⇨ Transferência de controlo entre Threads

1 – Pretende-se uma aplicação para gerir o dinheiro em **caixa de um clube recreativo**. Para isso construa as seguintes classes:

a) uma classe *contaBancaria* que deverá permitir:

- consultar o saldo disponível em cada instante;
- simular um levantamento, cada vez que um utilizador pretenda levantar uma quantia menor ou igual à existente;
- simular um depósito.

b) uma classe *financiador* que periodicamente vai depositando quantias num objeto do tipo contaBancaria.

c) uma classe utilizador que periodicamente vai levantando quantias do objeto do tipo contaBancaria.

d) Para testar as classes anteriores construa uma classe teste em que um objeto do tipo *contaBancaria* seja partilhado concorrentemente por um objecto do tipo *financiador* e por pelo menos 3 objectos do tipo *utilizador*.

e) E se quiser suspender o utilizador sempre que o saldo da conta for inferior ao valor do levantamento?

2 - "Readers-Writers Problem"

a) Construa uma classe em Java, RW, que possua um campo inteiro, XPTO, e dois métodos: ler e escrever. O método ler deve devolver o valor da variável XPTO; o método escrever deve adicionar o valor 100 à variável XPTO e seguidamente subtrair o mesmo valor à variável XPTO.

b) Pretende-se que um objecto da classe RW seja partilhado por vários processos (Threads) de dois tipos:

- processos Leitores – que lêem o valor da variável XPTO usando o método ler;

- processos Escritores – que alteram a variável XPTO usando o método escrever.

- Construa as classes Leitor e Escritor. Cada uma destas classes deve ter uma Thread de execução própria em que, num ciclo infinito, vão respectivamente lendo e alterando valores do objecto partilhado.

c) Construa uma classe de teste que crie um objecto do tipo RW, 3 objectos do tipo Leitor e 2 objectos do tipo Escritor. Estude o comportamento do seu programa

d) Pretende-se que modifique as classes anteriores de forma a que os vários processos Leitores possam executar concorrentemente o método ler, mas que quando um processo Escritor executar o método escrever o faça em exclusão mútua. Isto é, quando um processo está a escrever, nenhum outro pode em simultâneo ler ou escrever a variável XPTO.

3 - Suponha a classe exemplo esquematizada abaixo.

```
Class Exemplo {
        ...
    public void mX(){ ...}
  }
```

Supondo que se pretende construir uma aplicação cliente-servidor, em que o processo servidor contém um objecto partilhado do tipo Exemplo e para cada processo cliente que lhe acede lança uma nova thread que irá executar o método mX.

Construa a classe ThreadExemplo de que será criada uma instância sempre que um cliente acede ao servidor. Uma instância de ThreadExemplo terá uma sequência de execução própria, deverá invocar o método mX do objecto partilhado e garantir que nunca haverá mais de três clientes em simultâneo a executar o método mX. Quando um quarto cliente tenta executar o método, a thread correspondente deverá ser suspensa até que menos de três clientes estejam a executar mX.

Que modificações terá de fazer na classe Exemplo.