

Ficha Prática 4

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

2004/05

Objectivos

1. Apresentação dos packages `java.lang` e `java.util`.
2. As classes `ArrayList` e `Vector`.
3. Exercícios.

Classes a investigar

- `java.lang`
 1. `Number`, `Boolean`, `Character`, `Math`
 2. `String` e `StringBuffer`
- `java.util.Vector` e `java.util.ArrayList`

Exercícios

1. Escreva uma classe `CartãoCrédito` cujas instâncias deverão ter um comportamento semelhante ao do (demasiado?) conhecido cartão.
Cada instância deverá possuir, ao ser criada, um titular, um número único de 12 dígitos, um mês e ano de validade e um valor máximo de débito autorizado.
Igualmente deve ser guardada até ao momento o montante dispendido e um histórico dos movimentos. Por questões de simplicidade considere

que um movimento é uma string com o valor e uma descrição (ex: “30EUR - Bilhete de Futebol”). Cada instância deverá ser capaz de responder, entre outras, às seguintes mensagens:

- (a) `int saldo()` – apresentar o saldo do cartão, isto é, a diferença entre o montante gasto e o limite de endividamento.;
- (b) `void pagarCrédito(int pag)` – efectuar um pagamento, isto é abater ao montante em dívida.
- (c) `String gastar(int quantia, String descr)` – que permite registar um movimento.
- (d) `String getMovimentos()` – que apresenta na forma de uma String a lista de movimentos efectuados.
- (e) `String toString()` – apresentar, como String a informação do cartão.

2. Considere agora que pretende reproduzir a estrutura de uma carteira com espaço para cartões (considere que é uma carteira com slots infinitos...para guardar muitos cartões de crédito!!!). A carteira deve ter como informação os dados do dono (nome, morada, número de telefone), bem assim como os cartões que ele possui.

Modele esta classe, `Carteira`, recorrendo à utilização de `ArrayList`

Sobre uma carteira deve ser possível efectuar, entre outras, as seguintes operações:

- Guardar mais um cartão;
- Listar todos os cartões existentes;
- Retirar o cartão da posição X da carteira. Atenção que tal só é possível se o saldo devedor do cartão for igual a zero;
- 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.

3. Crie uma nova versão da classe `Carteira`, recorrendo a uma instância de `Vector` para guardar os cartões.

- (a) Implemente todos os métodos feitos nas versões anteriores.
 - (b) Quais as principais diferenças que encontra entre uma e outra implementação? Note que de um ponto de vista externo a API (conjunto de métodos visíveis) não se alterou.
4. Recorrendo à classe `Circulo` anteriormente feita, implemente um classe `ConjCirculos`, que permite gerir uma colecção de círculos. Utilize um `Vector` para guardar os círculos.

Implemente os seguintes métodos:

- (a) Adicionar um novo círculo;
- (b) Remover o círculo da posição central `X`;
- (c) Determinar o ponto central do círculo da posição `X`;
- (d) Devolver sobre a forma de `String` a representação do conjunto de círculos.