

INSTITUTO SUPERIOR de ENGENHARIA de LISBOA

Licenciatura em Engenharia Informática e Multimédia

1º Semestre Letivo 2019/2020

Fundamentos de Sistemas Operativos

1º Trabalho Prático

Objectivos:

- Desenvolvimento de interfaces gráficas em JAVA Swing utilizando o editor gráfico WindowBuilder para o Eclipse.
- Gestão de processos JAVA.
- Comunicação entre processos JAVA com memória partilhada.
- Modelo Produtor-Consumidor.

Pretende-se o desenvolvimento de dois processos JAVA, um processo implementa o comportamento COREÓGRAFO, o segundo processo implementa o comportamento DANÇARINO que conhece a API do robot.

O processo COREÓGRAFO comunica com o processo DANÇARINO utilizando memória partilhada suportada pela classe JAVA `MappedByteBuffer`.

A especificação da sintaxe da mensagem de comunicação entre os dois processos é a seguinte:

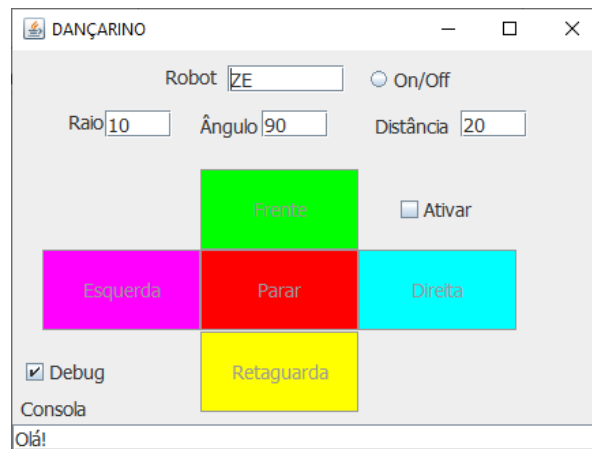
Número : <i>int</i>	Ordem: <i>int</i>
---------------------	-------------------

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.

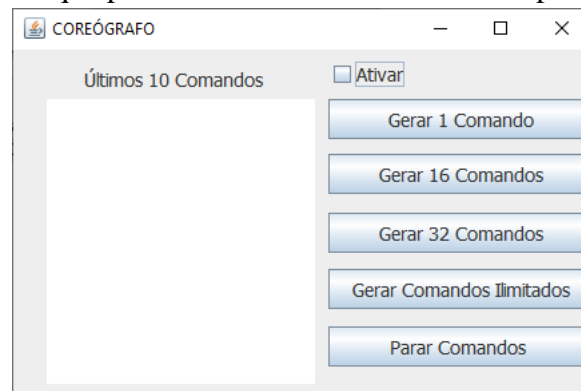
Ordem – É um campo do tipo *int* com o seguinte significado quanto a valores,

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

O comportamento DANÇARINO é um processo JAVA consumidor de mensagens. Este processo lê as mensagens enviadas pelo COREÓGRAFO, guarda-as em memória local para depois as ir executando no robot pela mesma ordem. Este processo conhece a API do robot. A interface gráfica desenvolvida na aula é a interface do DANÇARINO com mais uma *checkbox* designada “Ativar”. Esta *checkbox* permite ativar ou desativar o comportamento.



O comportamento COREÓGRAFO é um processo JAVA produtor de mensagens que gera uma mensagem de 500 em 500 ms quando ativo. A interface gráfica sugerida para este processo inclui uma *TextArea* onde devem ser apresentados os últimos 10 comandos enviados. Quatro botões para envio de comandos e um botão que pára o envio de comandos. E uma *checkbox* “Ativar” que permite ativar ou desativar o comportamento.



O canal de comunicação entre os processos COREÓGRAFO e DANÇARINO deve obedecer ao seguinte modelo:

buffer: <i>MappedByteBuffer</i>
fc: <i>FileChannel</i>
f: <i>File</i> “Comunicacao.txt”

O *buffer* do tipo *MappedByteBuffer* implementa um *ring buffer* que suporta 32 mensagens.

Sugestões para o desenvolvimento do trabalho:

1. Crie uma classe “CanalDeComunicacao” que cria o canal de comunicação e os métodos de leitura e escrita no canal necessários à comunicação entre os processos.
2. Os dois processos devem ter objetos da classe “CanalDeComunicacao” para a concretização da comunicação entre processos ponto a ponto.

Avaliação

A duração do trabalho prático é de **4** aulas práticas, é realizado em grupo e existe avaliação.

A avaliação do trabalho prático tem em conta os seguintes aspetos:

1. Estrutura e organização do relatório;
2. Código implementado;
3. Diagramas de atividade ou de estado de cada processo.

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,
Jorge Pais e Carlos Gonçalves