



Licenciatura em Engenharia Informática e Multimédia

Fundamentos de Sistemas Óperativos

Relatório do Trabalho Prático 1

Docente:

Eng. Jorge Pais

Alunos:

Nº 39205 - Diogo Fernandes

Nº 41543 - Miguel Carreira

26 de Outubro de 2019

Objetivos / Introdução

Para o primeiro trabalho prático da disciplina, foi-nos proposta a realização de uma interface visual de comunicação com os Robot's, bem como desenvolvimento no sentido de garantir a comunicação entre diversos processos JAVA com memória partilhada.

Os processos principais, COREOGRAFO e DANÇARINO, partilham o mesmo recurso de memória que consiste num ficheiro de dados partilhado pelas diversas instâncias de CanalComunicacao, que permite que o COREOGRAFO também seja capaz de enviar comandos ao Robot, ainda que apenas o DANÇARINO tenha acesso á interface do robot, MyRobotLego.

A motivação deste projeto consiste num entendimento a nível de processos concorrentes que partilham o mesmo recurso de memória e como o acesso a este recurso pode ser controlado.

Estrutura Geral

Este projeto consiste em duas interfaces visuais, que recebem inputs do utilizador e os traduzem em comandos para o robot, sejam os inputs 'diretos' da GUI Dançarino, ou os vindos da GUI Coreografo, que envia comandos na forma de objetos da classe Mensagem para o Canal de Comunicação, adicionando-os ao buffer circular sob forma binária, para serem posteriormente lidos pela GUI Dançarino, que os processa e envia para o robot, utilizando os métodos implementados na classe MyRobotLego.

O conteúdo do buffer circular, no momento terminal da aplicação, é conservado sob a forma de um ficheiro de texto, permitindo assim que o robot 'dance' sem serem inseridos novos comandos numa nova execução da aplicação. A alternativa a esta solução seria fazer com que, a cada nova execução, o conteúdo do ficheiro fosse limpo.

