



Threads



Escrita concorrente de nomes

Programe uma nova classe chamada `NameThread` que estende a classe `Thread`. Esta classe deve receber como argumento do construtor um inteiro que representa o seu identificador. Redefina o `run` desta classe de modo a que execute as seguintes instruções 10 vezes:

- escreva o seu identificador na consola
 - adormece um tempo aleatório entre 1 e 2 segundos.
- a. Implemente o método `main` de modo a que crie duas instâncias da classe `NameThread` e as inicie.
 - b. Altere o `main` usando o método `join` da classe `Thread` de modo a que a thread que está a executar o `main` espere pelo fim das outras duas threads antes de escrever na consola que terminou o programa.
 - c. Altere o método `main` para que ao fim de 12 segundos interrompa as duas threads. Quando as threads forem interrompidas devem terminar.

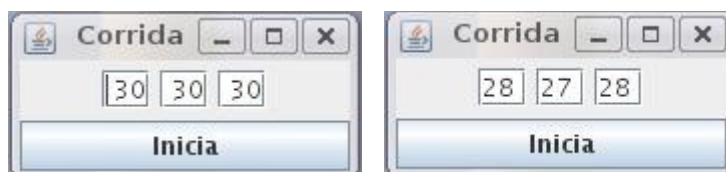


Corrida de cavalos simulada

Considere uma aplicação para simular uma corrida de cavalos. Na corrida participam sempre 3 cavalos que correm de forma independente. A evolução de cada cavalo deve ser visível num campo do tipo `TextField`, tal como se exemplifica na figura em baixo. A interface da aplicação inclui, ainda, um botão para iniciar a corrida. A movimentação dos cavalos deve obedecer aos seguintes requisitos:

- a) o percurso total dos cavalos é constituído por 30 movimentos;
- b) um movimento consiste em subtrair uma unidade aos movimentos que faltam;
- c) após cada movimento o cavalo deve dormir um tempo aleatório.

Exemplo:



Solução corrida cavalos



Gerador de números aleatórios

Escreva um programa com uma janela com um botão com o texto "Stop" (JButton ("Stop")). Quando o programa começa deve criar e iniciar duas threads de classes diferentes:



- A classe **ThreadA** deve gerar constantemente números aleatórios entre 1000 e 9999 até ser interrompida. Os números gerados devem ser escritos na consola. Não deve ser chamado o método Sleep. Nota: deve verificar usando o método interrupted() para verificar se a thread foi interrompida.
- A classe **ThreadB** deve gerar constantemente números aleatórios entre 1 e 9 até ser interrompida. Os números gerados devem ser escritos na consola. Entre a geração de cada número a thread deve dormir (sleep) 0.5 segundo.

Ambas as threads devem manter o registo de quantos números foram já gerados.

Quando o botão Stop for pressionado ambas as threads devem ser interrompidas e terminar. Só após as duas threads terem terminado deve ser escrito na consola o número de valores gerados por cada uma das threads.

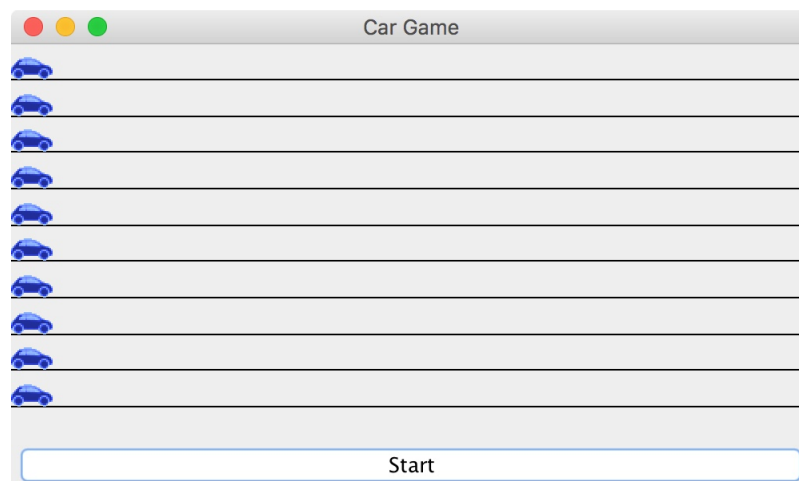


Corrida de automóveis

Arquivos anexados:  [Lab2Ex4.zip](#) (5,421 KB)
 [Lab2Ex4LMota.zip](#) (6,301 KB)

Faça um programa que permita simular uma corrida de vários automóveis.

A sua aplicação deve ter uma janela com dois componentes: um botão "Start" e uma instância da classe Track onde serão desenhados os automóveis durante a corrida.



Cada automóvel deve ser controlado por um processos ligeiros (threads) que vai alterando a sua posição.

Deve usar o padrão de desenho observador-observado para fazer a atualização da pista (Track). Na sua solução deve implementar uma nova classe que seja o observador dos carros. Sempre que são feitas alterações às posições dos carros deve ser feita a notificação dos

observadores. Estes observadores devem chamar um dos seguintes métodos da classe Track:

- método moveCar que recebe o numero do carro e a posição atual desse carro,
- método updateCarPositions(int[] positions) que recebe um array de inteiros que representa as posições atuais de todos os carros.

A classe Track recebe no construtor o número de automóveis que vão participar na corrida e o número de passos necessários para o automóvel chegar à meta: public Track(int numCars, int numSteps).

- a) Comece por implementar uma versão em que o observador escreve para a consola a posição de cada carro.
- b) Altere a sua solução de modo a usar uma instância da classe Track para mostrar a posição dos vários carros.
- c) Altere a sua aplicação de modo a que todos os processos ligeiros que controlam os automóveis terminem quando um dos automóvel chegar à meta.
- d) Altere a sua aplicação de modo a indicar o nº do carro vencedor.

Anexo: Projeto de Eclipse para ser importado.