

# PROGRAMAÇÃO ORIENTADA A OBJECTOS AVANÇADA

## Classes e Objectos

### AULA PRÁTICA

1. Escreva um programa que leia 3 inteiros do teclado e mostre o maior.

- ☐ Definição de classes instanciáveis
- ☐ Protecção de variáveis de instância
- ☐ Métodos construtores
- ☐ Métodos de consulta e modificação de variáveis de instância
- ☐ O método toString

2. Pretende-se implementar a classe Contador.

a) As instâncias da classe Contador deverão representar contadores do tipo inteiro capazes de responder a um conjunto de mensagens que implementam as seguintes operações:

- ☐ criar um Contador com valor inicial igual a zero;
- ☐ criar um Contador com valor inicial igual a um valor dado pelo utilizador;
- ☐ incrementar o Contador de uma unidade;
- ☐ incrementar o Contador de um valor dado pelo utilizador;
- ☐ decrementar o Contador de uma unidade;
- ☐ decrementar o Contador de um valor dado pelo utilizador;
- ☐ consultar o valor do Contador ;
- ☐ implemente ainda o método **toString**.

b) Construa uma classe de Teste para verificar a correcção da classe anterior.

c) Depois de testar o método toString estude o que acontece quando numa instrução de escrita coloca apenas o nome do objecto sem lhe enviar a mensagem toString. Finalmente, transforme em comentário o método toString e volte a executar o programa anterior. O que acontece?

d) Construa um programa que gere aleatoriamente N valores inteiros no intervalo de [-100, 100[ sendo o valor de N dado pelo utilizador. Usando objectos da classe Contador o programa deverá contar quantos valores gerados são positivos e quantos são negativos.

3. Considere a classe Espetaculo. Um espetáculo é um evento que tem um **nome**, tem uma certa **capacidade** (isto é, o número máximo de pessoas que pode assistir ao espetáculo) e ao qual assiste um certo número de **espetadores** que não poderá ser superior à capacidade. Um espetáculo tem ainda um **custo** que corresponde ao preço do bilhete para assistir ao espetáculo.

a) Defina o cabeçalho e os atributos da classe Espetaculo.

b) Defina o construtor que recebe como parâmetros o nome, a capacidade e o custo do Espetáculo.

c) Construa os getters e setters para cada atributo da classe.

d) Construa o método toString para a classe Espetaculo.

e) Para a classe Espetaculo construa o método comprarBilhete. O método deverá adicionar uma unidade ao número de espectadores caso ainda haja lugares disponíveis. Nesse caso, o método deve devolver como resultado a String “Tem a pagar €€€”, onde €€€ deve ser substituído pelo custo do bilhete. Se já não houver lugares disponíveis, deverá devolver a mensagem “Espetáculo esgotado”

f) Construa uma classe para testar a classe Espetaculo.

□ A classe String

**4.** Construa a classe Disciplina esquematizada abaixo:

```
public class Disciplina {
    private int codigo;
    private String nome;
    private String prof;

    public Disciplina ( ) {
        codigo = 0;
        nome = " ";
        prof = " ";
    }
    public Disciplina (int codigo, String nome){
        this.codigo = codigo;
        this.nome = nome;
        this.prof = " ";
    }
    ...
}
```

a) Para a classe Disciplina construa o método “toString”.

b) Para a classe Disciplina construa os métodos de consulta (getters) e de modificação (setters) para todos os atributos.

c) Para a classe Disciplina construa um método que verifique se o nome da disciplina contém uma substring dada pelo utilizador.

d) Para a classe Disciplina construa um método que verifique se o seu código está num dado intervalo do tipo [inicio, fim[;

**5.** Construa uma classe de teste para a classe Disciplina.

**6.** Suponha uma classe que representa as 3 cadeiras mais críticas de um curso.

a) Construa a classe `CadeirasCriticas`, definindo os atributos *curso* (uma *String*) que representa o nome do curso, e *disciplinas* (um array com 3 *Strings*) que representa as 3 disciplinas críticas desse curso. Defina os atributos, um construtor que receba como parâmetro o nome do curso, construa os getters os setters e o método `toString`.

b) Construa uma classe para testar os **métodos** da classe `CadeirasCriticas`. Deverá declarar pelo menos dois objectos do tipo cadeiras críticas, `cc1` e `cc2`, instanciar os objectos, atribuir valores a todos os atributos do objectos, modificar esses valores e mostrar o conteúdo de cada objecto.

c) Teste novamente o método `setDisciplinas`, declarando no main um array de *Strings* com dimensão 3 (de nome `novasCriticas`), inicializando-o com valores à sua escolha, e usando-o para modificar o valor da lista de cadeiras críticas do objecto `cc1`. No final, invoque o método `toString` no objeto `cc1` para ver o conteúdo do objecto.

