

# INSTITUTO SUPERIOR de ENGENHARIA de LISBOA

Licenciatura em Engenharia Informática e Multimédia

1º Semestre Letivo 2019/2020

## Fundamentos de Sistemas Operativos

### 2º Trabalho Prático

#### Objectivos:

- Desenvolvimento de aplicações multitarefa em Java.
- Gestão do ciclo de vida de tarefas.
- Comunicação e sincronização entre tarefas Java.
- Modelo Produtor-Consumidor.

Pretende-se o desenvolvimento de uma aplicação, escrita na linguagem Java, que permite o controle simultâneo de vários robots.

Neste trabalho, pretende-se que existam vários comportamentos do tipo COREÓGRAFO e DANÇARINO. Estes comportamentos são suportados por tarefas Java (por derivação da classe `Thread` ou implementação da interface `Runnable`).

Os comportamentos do tipo COREÓGRAFO geram movimentos aleatórios de dança que são transmitidos aos diferentes DANÇARINOS. No entanto, cada movimento gerado por cada COREÓGRAFO tem de ser entregue de forma atómica a todos os DANÇARINOS.

A especificação das mensagens trocadas entre os COREÓGRAFOS e os DANÇARINOS obedece à sintaxe utilizada no primeiro trabalho:

Número: <code>int</code>	Ordem: <code>int</code>
--------------------------	-------------------------

Onde:

- **Número** – É um campo do tipo `int` e identifica a cardinalidade da mensagem enviada. Por exemplo, a primeira mensagem é identificada pelo número 1, a segunda pelo número 2 e assim sucessivamente.
- **Ordem** – É um campo do tipo `int` com o seguinte significado quanto a valores:

Ordem	Valor
0	Parar( false )
1	Reta( 10 )
2	CurvarDireita( 0, 45 )
3	CurvarEsquerda( 0, 45 )
4	Reta( -10 )
5	Parar (true )

A comunicação entre os COREÓGRAFOS e os DANÇARINOS é suportada por um canal de comunicação (objecto partilhado entre os diferentes comportamentos) baseado/adaptado do **Buffer Circular** apresentado nos *slides* de suporte à Unidade Curricular.

O canal de comunicação tem uma dimensão fixa (32 mensagens). As operações de **escrita** e **leitura** de mensagens do canal de comunicação deverão ser bloqueantes se o canal estiver respectivamente **cheio** ou **vazio**.

A sincronização entre os diferentes comportamentos pode ser implementada com recurso a semáforos ou monitores.

A criação de comportamentos, COREÓGRAFO e DANÇARINO, deve ser efetuada na interface gráfica a desenvolver, por exemplo tal como apresentado na Figura 1.

### Desafio Adicional

Considere igualmente a possibilidade de existir um terceiro comportamento, EVITAR, que trabalha em cooperação com o comportamento DANÇARINO. Neste modo por cada comportamento DANÇARINO existe obrigatoriamente um comportamento EVITAR.

O comportamento EVITAR deverá testar periodicamente o sensor de toque do robot (por exemplo a intervalos de tempo de 200 ms) e quanto a leitura do sensor de toque devolver ativo (valor 1) então o robot para e o comportamento DANÇARINO termina a sua execução.



**Figura 1 – Interface gráfica sugerida para o 2º trabalho**

## Avaliação

A duração do trabalho prático é de 4 aulas práticas. O trabalho é realizado em grupo e está sujeito a avaliação, tendo em conta os seguintes aspetos:

1. Estrutura e organização do relatório;
2. Código implementado;
3. Descrição dos protocolos de comunicação;
4. Diagramas de atividade ou de estado de cada comportamento.

O relatório do trabalho prático e o código implementado devem ser entregues uma semana após a realização da última aula prática. A discussão será realizada na semana seguinte à semana de entrega do relatório.

Os Docentes,  
Carlos Gonçalves e Jorge Pais