



Introdução à Programação Orientada por Objetos

Projeto – 1ª Fase

Sistema de Distribuição de Mercadorias

Ano Letivo: 2016/2017

Época Normal e de Recurso

Índice

1	Introdução	3
2	Detalhes de Implementação	3
2.1	Funcionalidades	3
2.1.1	Packs de Bens.....	3
2.1.2	Camiões	3
2.1.3	Contentores.....	4
2.1.4	Estações Abastecedoras ou de Distribuição	4
2.1.5	Lojas.....	4
2.1.6	Garagens	5
3	Regras de Desenvolvimento e Entrega do Projeto	6
3.1	Entrega.....	6
3.2	Implementação e codificação.....	6
3.3	Constituição de grupos.....	6
3.4	Entrega do projeto.....	7
4	Regras e Critérios de Avaliação do Projeto	7
4.1	Regras de Avaliação.....	7
4.2	Critérios de Avaliação	8

1 Introdução

O objetivo deste projeto é desenvolver uma aplicação, utilizando a linguagem Java e a Programação Orientada por Objetos (POO), com o intuito de gerir e simular a atividade de distribuição de mercadorias entre estações de distribuição e lojas.

No problema a modelar, destacam-se as seguintes entidades:

- Estações de distribuição que armazenam os produtos para consumo e disponibilizam contentores para o seu transporte.
- Lojas onde os produtos são vendidos aos consumidores.
- Camiões com contentores que armazenam e transportam os bens entre estações de distribuição e lojas.
- Garagens onde os camiões estão estacionados e que efetuam operações de inspeção e manutenção desses camiões.

Na secção seguinte são apresentados os detalhes a considerar na implementação do projeto.

2 Detalhes de Implementação

2.1 Funcionalidades

2.1.1 Packs de Bens

A aplicação irá gerir o transporte de bens entre diversas entidades. Os bens existentes devem ser representados em conjuntos (packs) onde se definem o código do produto (um número inteiro), o nome, a quantidade, o peso unitário (em Kg) e o volume unitário (em cm³).

2.1.2 Camiões

O transporte de mercadorias é efetuado por camiões que são caracterizados pela seguinte informação:

- Uma designação textual e um número para identificar o camião.
- Um contentor onde as mercadorias são armazenadas, o qual será descrito no ponto seguinte.
- Uma posição, representada através de coordenadas geográficas. Assuma o valor da latitude no intervalo de $[-90.0, 90.0]$ graus (>0 é norte, <0 é sul) e o da longitude de $[-180.0, 180.0]$ graus (>0 é este, <0 é oeste).
- Número total de quilómetros percorridos.
- Número de quilómetros efetuados desde a última inspeção realizada ao camião.
- Número de inspeções efetuadas.

As principais funcionalidades a implementar nesta entidade são as seguintes:

- Mover o camião para uma outra posição. Tenha em atenção que é necessário atualizar o número total de quilómetros percorridos. Utilize, por simplicidade, distância euclidiana para o cálculo da distância entre dois pontos sabendo que cada grau corresponde a 60 milhas náuticas, ou seja, 60X1852 metros.
- Carregar e descarregar o camião. Neste caso assume-se que o contentor é colocado ou retirado do camião. A carga/descarga parcial de um contentor só pode ser efetuada quando o contentor se encontra fora do camião, por exemplo numa estação de distribuição ou numa loja.

2.1.3 Contentores

Os contentores armazenam os packs de bens a transportar e devem guardar informação da sua capacidade e da sua carga. Considere que cada contentor pode levar até 22000Kg de carga com, no máximo, 33 m³ de capacidade cúbica. A aplicação deve garantir que a carga nunca excede qualquer destes valores. Como principais funcionalidades há a referir:

- Identificação do contentor (um número inteiro).
- Possibilidade de carregar e descarregar diferentes packs de produtos (assuma um máximo de 10 packs de produtos).
- Informação do espaço ocupado e do espaço livre em cm³.
- Informação do peso da carga transportada e do peso de carga disponível em Kg.
- Informação diversa sobre os produtos no contentor.

2.1.4 Estações Abastecedoras ou de Distribuição

Os camiões podem ser carregados com contentores nas estações de distribuição. Na sua estrutura básica, estas estações são caracterizadas pelos seguintes elementos:

- Nome/designação.
- Uma posição no mapa, definida por coordenadas geográficas, à semelhança dos camiões.
- Um armazém com os conjuntos de bens a fornecer.
- Contentores vazios para carregamento dos diversos bens. O número máximo de contentores vazios é definido inicialmente para cada uma das estações criadas.

Uma funcionalidade das estações de distribuição é o carregamento de camiões com contentores. Tenha em conta que uma estação abastecedora só pode carregar um camião que esteja efetivamente na sua localização, ou seja, o camião deve ter posição igual à da estação de distribuição.

Antes do camião ser carregado com um contentor, esse contentor deve ser carregado com diversos bens. Neste caso deverá ser disponibilizado um método que carregue parcialmente um contentor com um determinado pack de produtos. Este método deverá receber o pack de produtos a carregar. Tenha em atenção que o produto e a quantidade a carregar devem existir no armazém. Os valores retirados do armazém devem ser atualizados.