Diagrama de Classes

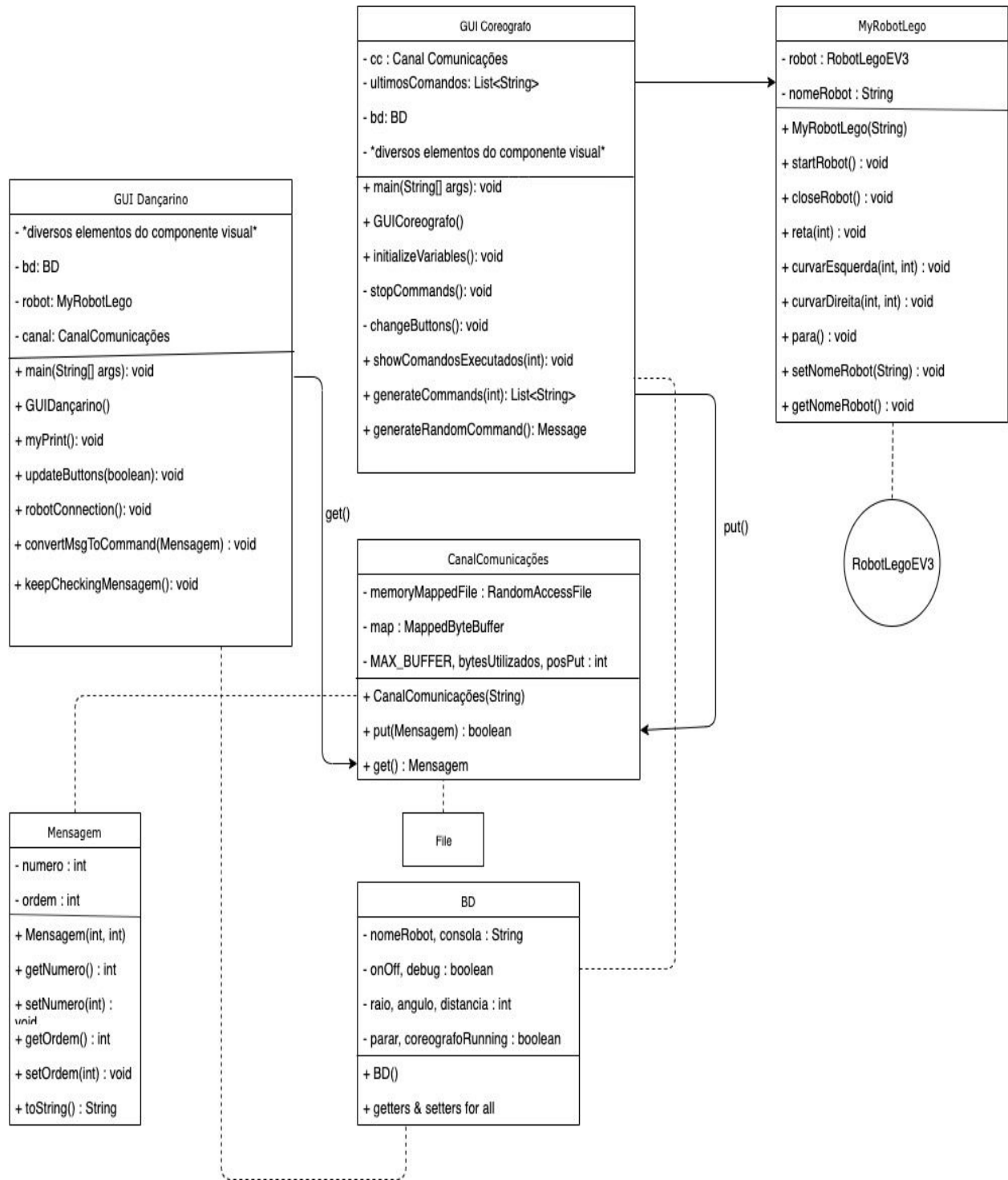
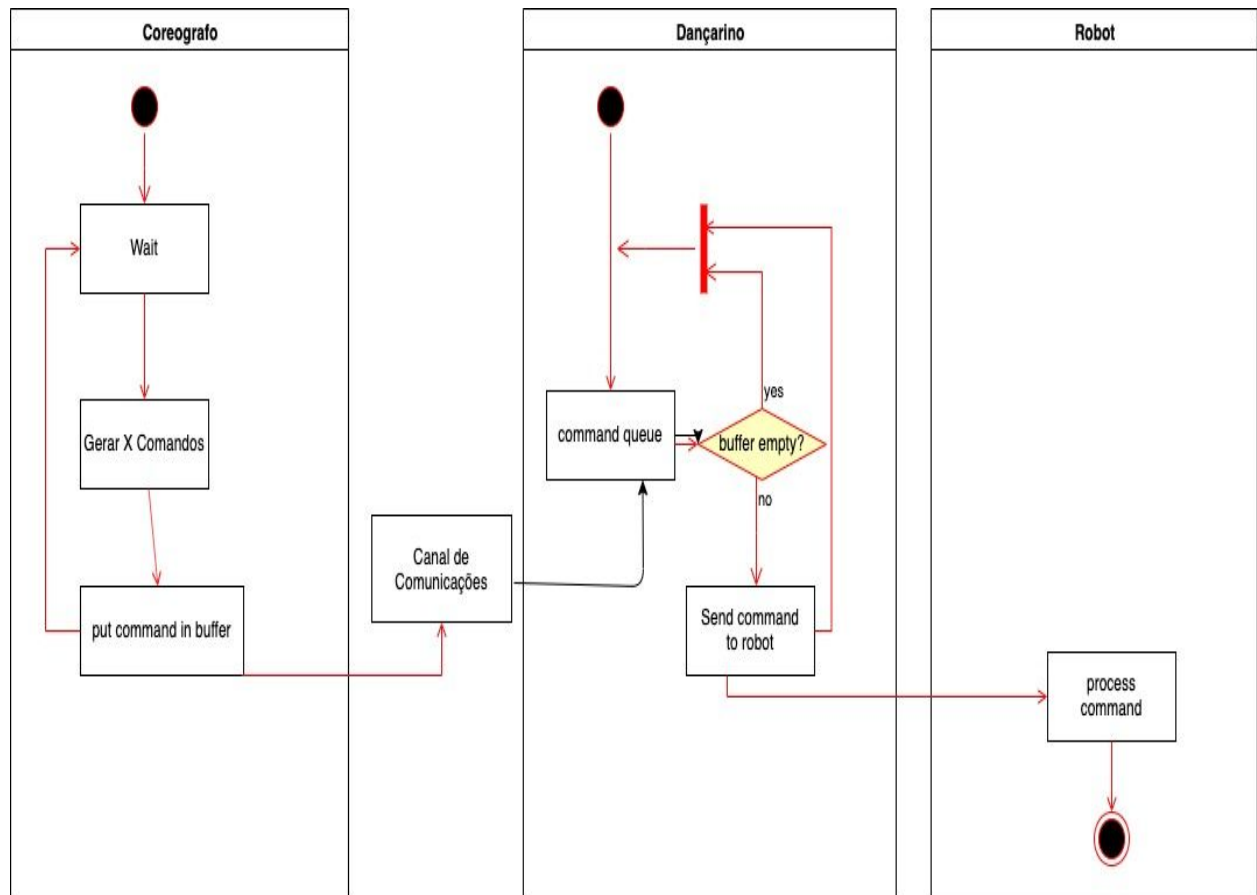


