

# Ficha Prática 6

António Nestor Ribeiro, António Fernandes, F. Mário Martins  
{anr,af,frm}@di.uminho.pt  
PPIV (LESI)

2003/04

## Objectivos

1. Utilização de HashMaps como implementação de uma estrutura de dados dinâmica.
2. Iterators como mecanismo de visita de uma colecção.
3. Exercícios.

## Iterators

Um `iterator` é uma estrutura que permite percorrer uma colecção. Na nova arquitectura das colecções do Java, o `Iterator` toma o lugar da `Enumeration`, devido a ser uma implementação mais recente e optimizada, e apresenta como principais diferenças:

1. permitir que quem cria o iterador remova elementos da colecção. A `enumeration` apenas permitia visitas sem alterar, ou remover, os elementos;
2. os nomes dos métodos foram revistos (e encurtados).

No exemplo da carteira de acções utilizando uma instância de `Vector`, um dos construtores apresentava a seguinte implementação:

```
public CarteiraAccoes(String dono, Vector listaDeAccoes)
{
    AccaoBolsa accao; Object temp;
```

```

    titular = dono;
    carteira = new Vector();
    for(Enumeration e = listaDeAccoes.elements(); e.hasMoreElements();) {
        temp = e.nextElement();
        if(temp instanceof AccaoBolsa) carteira.addElement(temp);
    }
}

```

Este mesmo construtor pode ser implementado recorrendo à utilização de um `Iterator` com o seguinte código:

```

public CarteiraAccoes(String dono, Vector listaDeAccoes)
{
    AccaoBolsa accao; Object temp;
    titular = dono;
    carteira = new Vector();
    for(Iterator it = listaDeAccoes.iterator(); it.hasNext();) {
        temp = it.next();
        if(temp instanceof AccaoBolsa) carteira.addElement(temp);
    }
}

```

Nota: é necessário ter em atenção que o método `public Object next()`, devolve um `Object` como tipo estático, logo é necessário fazer o *casting* para o tipo desejado, sempre que se pretendam enviar métodos ao objecto.

## HashMap

Um `HashMap` é uma implementação de uma colecção em que as entradas são pares de chave e valor. O objecto chave está associado ao objecto valor, sendo que não podem existir repetições entre as chaves, bem assim como uma chave aponta apenas para um objecto valor.

Como métodos mais interessantes de `HashMap` podem ser referidos:

- `HashMap` - cria um `HashMap` vazio;
- `public Object get(Object key)` - dada uma chave devolve o objecto por ela apontada (caso exista a chave);
- `public boolean containsKey(Object key)` - devolve `true` se a chave existir;

- `public Object put(Object key, Object value)` - insere uma nova associação chave/valor. Se a chave já apontava para um objecto valor, então este método efectua a alteração;
- `public int size()` - devolve o número de associações chave/valor existentes;
- `equals`, `toString` e `clone` - com o comportamento esperado;
- etc.

## Exercícios

1. Compare a sua solução da ficha prática 5 com a solução presente na página da disciplina, em que encontra a Carteira de Acções implementada à custa de `Vector` e `ArrayList`.
2. Implemente a carteira de acções recorrendo a um `HashMap`, em que a chave deve ser o nome da empresa.
3. Crie uma implementação completa com todos os métodos necessários, nomeadamente `equals`, `toString` e `clone`.