

Trabalho Prático
Programação Orientada por Objetos

ROBOFLEET
automação industrial

Índice

1	Introdução	3
2	O ROBOFLEET	3
2.1	Técnicos	3
2.2	Robots.....	3
2.3	Centro de Comando	3
3	Requisitos funcionais	4
3.1	Técnicos	4
3.2	Robots.....	4
3.2.1	R-Carry (Transporte).....	4
3.2.2	R-Clean (Limpeza).....	4
3.2.3	R-Factory (Produção).....	4
3.2.4	R-Inspect (Inspeção).....	5
3.3	Centro de comando	5
3.4	Outras funcionalidades e restrições	5
3.5	Modos de Funcionamento:	5
4	Regras de Implementação e Avaliação	6
4.1	Constituição de grupos.....	6
4.2	Regras de entrega do projeto	6
4.3	Regras gerais de implementação.....	6
4.4	Fases do projeto e documentos a entregar.....	6
4.5	Regras de avaliação do Projeto.....	6
4.5.1	Cálculo da Nota	7

1 Introdução

O objetivo deste projeto é desenvolver o sistema ROBOFLEET, que permitirá ao utilizador gerir robots num Complexo Industrial–Logístico Automatizado.

O complexo é constituído por várias zonas operacionais, onde robots realizam tarefas de transporte, limpeza, produção e inspeção.

O sistema será desenvolvido segundo os princípios da Programação Orientada por Objetos (POO) utilizando a linguagem Java, e disponibilizará uma interface textual baseada em menus.

2 O ROBOFLEET

O ROBOFLEET disponibiliza operações de registo, consulta, ativação e gestão de técnicos e robots associados ao complexo industrial. Toda a interação será realizada através de menus/submenus em consola.

2.1 Técnicos

Os técnicos podem ficar responsáveis por determinados robots, dependendo do tipo de robot e das respetivas necessidades de operação.

É necessário registar: nif, nome, data de nascimento e especialização.

Cada técnico pode ter várias especializações (Robótica, Manutenção, Sistemas).

2.2 Robots

Os Robots são identificados por um id, marca, modelo, ano de fabrico, bateria (capacidade [Ah] e autonomia [horas]), e motores (quantidade e potencia [w]).

Cada Robot deve saber quantos Robots existem no total e cada um pertence obrigatoriamente a uma zona de operação do complexo: Armazém, Laboratório de Triagem, Linha de Produção 1, Linha de Produção 2. Há ainda a Estação de Carga, onde todos ficam a carregar.

Existem 4 tipos de robots: R-Carry (Transporte), R-Clean (Limpeza), R-Factory (Produção), R-Inspect (Inspeção).

Cada tipo possui regras e capacidades específicas (ver nas próximas secções).

2.3 Centro de Comando

O Centro de Comando deve registar os seguintes dados: id sequencial, conjunto de robots registados e um contador do número total de ordens já enviadas.

O Centro tem um radar que consegue detetar robots na estação de carga, bem como aqueles localizados nas zonas de produção, revelando detalhes técnicos. Para isso, o radar deve estar ativo.

3 Requisitos funcionais

3.1 Técnicos

Os Técnicos têm os seguintes dados: id (sequencial e atribuído automaticamente), nome e idade.

Técnicos são iguais se tiverem o mesmo id e não deve ser possível registar duas vezes o mesmo técnico.

Não podem existir Engenheiros de Robótica com menos de 30 anos.

Todas as listagens devem apresentar técnicos por ordem crescente de nome.

Deve ainda ser possível listar técnicos por ordem crescente de id e ordem decrescente de data de nascimento (mostrando a respetiva idade).

3.2 Robots

Um robot pertence apenas a uma zona e associações múltiplas devem produzir uma exceção.

Os Robots são identificados por um id sequencial e atribuído automaticamente. Dois robots são iguais se tiverem o mesmo id. Devem ainda ter um nome, que deve ser único e não pode ser nulo (deve gerar exceção nesse caso).

Os Robots de uma zona devem ser listados por id crescente e deve ainda ser possível listar robots por ordem crescente de marca e ordem decrescente do ano de fabrico (indicando anos e meses desde o fabrico).

Os Robots só podem ser ativados se tiverem um mínimo de técnicos associados, conforme os requisitos de cada tipo (ver nas seções seguintes).

3.2.1 R-Carry (Transporte)

Robots dedicados ao transporte interno de materiais.

Características: 2 a 4 motores elétricos; 1 a 3 técnicos (obrigatório 1 Técnico de Manutenção); Capacidade de carga (kg) e Holofote auxiliar.

Funcionamento: Várias unidades podem ser ativadas para a mesma zona; Ao concluir a operação, regressam individualmente ao Centro de Comando.

3.2.2 R-Clean (Limpeza)

Robots responsáveis pela higienização industrial e recolha de resíduos.

Características: 1 a 2 motores; Sistema de sucção; 1 a 2 técnicos (obrigatório 1 Técnico de Manutenção) e Luz de inspeção de piso.

Funcionamento: Podem operar simultaneamente na mesma zona; A atividade pode incluir limpeza profunda ou de rotina; Ao concluir a operação, regressam individualmente ao Centro de Comando.