Diagrama de Atividades



Classe “Mensagem”

A classe Mensagem contém a informação necessária á máquina de estados da GUI Dançarino para a criação de um comando do robot.

Essa informação consiste num inteiro de nome “numero” e outro de nome “ordem”, que correspondem ao numero de mensagem enviada e ao código da mensagem em si, respetivamente. A ordem será utilizada na GUI Dançarino de modo a enviar um certo comando ao robot, ou alterar funcionalidades na mesma. O objeto Mensagem é parte integral do funcionamento do Canal de Comunicações.

Classe “CanalComunicacao”

A classe CanalComunicacao consiste na implementação de um buffer circular de mensagens, que recebe uma mensagem e a converte num valor binário que será adicionado ao buffer, utilizando o método ‘putInt()’ da classe MappedByteBuffer, que permite alocar 4 bytes de memória a um valor inteiro. Esta classe gera também um ficheiro de texto contendo registo dos ultimos 32 comandos enviados utilizando este canal de comunicação.

A nível de sincronismo, na nossa implementação, o conceito de sincronismo é facilitado pelo facto de que apenas um processo escreve para o ficheiro e apenas um processo lê os conteudos do mesmo.

O método put(Mensagem msg) adiciona ao nosso buffer circular, um conjunto de 8 bytes, 4 dos quais para o numero de mensagem e outros 4 para o comando a ser efetuado. Logo após, actualiza a variável ‘posPut’.

O método get() posiciona o buffer na posição ‘pos’ que corresponde á posição da ultima mensagem lida e lê a partir daí, os próximos 2 inteiros (4 bytes + 4 bytes). É também neste método que lidamos com os casos particulares envolvidos na leituras de um buffer circular. Caso não haja mensagem para ler, retorna uma mensagem com numero e ordem iguais a zero.

Classe “MyRobotLego”

Esta classe foi criada em resposta á necessidade de efetuar testes ao código fora do contexto das aulas, em que o acesso aos robots era limitado e funciona também, em contexto prático, como um intermediário no contacto com o robot, assim evitando expôr a classe RobotLegoEV3 á interface gráfica do processo.

Esta classe contém um construtor que inicializa o construtor da classe fornecida para utilização dos robots e que guarda na sua instância o nome do robot a ser utilizado.

Conclusões

A nível de desenvolvimento, este projeto ajudou-nos a compreender a necessidade de forma de controlar o acesso a recursos partilhados, como ligações Bluetooth ou escrita em ficheiros. Partindo deste princípio, foi-nos possível criar uma implementação básica de contacto com o Robot (via classe RobotLegoEV3), que permite que o COREOGRAFO envie mensagens ao DANÇARINO, que por sua vez executa os métodos fornecidos pela classe RobotLegoEV3.

A utilização de um RandomAccessFile é justificada com a sua disponibilização de um apontador que permite o posicionamento do 'marker' no ponto após a ultima leitura de uma mensagem.

Bibliografia

- Slides disponíveis no Moodle.
- Documentação JAVA disponível em:
 - RandomAccessFile:
<https://docs.oracle.com/javase/7/docs/api/java/io/RandomAccessFile.html>
 - MappedByteBuffer:
<https://docs.oracle.com/javase/7/docs/api/java/nio/MappedByteBuffer.html>