data de entrega	5
Implementação e codificação	5
Constituição de grupos.....	5
Entrega do projeto.....	5
Regras e Critérios de Avaliação do Projeto	6
Regras de Avaliação	6
Critérios de Avaliação	6

Introdução

O objetivo deste mini-projeto é desenvolver, utilizando a linguagem Java e a Programação Orientada por Objetos (POO), uma aplicação em consola, capaz de simular uma máquina de venda automática de produtos alimentares, no que diz respeito às operações de venda e às operações de manutenção.

Descrição Geral do Problema

Pretende-se efetuar a simulação duma máquina de venda automática de produtos alimentares. A máquina de venda automática a simular possui vários compartimentos onde estão armazenados um conjunto de produtos do mesmo tipo.

Para adquirir um produto o utilizador deve selecionar a localização do compartimento em que se encontra o produto pretendido e a seguir fazer o pagamento com um cartão porta-moedas eletrónico. O valor do produto é debitado automaticamente no cartão.

A máquina possui um teclado para a introdução das localizações e um pequeno ecrã onde são mostradas informações tal como o produto selecionado, o preço do produto e mensagens adicionais de funcionamento.

A manutenção da máquina é feita por um gestor que se encarrega de adicionar, retirar ou substituir os produtos alimentares.

Descrição adicional de características e funcionalidades pretendidas

As máquinas de venda automática desta aplicação possuem um código de identificação composto por 6 dígitos. Os compartimentos destas máquinas estão distribuídos por 6 prateleiras, onde as primeiras 3 têm 5 compartimentos grandes com capacidade para 8 produtos cada e as outras 3 têm 10 compartimentos com capacidade para 12 produtos cada. A localização dos produtos é representada por uma letra e um algarismo. A letra entre 'A' e 'F' indica a prateleira e o algarismo entre 0 e 9 indica a posição dentro da prateleira.

A interação com o utilizador é efetuada por um pequeno teclado embutido na máquina. Este teclado possui 6 teclas com as letras correspondentes às prateleiras, teclas com os algarismos entre 0 e 9, uma tecla de 'ok' e uma tecla para cancelar as operações. O ecrã da máquina fornece as informações de operação. Inicialmente mostra uma mensagem para que se escolha o produto, o utilizador fornece de seguida a localização do produto, depois é mostrada a informação da localização do produto pretendido e do preço do mesmo. Se a localização não existir ou o produto estiver esgotado deve ser mostrada essa informação. O pagamento é feito com a introdução dum cartão porta-moedas pré carregado com a quantia necessária à aquisição. Se não existir esse valor no cartão é mostrada uma mensagem de quantia insuficiente, caso contrário o valor do produto é deduzido ao cartão e uma unidade do produto escolhido é retirada da localização fornecida. A tecla de 'ok' deve ser usada após se ter fornecido uma localização e a tecla de cancelar reinicia em qualquer altura o processo de aquisição, voltando-se a exibir a mensagem inicial.

O cartão porta-moedas possui um número inteiro de identificação único (sequencial e gerado automaticamente) e o valor em cêntimos. Este cartão pode ser carregado com uma determinada quantia sendo que o limite máximo são 10000 cêntimos.

Os produtos em venda possuem uma descrição e um preço.

A manutenção da máquina é feita por um gestor que pode adicionar, retirar ou substituir produtos. Estas operações devem ser simuladas utilizando um método que recebe um código “secreto”, o tipo de operação a efetuar (adicionar, retirar ou substituir produtos) e a informação de produtos e localizações necessária à operação em causa. Neste caso o(s) parâmetro(s) de produtos e/ou de localizações pode(m) ir vazio(s). O valor de retorno indica se a operação foi bem-sucedida ou não. O código secreto é um número inteiro que é criado aleatoriamente pela máquina de venda automática e que é conhecido apenas pelo gestor.

Além das operações anteriores o gestor deve ainda poder:

1. Reiniciar a máquina eliminando toda a informação de produtos existentes e de créditos acumulados. Deve ser igualmente produzido um novo código secreto de acesso.
2. Saber qual o crédito acumulado da venda de produtos.
3. Retirar o crédito acumulado total ou parcialmente.
4. Listar todos os produtos existentes numa localização ou em todas as localizações.

As operações descritas anteriormente devem igualmente ser implementadas por métodos em que todos eles recebem como primeiro parâmetro o “código secreto” do gestor. A operação 1. deve retornar o novo código secreto, a operação 2. retorna o valor do crédito acumulado, a operação 3. retorna o valor do crédito retirado e finalmente a operação 4. retorna as listagens pedidas em forma de texto.

