

Ficha Prática 5

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

2004/05

Objectivos

1. Colecções: continuação do estudo do package java.util.
2. A classe HashMap.
3. Exercícios.

A classe 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 Object get(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. Considere as classes `CartaoCredito` e `Carteira`, desenvolvidas nas aulas anteriores.

Modifique a implementação de `Carteira`, por forma a que a estrutura do seu estado interno que guarda os cartões de crédito seja uma `HashMap`.

O comportamento da classe `Carteira` consiste nos seguinte métodos:

- Guardar mais um cartão;
 - Retirar o cartão da posição X da carteira. Atenção que tal só é possível se o saldo devedor do cartão for igual a zero. Este método faz sentido nesta nova implementação de `Carteira`? Justifique!
 - Determinar quantos cartões estão na carteira;
 - Determinar o montante em débito na carteira?
 - Apresentar sob a forma de `String` todos os movimentos de cartões actualmente na carteira;
 - Representar sobre a forma de `String` a carteira (com informação dos cartões e respectivo saldo devedor). Deverá também apresentar como resumo o total em débito.
2. Considere que se pretende modelar uma biblioteca de entradas multimédia. Nessa biblioteca existirá informação sobre existências de items multimédia que vamos adquirindo, ou apenas tomando conhecimento (para pedir no Natal!).

Embora tendo a certeza que novas aquisições vão ser feitas, neste momento a biblioteca contém apenas dois tipos de entradas: CDs e DVDs. A funcionalidade a disponibilizar pela classe `BibMultimedia` baseia-se nos seguintes requisitos:

- Deve permitir acrescentar um novo CD ou DVD às entradas já existentes.
- Deve permitir procurar um CD e um DVD pelo seu nome.
- Deve permitir procurar CDs e DVDs dado um nome de um artista para os CDs ou o nome do realizador para os DVDs.

- Deve permitir listar (devolver como String) toda a informação existente tanto nos CDs como nos DVDs.

Um CD pode ser categorizado como tendo:

- um título;
- um artista (ou nome da banda);
- número de músicas;
- tempo total das músicas do disco;
- uma flag indicando que se possui uma cópia do mesmo;
- um comentário.

Para um DVD guarda-se a seguinte informação:

- o título;
- o nome do realizador;
- o tempo total do filme;
- uma flag indicando que se possui o filme;
- um comentário.

Para resolver este exercício cumpra as seguintes etapas:

- (a) Construa as classes CD e DVD.
- (b) Implemente a classe que modela a biblioteca. Use duas Hashmap, uma para guardar os CDs e outra para os DVDs;