3.2.3 R-Factory (Produção)

Robots da área fabril (Linhas de Produção).

Características: 2 motores; 2 a 4 braços robóticos; 2 a 3 técnicos (obrigatório 1 Engenheiro de Robótica); Sistema de orientação laser.

Funcionamento: Operam na zona fabril (Linha 1 ou Linha 2); Só podem ser ativados se tiverem equipa completa; Podem detetar outros robots da mesma linha através de sensores laser.

3.2.4 R-Inspect (Inspeção)

Robots de análise e controlo de qualidade.

Características: 1 Motor; 1 a 2 técnicos (sem obrigatoriedade de especialização); Câmaras de alta resolução; Scanner térmico e ótico.

Funcionamento: Várias unidades podem ser ativadas para a mesma zona. Quando um deteta o problema, todos os R-Inspect **da mesma zona** regressam ao Centro e ficam estacionados.

3.3 Centro de comando

O centro de comando pode dar ordens de ativação de Robots para atividades nas diferentes zonas do complexo. Os Robots sem atividade encontram-se estacionados no Centro.

O centro tem um radar central que deteta robots estacionados na estação de carga e a operar nas zonas de produção (quando ativado).

3.4 Outras funcionalidades e restrições

Não pode haver registos duplicados de robots ou técnicos.

Um robot não pode ser registado em duas zonas distintas.

Deve ser possível gravar todas as entidades em ficheiro binário (para preservar estado entre execuções).

Deve ser possível exportar uma lista de robots e respetivos dados para ficheiro de texto formatado (txt).

O sistema deve produzir exceções adequadas sempre que regras sejam violadas.

3.5 Modos de Funcionamento:

O menu deverá disponibilizar opções para:

- criar, editar e remover técnicos
- criar, editar e remover robots
- associar/desassociar técnicos
- definir zonas
- ativar robots
- consultar estado do complexo
- listar robots/técnicos
- utilizar radar
- exportar dados

Deverá existir uma função RECOVER, que colocará todos as entidades do sistema com os últimos valores gravados em ficheiro binário.

4 Regras de Implementação e Avaliação

4.1 Constituição de grupos

Cada projeto deverá ser elaborado respeitando os grupos definidos em aula. Possíveis exceções devem ser discutidas com o docente.

4.2 Regras de entrega do projeto

- Cada fase do projeto deve ser entregue até à data estipulada, por **via exclusivamente eletrónica, utilizando o TEAMS**.
- Todos os ficheiros que compõem cada fase deverão ser compactados num único ficheiro em **formato ZIP**, denominado GrupoK_XXXXX_YYYYY_ZZZZZ.zip, em que K é o número do grupo e X, Y e Z são os números dos alunos que constituem o grupo indicado.

4.3 Regras gerais de implementação

- O programa deve ser desenvolvido utilizando a linguagem Java em modo de consola e colocando em prática os conceitos fundamentais do paradigma de programação orientada por objetos, nomeadamente: encapsulamento de dados, classe e objetos, composição e herança de classes, classes abstratas, interfaces, polimorfismo e serialização.
- **Sugestão:** Todas as classes de coleção podem ser implementadas como heranças de classes de coleção parametrizáveis (e.g., `public class ColecaoXYZ extends xxxxx`).

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

- A notação camelCase para o nome das variáveis locais e identificadores de atributos e métodos;
- A notação PascalCase para os nomes das classes e interfaces;
- Utilização de maiúsculas para os nomes das constantes e dos valores enumerados;
- Não utilize abreviaturas nem o símbolo '_' nos identificadores (exceto nas constantes)
- O projeto deve cumprir tudo o que é pedido no seu enunciado, **sendo deixado ao critério do programador qualquer pormenor de implementação que não seja referido** no mesmo e que deve ser devidamente documentado.
- O desenvolvimento dos menus da interface com o utilizador fica ao critério do programador, sendo premiadas a facilidade de utilização.

4.4 Fases do projeto e documentos a entregar

O projeto será desenvolvido em fases, que serão detalhadas nas aulas, ao longo do semestre.

A **defesa/discussão obrigatória** do projeto terá lugar: no dia xx de xx.

4.5 Regras de avaliação do Projeto

- A classificação de cada fase do projeto terá em conta as especificidades dos materiais exigidos e focará a avaliação da qualidade da programação (fatores de qualidade do software), a estrutura do código criado segundo os princípios da programação orientada por objetos, tendo em conta conceitos como: a coesão de classes e métodos, o grau de acoplamento entre classes e o desenho de classes orientado pela responsabilidade, bem como a utilização/conhecimento da linguagem Java.

- No acompanhamento de projeto, será apurada a capacidade de o aluno produzir os materiais apresentados. Nos casos em que essa capacidade não for demonstrada, a nota atribuída será zero valores.
- Todos os projetos serão submetidos a um sistema de **detecção de cópias automático e de inteligência artificial**. Os projetos que forem assinalados como suspeitos, caso se confirme a fraude, **serão anulados**.

4.5.1 Cálculo da Nota

A nota final do projeto será o resultado da média das avaliações realizadas durante as aulas tendo em conta o trabalho individual desenvolvido por cada aluno, bem como a avaliação da defesa dos projetos entregues.