Deverá ainda ser implementado um mecanismo que permita a estas estações de distribuição receber os bens que fornecem, incrementando, neste caso, o stock existente no armazém. O respetivo método deverá receber como parâmetro o código do produto e a quantidade recebida.

2.1.5 Lojas

As lojas funcionam de forma semelhante às estações de distribuição, mas facultando a tarefa inversa, ou seja, estas recebem os contentores dos camiões e guardam o bem-recebido nos seus armazéns. Assim, são caracterizadas pelos seguintes aspetos:

- Nome/designação.
- Uma posição no mapa, definido por uma coordenada geográfica, à semelhança dos camiões.
- Um armazém com os bens existentes.

Como funcionalidade as estações de recolha devem descarregar os contentores dos camiões e retirar os packs de produtos que necessitam. Tenha em atenção as validações já apontadas no caso das estações de distribuição.

Dado o espaço limitado, apenas podem existir a qualquer momento na loja um máximo de dois contentores.

Deverá ainda ser implementado um mecanismo que permita a estas estações consumir/vender o bem que recebem, libertando espaço no armazém.

2.1.6 Garagens

Deverá ser criada uma entidade para representar garagens para camiões. Uma garagem é caracterizada pelos seguintes aspetos:

- Nome/Descrição.
- Localização em coordenadas geográficas.
- Um conjunto finito de lugares de estacionamento definido inicialmente para cada garagem, lugares esses que devem ser identificados numericamente, por ordem crescente, começando em 1.
- Cada lugar pode ter um contentor, usado para troca com o contentor que está num camião. Inicialmente nenhum dos lugares tem contentores, apenas serão preenchidos por contentores retirados de camiões.

As garagens devem possuir as seguintes funcionalidades:

- Estacionar um camião, num determinado lugar. Tenha em atenção que deve ser validado se o lugar existe e se não está nenhum outro camião estacionado nesse lugar.
- Registrar a saída de um camião de um determinado lugar. Este deverá ficar vago.
- Inspeccionar o camião estacionado num determinado lugar. A inspeção deve reiniciar o número de quilómetros percorridos no camião e incrementar o número de inspeções efetuadas.
- Trocar o contentor de um camião por um outro, fornecido como parâmetro. É possível não fornecer nenhum contentor no parâmetro, neste caso faz-se apenas a descarga de um contentor.
- Conferir estacionamentos, percorrendo a lista de camiões estacionados a verificar se estes se encontram mesmo na garagem (posição igual à da garagem).

3 Regras de Desenvolvimento e Entrega do Projeto

3.1 Entrega

A data de entrega do projeto é **9 de janeiro de 2017**.

3.2 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 adotadas 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.

Nas funcionalidades desenvolvidas, deverão ser incluídas todas as validações necessárias para impedir um comportamento incorreto do sistema.

Sempre que fizer sentido, os métodos deverão ser responsáveis por apresentar no ecrã mensagens de informação e/ou erro, indicando o processamento que foi feito ao objeto. Por exemplo, quando um camião transpota um determinado bem, deve ser apresentada uma mensagem no ecrã indicando que o camião X viajou entre o elemento Y e o elemento Z.

Nas situações em que for adequado, devem ser facultadas diferentes assinaturas para os métodos implementados.

Os nomes das classes, atributos e métodos deverão ser definidos em língua inglesa. As mensagens apresentadas pela aplicação podem ser apresentadas em português e/ou inglês.

3.3 Constituição de grupos

Cada projeto deverá ser elaborado em grupos de dois alunos, podendo ser desenvolvido individualmente em casos pontuais devidamente justificados. Não serão permitidos, em nenhum caso, grupos com mais do que dois alunos.

Os grupos 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 respetivo docente de laboratório para regularizar a situação.

3.4 Entrega do projeto

- O projeto deverá ser entregue até à data limite especificada por **via exclusivamente eletrónica, utilizando a área dos trabalhos na plataforma Moodle**. 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 do programa, explicando de forma simples as classes criadas, juntamente com os seus atributos e métodos, bem como qualquer detalhe de implementação que necessite de explicações adicionais.
- O código fonte do programa na forma de projeto em BlueJ, com todas as funcionalidades implementadas. **Não existe necessidade de implementação de uma interface com o utilizador.**
- 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** (exemplo: **EI_12345678_87654321**). **O incumprimento das normas de entrega está sujeito a uma penalização de até 1 valor.**

4 Regras e Critérios de Avaliação do Projeto

4.1 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 respetiva. 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 respetivo 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 efetivamente cópias, serão anulados.

4.2 Critérios de Avaliação

Este primeiro projeto será avaliado segundo os seguintes critérios:

Entidades	50%
Bens	5%
Contentores	10%
Coordenadas	5%
Camiões	5%
Estações de distribuição e Lojas	15%
Garagens	10%

Implementação	40%
Estrutura do projeto, definição e relação entre as classes	10%
Estrutura e funcionamento interno das classes	10%
Conhecimento e boa utilização da linguagem	10%
Bom estilo (identificadores, comentários, indentação)	10%

Avaliação qualitativa	10%
Qualidade geral, detalhes de implementação e funcionalidades adicionais	10%

Bom trabalho!