

**Universidade de Aveiro**  
**Departamento de Eletrónica, Telecomunicações e Informática (DETI)**  
**Padrões e Desenho de Software (PDS)**  
**Exemplo de Avaliação Prática - duração: 1h30**

**Nome:**

**Nº Mec.**

**Notas importantes:**

1. Faça login com a conta **sessão1** e password indicada no início da prova.
2. Leia com atenção o enunciado do exame e crie um diagrama de classes na folha de rascunho fornecida.
3. Crie uma pasta com o seu número mecanográfico no Desktop, onde deve deixar o código fonte que será avaliado – código em qualquer outra pasta será ignorado.
4. A Parte C deve ter de ser implementada num ficheiro separado das Partes A e B.
5. Como a solução é curta em termos de linhas de código, implemente todas as classes no mesmo ficheiro (`Sessao1PartAB.java` para as Partes A e B - `Sessao1PartC.java` para a Parte C, respetivamente) – não se esqueça que só pode ter uma classe pública por ficheiro.
6. Deve respeitar as funções `main` fornecidas, não sendo permitidas alterações às mesmas.
7. Deve respeitar o conteúdo e formato das mensagens geradas, tomando como referência o exemplo de execução da página seguinte.
8. Coloque o seu nome e número mecanográfico como comentário no início de cada ficheiro, no campo designado.
9. Será valorizada a submissão de uma solução que compile e possa ser executada, mesmo que incompleta – considere colocar em comentário código não funcional.

Projete e implemente um programa em JAVA para simular um jogo de futebol entre robôs. O programa deve utilizar de forma efetiva vários padrões de software para gerir a criação, interação e comportamento dos objetos necessários.

## **Parte A e B**

**Requisitos:**

1. Criação de Robôs:
  - o Implemente um mecanismo para criar diferentes tipos de robôs: Defender, Attacker e Golie.
  - o Cada robô deve ter um ID e um nome.
  - o Os robôs devem permitir as ações de move e kick, devendo ser atualizado o estado dos robôs com a contagem de kicks e moves.
2. Simulação da Bola de jobo:
  - o A bola usada no jogo deve ter como atributo a cor e deve ser um objeto único no jogo.
3. Integração de Robô Antigo:
  - o Implemente uma solução para integrar uma classe de robôs antigos, que apenas têm como atributo um nome e as ações oldMove e oldKick. Estes deverão ser operados da mesma forma.
4. Registo das ações dos robôs:
  - o Implemente um sistema para registrar num ficheiro com o nome “plays.txt” as ações dos robôs efetuadas até esse momento.

## **Parte C**

**Requisitos:**

1. Comunicação entre robôs:
  - o Implemente um mecanismo para facilitar a troca de mensagens entre os robôs durante o jogo. Todas as mensagens devem ser moderadas por uma BaseStation que deve reencaminhar para todos os robôs as mensagens enviadas, exceto para o robô que a envia.

## **Exemplo de execução:**

PartA: Playing with robots:

```
Defender defender1 is moving
Defender defender1 is kicking
Attacker attacker1 is kicking
Goalie goalie1 is moving
Goalie goalie1 is kicking
Defender defender1 is moving
Ball color: white
OldRobot oldRobot1 is moving
OldRobot oldRobot1 is kicking
OldRobot oldRobot1 is moving
OldRobot oldRobot1 is kicking
```

Part B: Save number of plays from each robot:

File written with the number of plays from each robot

Part C: Base Station:

```
Defender defender1 is moving
Robot goalie1 received the play: move from defender1
Robot attacker1 received the play: move from defender1
Robot oldRobot1 received the play: move from defender1
Attacker attacker1 is kicking
Robot goalie1 received the play: kick from attacker1
Robot defender1 received the play: kick from attacker1
Robot oldRobot1 received the play: kick from attacker1
Goalie goalie1 is moving
Robot defender1 received the play: move from goalie1
Robot attacker1 received the play: move from goalie1
Robot oldRobot1 received the play: move from goalie1
OldRobot oldRobot1 is kicking
Robot goalie1 received the play: kick from oldRobot1
Robot defender1 received the play: kick from oldRobot1
Robot attacker1 received the play: kick from oldRobot1
```

## **Conteúdo do ficheiro gravado “plays.txt”:**

```
Defender defender1 has 2 moves and 1 kicks
NewAttacker attacker1 has 0 moves and 1 kicks
Goalie goalie1 has 1 moves and 1 kicks
Attacker oldRobot1 has 0 moves and 0 kicks
```