Programação Orientada por Objectos

2013/2014

Trabalho de Laboratório – Curso EI

9

Objectivos:

- Exceções e Entradas / Saídas em Java.

Programas:

- Construção de um programa para gerir um cartão "porta moedas".

Regras de implementação:

- Criar uma aplicação Java de consola.
- **Implementar o código necessário a testar no fim de cada nível, onde as excepções devem ser capturadas e tratadas, caso tal seja aplicável.**
- Use as convenções de codificação adoptadas para a linguagem Java (ver Notas).
- Na classe do programa não coloque atributos nem crie qualquer outro método para além do *main*.

Implementação:

Nível 1:

- Crie uma classe **CartaoPortaMoedas** que deve guardar a seguinte informação:
  - id do cartão com 10 dígitos
  - código de acesso com 4 dígitos
  - nome do utilizador
  - saldo do cartão
- Deverá definir um construtor que inicializa o código de acesso através de uma função Random (nota: utilize a classe java.util.Random para o efeito).
- Defina o método **toString** que devolve uma String com toda a informação sobre o cartão, excepto a password.

Nível 2:

- Crie uma classe **CartaoException** do tipo *Exception*, que deve ter um construtor que recebe como argumento uma mensagem de erro.
- Crie um método **alterarCodigo** que recebe o código antigo e o código novo e caso o código antigo não coincida com o que está no cartão gere uma excepção com a mensagem - "Código invalido";
- Crie um método para **obter o código**, mas que só devolve o código de acesso uma vez. Caso se tente obter o código mais que uma vez o método deve gerar uma excepção com a Mensagem – "Acesso não permitido");
  Nota: Para implementar este método terá que acrescentar mais atributos à classe **CartaoPortaMoedas**.

Nível 3:

- Crie um método **carregarCartao** que deverá gerar uma excepção caso se tente carregar o cartão com valores negativos. A mensagem de erro deve ser: - "Montante Inválido"
- Crie um método **pagarCartao** que deverá gerar uma excepção:
  - caso tente pagar um montante superior ao saldo do Cartão, a mensagem de erro deve ser: - "Saldo insuficiente.
  - caso o código de acesso seja incorrecto. A mensagem de erro deve ser: - "Código invalido"

Nível 4:

- Crie uma classe **Ficheiro** para escrita e leitura de informação em modo de texto e que contenha os seguintes métodos:
  - **criar** - permite a criação e a abertura de um ficheiro de escrita
  - **abrir** - permite a abertura do ficheiro para leitura..
  - **existeFicheiro** - verifica a existência de um ficheiro.
  - **escreverLinha** – recebe uma string como parâmetro e escreve-a para o ficheiro.
  - **lerLinha** - permite ler uma linha de texto do ficheiro e retorna essa linha.
  - **fecharFicheiroEscrita** - fecha o ficheiro de escrita.
  - **fecharFicheiroLeitura** - fecha o ficheiro de leitura.

Nível 5:

- Crie uma classe **GestaoCartoes** que guarda objectos do tipo **CartaoPortaMoedas**. Esta classe deve guardar o conjunto de Cartões num *array* e deve incluir um método chamado **adicionarCartao** que recebe um CartaoPortaMoedas e o acrescenta ao array. Este método deve verificar se o cartão já existe (através do atributo "id" do cartão).
- Usando os métodos da classe Ficheiro, implemente os seguintes métodos:
  - **guardarCartoes** – que guarda em ficheiro a lista de cartões existentes no array da classe CartaoPortaMoedas.
  - **lerCartoes –** lê do ficheiro todos os cartões para o array existente na classe GestaoCartoes. Só deve de inserir cartões que ainda não existam no array (verificar o "id" do cartão).
- Percorra o *array* existente na classe GestaoCartoes fazendo uma listagem com a descrição de todas as cartões existentes.

Notas:

Para os identificadores siga as convenções adoptadas normalmente, em particular:

1) A notação camelCase para o nome das variáveis locais e identificadores de atributos e métodos.
2) A notação PascalCase para os nomes das classes.
3) Não utilize o simbolo '_', nem abreviaturas nos identificadores.