

# Programação Distribuída

Avaliação prática – 21/11/2011

(versão modificada: 6/11/2012)

Complete o código fornecido (***DistributedFactorial.java***) de modo a que sejam obtidas as seguintes funcionalidades:

- Uma instância da classe ***DistributedFactorial*** encapsula o valor do factorial de um determinado número ( $0! = 1$ ,  $1! = 1$ ,  $2! = 1*2$ ,  $3! = 1*2*3$ , ...,  $n! = 1*2*...*n$ ) obtido através de cálculo distribuído/paralelo, recorrendo a vários processos remotos;
- A classe ***DistributedFactorial*** deve ser instanciada fornecendo como parâmetros: **(1)** o número pretendido; **(2)** um porto UDP; e **(3)** um porto TCP;
- A classe ***DistributedFactorial*** também representa uma ***thread*** que, de forma autónoma, desencadeia as acções necessárias à determinação do resultado pretendido:
  1. [método ***requestWorkers***] Envia a mensagem representada pela constante ***COOPERATION\_REQUEST*** seguida de um espaço em branco e do valor do porto TCP (e.g., "JOIN 5001"), sob a forma de uma cadeia de *caracteres* (i.e., sem serialização de objectos), para o endereço representado pela constante ***BCAST\_ADDR*** e para o porto UDP fornecido;
  2. [método ***acceptWorkers***] Vai aguardando por pedidos de ligação TCP no porto de escuta TCP fornecido no constructor;
  3. [método ***acceptWorkers***] Termina a fase de aceitação de *workers* se deixar de existir qualquer pedido de ligação durante um período de ***TIMEOUT*** milissegundos (método ***acceptWorkers***);
  4. [método ***startWorkers***] Envia a cada *worker*, via ligação TCP estabelecida, o número pretendido, o número total de *workers* e o seu índice (os índices começam em 1), sob a forma de **três objectos serializados** consecutivos do tipo ***Integer***;
  5. [método ***waitPartialResults***] Vai aguardando pela resposta dos *workers*, sob a forma de um objecto do tipo ***Double*** serializado, e, em simultâneo, vai completando o valor do factorial;
- Sempre que ocorre algum problema (e.g., um excepção), a *thread* deve terminar, o valor do atributo ***result*** deve ser posto a -1 e as ligações TCP estabelecidas com os *workers* devem ser encerradas e retiradas da lista (método ***removeWorkers***);
- As excepções que levam à conclusão do processo de cálculo são tratadas apenas no método ***run*** (i.e., são lançadas pelos restantes métodos);
- Depois de calculado, com ou sem sucesso, o valor do factorial encapsulado pela classe ***DistributedFactorial*** não deve poder voltar a sê-lo.