Prática 5 Classes

Tópicos

- Classes, instâncias e metodologias orientadas a objetos
- Construtores, atributos e métodos
- Métodos especiais (`toString(), equals(), get…(), set…(), ...`)

Exercício 5.1

Respeitando os conceitos de encapsulamento, implemente classes que permitam modelar as seguintes formas geométricas:

- Círculo, caracterizado por um *centro* e *raio*;
- Triângulo, caracterizado pela dimensão dos seus lados (*lado1, lado2* e *lado3)*;
- Retângulo, caracterizado por *comprimento* e *altura*.

Garanta ainda as seguintes especificações:

a) construa a classe Ponto;

```
public class Ponto {
   private double x;
   private double y;
   public Ponto(double x, double y) { .. }  // completar
   public getX() { .. }
   public getY() { .. }
   public String toString { .. }
   // ..
}
```

b) crie classes que representem cada uma das figuras geométricas, implementando construtores e métodos adequados para cada classe.

c) adicione todos os métodos especiais importantes (`toString(), equals(), get…(), set…(), ...`);

d) implemente um método para calcular a área de cada tipo de figura;

e) implemente um método para calcular o perímetro de cada tipo de figura;

f) implemente um método para verificar se os dois círculos se intercetam;

g) implemente um programa que lhe permita testar todas as classes criadas.

Exercício 5.2

Pretende-se construir um sistema de informação simplificado para a gestão da biblioteca de uma universidade. A biblioteca contém um catálogo de livros e um conjunto de utilizadores (só alunos). Todos os utilizadores são identificados pelo seu número mecanográfico, nome e curso. Os livros são caraterizados por um ID (numérico e sequencial, começando em 100), título e tipo de empréstimo (CONDICIONAL ou NORMAL). Comece com as definições seguintes:

```
public class Utilizador {
   private String nome;
   private int nMec;
   private String curso;
```

```
public class Livro {
    private int id;
    private String titulo;
    private String tipoEmprestimo;
```

```
    // .....
}
```

```
    // .....
}
```

Faça uso de modificadores de acesso para garantir que todos os atributos das classes não estão acessíveis do exterior. Em caso de necessidade, defina novos atributos para responder aos requisitos do enunciado. Teste as classes desenvolvidas com o programa seguinte:

```
import java.util.ArrayList;

public class Ex52 {

   public static void main(String[] args) {

      // Para o conjunto de Livros vamos criar um vetor de 10 posições
      // Este vetor tem uma dimensão fixa pelo que se for necessário guardar
      // mais livros, teremos de criar um vetor de maior dimensão.
      Livro catalogo[] = new Livro[10];
      catalogo[0] = new Livro("Java 8", "CONDICIONAL");
      catalogo[1] = new Livro("POO em Java 8");
      catalogo[2] = new Livro("Java para totós", "NORMAL");
      System.out.println("ID = " + catalogo[1].getId() + ", "
                              + catalogo[1].getTitulo());
      catalogo[2].setTipoEmprestimo("CONDICIONAL");

      for (int i = 0; i < catalogo.length; i++) { // usando o indice do vector
         if (catalogo[i] != null) // porque o vector catalogo não está cheio
            System.out.println(catalogo[i]);
      }

      // Para o conjunto de utilizadores usamos a classe java.util.ArrayList
      // É uma implementação de um vetor com tamanho variável
      ArrayList<Utilizador> alunos = new ArrayList<>();
      alunos.add(new Utilizador("Catarina Marques", 80232, "MIEGI"));
      alunos.add(new Utilizador("Joao Silva", 90123, "LEI"));
      alunos.get(1).setnMec(80123);

      for (Utilizador u : alunos) { // usando foreach
            System.out.println(u);
      }
   }
}
```

Cujo resultado da sua execução deve ser:

```
ID = 101, POO em Java 8
Livro 100; Java 8; CONDICIONAL
Livro 101; POO em Java 8; NORMAL
Livro 102; Java para totós; CONDICIONAL
Aluno: 80232; Catarina Marques; MIEGI
Aluno: 80123; Joao Silva; LEI
```

Exercício 5.3

Utilizando as classes desenvolvidas no exercício anterior, implemente um programa que permita gerir os utilizadores e empréstimos numa biblioteca. Comece por construir, de uma forma interativa, o seguinte menu:

```
1 - inscrever utilizador
```

```
2 - remover utilizador
3 - imprimir lista de utilizadores
4 - registar um novo livro
5 - imprimir lista de livros
6 - emprestar
7 - devolver
8 - sair
```

Condições adicionais:

a) Recomenda-se que as operações de empréstimo e devolução sejam efetuadas com base no ID do livro e no número mecanográfico do aluno.

b) Cada aluno só poderá requisitar simultaneamente um máximo de 3 livros. Deve modificar a classe utilizador para poder guardar os IDs dos livros requisitados, bem como a classe livro para indicar a sua disponibilidade.

c) Para simplificar, considere que apenas existe uma cópia de cada livro e que os livros com tipo de empréstimo CONDICIONAL não podem ser requisitados.

d) Para guardar o catálogo de livros e a lista de alunos, utilize vetores, considerando que no máximo a biblioteca pode ter 100 livros e 100 utilizadores.