MÉTODOS DE PROGRAMAÇÃO II

1º ANO/2º SEMESTRE - LECOM

EXAME de 1ª CHAMADA – 17 de Junho de 2005
Cotação - 20 valores
Duração - 2h00m

Apresentam-se em seguida as definições principais de 3 classes de JAVA que irão permitir implementar as operações fundamentais de um sistema de gestão de contas bancárias. A classe **Conta**, apresentada a seguir, representa toda a informação necessária sobre uma dada conta bancária, designadamente: número de conta, titular, saldo actual e lista de movimentos efectuados. Os métodos daX() e mudaX(..) para cada variável de instância de cada classe devem assumir-se como sendo pré-definidos, logo já existentes.

```
public class Conta implements Serializable, Cloneable {
        private String numConta;
        private String titular;
        private double saldo;
        private ArrayList movimentos; // instâncias de Movimento

        public Conta() {
           numConta = ""; titular = ""; saldo = 0.0;
           movimentos = new ArrayList();
        }

        public Conta(String num, String tit, double saldo,
                        ArrayList movs) {
             numConta = num; titular = tit; this.saldo = saldo;
             movimentos = new ArrayList(); Movimento mov = new Movimento();
             for(Iterator it = movs.iterator(): it.hasNext();) {
                mov = (Movimento) it.next();
                movimentos.add((Movimento) mov.clone());
             }
        }
        // outros métodos cf. daX(), mudaX(..), toString(), clone() estão disponíveis
}
```

Os movimentos (diversas operações) realizados sobre uma conta são representados por instâncias da classe **Movimento** definida como:

```
public class Movimento implements Serializable, Cloneable {
        private String entidade;
        private double valor;

        public Movimento() {
           entidade = ""; valor = 0.0;
        }
        public Movimento(String ent, double val) {
           entidade = ent; valor = val;
        }
        // e daX(), mudaX(..), toString() e clone(), etc.
}
```

Chegamos assim à classe principal que se pretende implementar, a classe **Banco**. A classe **Banco** é representada por um **HashMap** que associa um *número de conta bancária* a uma instância de **Conta**, ou seja, *a uma determinada conta bancária*, cf.:

```
public class Banco implements Serializable, Cloneable {

        private HashMap contas;
        // Trata-se um HashMap de NumConta -> Conta

        public Banco() {
          contas = new HashMap();
        }

        // métodos de instância a definir neste teste
}
```

A classe **Banco** deverá agora ser completada com um conjunto de métodos que permitam realizar as mais diversas operações sobre as várias contas bancárias registadas. Estes métodos devem ser, idealmente, robustos, ou seja, lançando excepções caso não possam por qualquer razão ser executados correctamente. Caso não consiga implementar tal requisito, apresente o código que implementaria sem usar excepções.
São as seguintes as funcionalidades da classes Banco a implementar:

- Método que determina o número total de contas do banco;
- Método que dado um número de conta garantidamente correcto devolve o saldo dessa conta;
- Método que insere uma nova conta bancária no banco;
- Método que devolve uma conta bancária dado o seu número;
- Método que calcula o saldo total de todas as contas bancárias;
- Método que determina o número da conta bancária com maior saldo actual;
- Método que devolve um ArrayList dos movimentos de uma conta cujo número é dado;
- Método que devolve um conjunto com os nomes dos titulares das contas;
- Método toString() da classe Banco;
- Método que grava toda a informação do banco num ficheiro de texto;

Prof. F. Mário Martins