Regras de codificação

A descrição do problema não fornece a informação das classes e métodos a criar, essa é uma decisão que deverá ser tomada na implementação da aplicação. As classes e métodos criados devem ter em conta os princípios de coesão, acoplamento e desenho orientado por responsabilidades explicados nas aulas.

Deve igualmente pensar antecipadamente nas alterações ou adaptações mais óbvias que o código poderia vir a ter criando a solução de acordo com essa visão.

Não é necessário criar qualquer funcionalidade extra que não esteja descrita no enunciado.

Todas as classes devem ser testadas para que se garanta o seu funcionamento correto.

Regras de Desenvolvimento e Entrega do Projeto

Data de entrega

1 de fevereiro de 2016.

Implementação e codificação

O programa deve ser desenvolvido utilizando a linguagem Java, colocando em prática os conceitos fundamentais do paradigma de Programação Orientada por Objetos.

Em relação às regras de codificação, siga as convenções adoptadas normalmente para a linguagem Java:

- A notação *camelCase* para os nomes de métodos e dos seus parâmetros, de variáveis locais e de atributos;
- A notação *PascalCase* para os nomes das classes;
- Não utilize o símbolo '_' nos identificadores, nem abreviaturas.

É necessário que o projeto cumpra o que é pedido no seu enunciado, sendo deixado ao critério do programador qualquer pormenor de implementação que não seja referido, o qual deverá ser devidamente documentado.

Constituição de grupos

Cada projeto deverá ser elaborado em grupos de dois alunos, podendo eventualmente ser elaborado individualmente. Não serão permitidos, em nenhum caso, grupos com mais do que dois alunos.

Os grupos dos alunos já se encontram determinados através da metodologia de *pair programming* que está a ser utilizada nos laboratórios. Caso existam alunos que não têm o grupo escolhido, deverão contactar o respectivo docente de laboratório para regularizar a situação.

Entrega do projeto

- O projeto deverá ser entregue até à data limite especificada por **via exclusivamente eletrónica utilizando a área dos trabalhos no Moodle 2.0**. Todos os ficheiros que compõem o projeto deverão estar guardados num único ficheiro compactado em **formato ZIP**. Em caso de dificuldades no acesso à plataforma Moodle, o envio dos ficheiros poderá ser feito por correio eletrónico para o respetivo docente de laboratório, dentro do prazo acima indicado.
- **Não serão aceites quaisquer projetos entregues fora do prazo!**
- Todos os materiais do projeto devem ser devidamente identificados com nome, número e endereço de correio eletrónico dos alunos.

Os materiais do projeto deverão incluir:

- A documentação das interfaces da classe gerado pelo JavaDoc.
- A documentação do programa que explique de uma forma simples as classes criadas, juntamente com os seus atributos e métodos e qualquer detalhe de implementação que necessite de explicações adicionais.
- O código fonte do programa na forma de projeto em *NetBeans*, com todas as funcionalidades implementadas.
- Todos os ficheiros que compõem o projeto deverão estar guardados num único ficheiro compactado em formato ZIP cujo nome deverá ter a seguinte nomenclatura:
<curso>_<numAluno1>_<numAluno2>.zip.

Regras e Critérios de Avaliação do Projeto

Regras de Avaliação

A avaliação do projeto está sujeita às seguintes regras:

- **Não serão aceites quaisquer projetos entregues fora do prazo!**
- A classificação do programa terá em conta a qualidade da programação e a estrutura do código criado segundo os princípios da Programação Orientada por Objetos.
- Serão premiadas a imaginação e a criatividade.
- O projeto terá uma componente de avaliação oral obrigatória com classificação individual dos elementos do grupo.
- Os alunos que não comparecerem à discussão serão classificados com zero na fase respectiva. Nesta discussão será apurada a capacidade do aluno de produzir o código apresentado. Nos casos em que essa capacidade não for demonstrada, a nota atribuída será zero.
- A avaliação oral é realizada pelo respectivo professor de laboratório e irá ser feita uma marcação prévia para cada grupo de trabalho.
- Todos os projetos serão submetidos a um sistema automático de deteção de cópias. Os projetos que forem identificados como possíveis cópias, e verificando-se serem realmente cópias, serão anulados.

Critérios de Avaliação

O mini-projeto 3 será avaliado segundo os seguintes critérios:

Entidades	50%
Funcionalidades Requeridas	20%
Coesão de classes e métodos	10%
Acoplamento	10%
Desenho orientado por responsabilidades	10%

Implementação	20%
Conhecimento e boa utilização da linguagem	10%
Bom estilo (nomes, comentários, indentação, elementos JavaDoc)	10%

Testes	10%
Testes utilizando a framework JUnit	10%

Documentação	10%
---------------------	------------

Avaliação qualitativa	10%
------------------------------	------------