

Aula prática nº 4

Tópicos

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

Exercícios

1. Implemente um programa que permita gerir figuras geométricas: círculo, quadrado e retângulo. Cada figura é caracterizada por dois atributos obrigatórios: cor (String), centro em sistema de coordenadas bidimensional (2D) e atributos adicionais que especificam as dimensões da figura (raio para círculo, dois lados para retângulo, um lado para quadrado). Tenha em atenção os conceitos de encapsulamento e visibilidade.

Garanta as seguintes especificações:

- a. crie uma classe própria para suportar cada uma das figuras;
- b. utilize a classe `Ponto` como atributo das várias classes que implementam as diferentes figuras (composição);
- c. tenha em atenção os conceitos de encapsulamento e visibilidade (ex. para a classe `Ponto`)

```
public class Ponto {  
    private double x;  
    private double y;  
    // .....  
}
```

- d. Implemente diversos construtores:

```
public Circulo(double x, double y, double r) {...}  
public Circulo(Ponto centro, double r) {...}  
// O mesmo para as outras figuras  
public Ponto(double x, double y) {...}
```

- e. adicione todos os métodos especiais importantes (`toString()`, `equals()`, `get...()`, `set...()`, ...);
- f. implemente um método para calcular área de cada tipo de figura;
- g. implemente um método para calcular perímetro de cada tipo de figura;
- h. implemente um método para verificar se os dois círculos se intercetam;
- i. implemente um programa que lhe permita testar todas as classes criadas.

2. Pretende-se implementar 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 caracterizados 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 {
    String nome;
    String curso;
    int nMec;
    // .....
}

public class Livro{
    int id;
    String titulo;
    String tipo;
    // .....
}
```

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:

```
public static void main(String[] args) {

    Livro catalogo[] = new Livro[10];

    int nLivros = 0;

    Utilizador alunos[] = new Utilizador[10];

    int nUtilizadores = 0;

    catalogo[nLivros] = new Livro("Java 8", "CONDICIONAL");

    nLivros++;

    catalogo[nLivros] = new Livro("POO em Java 8");

    nLivros++;

    catalogo[nLivros] = new Livro("Java para totós", "NORMAL");

    nLivros++;

    catalogo[2].setTipo_emprestimo("CONDICIONAL");

    System.out.println("ID = " + catalogo[1].getId() + ", " + catalogo[1].getTitulo());

    alunos[nUtilizadores] = new Utilizador("Catarina Marques", 80232, "MIEGI");

    nUtilizadores++;

    alunos[nUtilizadores] = new Utilizador("Joao Silva", 90123, "LEI");

    nUtilizadores++;

    alunos[1].setNmec(80123);

    for(int i = 0 ; i < nLivros ; i++)

        System.out.println(catalogo[i]);

    for(Utilizador u : alunos){

        if(u == null)

            break;

        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
```

3. Utilizando as classes desenvolvidas no exercício anterior, implemente um programa que, de uma forma interativa (recorrendo a um menu), permita gerir a biblioteca, utilizadores e empréstimos. 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.

O programa deve permitir a inscrição e remoção de utilizadores. Relativamente aos livros, o programa deve permitir a introdução de livros, efetuar um empréstimo e permitir a devolução dos mesmos. Para simplificar, considere que apenas existe uma cópia de cada livro e que os livros com tipo de empréstimo CONDICIONAL não podem ser requisitados.

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

Faça um programa que, de uma forma interativa (recorrendo a um menu), permita gerir a biblioteca, utilizadores e empréstimos. O programa deve permitir a inscrição e remoção de utilizadores. Relativamente aos livros, o programa deve permitir a introdução de livros do sistema, verificação de disponibilidade de determinado item, empréstimo e devolução de livros. Recomenda-se que as operações de empréstimo e devolução sejam efetuadas com base no ID do livro e no número mecanográfico do aluno.

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