**De seguida** atribua novos valores ao array `novasCriticas`, no main da classe de teste. Invoque o método `toString` e analise o que aconteceu? O seu método `setDisciplinas` estava mesmo correcto? Se não estava, corrija-o.

## □ Variáveis de Classe e Composição de Classes

7. Construa a classe `Aluno` esquematizada abaixo:

```
public class Aluno{
    private static int ultimo = 0;
    private int numero;
    private String nome;
    private Disciplina [] lista; // array de objectos do tipo Disciplina (ver exerc. 4)
    ...
}
```

a) Para a classe `Aluno` defina um construtor que tenha como parâmetro um valor para o nome do `Aluno`. O construtor deverá fazer com que o número de aluno seja atribuído automaticamente de forma sequencial, sempre que um novo aluno é criado. A variável de classe **ultimo** contém o número do último aluno que foi instanciado. Considere ainda que todos os alunos têm 3 disciplinas.

b) Defina para a classe `Aluno` os métodos de consulta (getters) e de modificação (setters)

c) Defina para a classe `Aluno` um método que permita atribuir uma `Disciplina`, a uma dada posição do atributo `lista`. A assinatura do método deverá ser:

`adicionarDisciplina (Disciplina d, int p).`

d) Construa o método “`toString`” para a classe `Aluno`.

e) Para a classe `Aluno`, construa um método que verifique se uma dada disciplina faz parte da lista de disciplinas do `Aluno`.

8. Construa uma classe de teste para a classe Aluno, tal que:

i) Declare e instancie dois objectos do tipo Aluno:

O aluno a1 com o nome “Manuel Silva”, tem as disciplinas (1000, “POO”, “Joseph Katame”), (2000, “IA”, “Luís Alexandre”), (3000, “Multimédia”, Manuela Pereira).

O aluno a2 com o nome “Maria Espinho”, tem as disciplinas (1000, “POO”, Joseph Katame”), (4000, “Probabilidades e Estatística”, “João Duque”).

ii) Teste os métodos os restantes métodos da classe.

9. Construa uma classe que represente os Empregados de uma empresa. Um Empregado tem um número de segurança social, um nome e um salário. Defina os atributos, dois construtores à sua escolha, os métodos de consulta (getters) e os de modificação (setters), e o método toString. Construa ainda um método que permita subir o salário do empregado de uma dada percentagem dada como parâmetro.

Construa uma classe de teste que permita verificar todos os métodos da classe Empregado

10. A Federação Moçambicana de Futebol encomendou um programa para gerir o número de golos marcados pelos jogadores de cada equipa da 1ª divisão. Para isso pretende-se desenvolver a classe Jogador. Deve definir os modificadores de acesso para que apenas os métodos da classe acedam aos atributos da classe e que qualquer classe que tenha acesso ao package da classe possa aceder aos seus métodos.

- Cada instância da classe Jogador terá como atributos: um número do Jogador que deverá ser atribuído automaticamente e de forma sequencial, sempre que um novo Jogador é criado: o nome do jogador (cadeira de caracteres); o salário do Jogador (um valor do tipo double); e um array com 34 valores inteiros onde serão armazenados o número de golos que o Jogador marcou. (Na posição 0, número de golos marcados na 1ª jornada, na posição 1, o número de golos marcados na 2ª jornada e assim por diante).

A classe deverá ter dois construtores, um com o parâmetro nome e outro com os parâmetros nome e salário.

Cada instância da classe Jogador deverá poder responder a mensagens para:

- consultar o nome do jogador;
- modificar o nome do jogador;
- consultar o salário do jogador;
- modificar o salário do jogador;
- consultar o array de golos do jogador;
- modificar o array de golos do jogador;
- aumentar o salário de uma dada percentagem (double, entre 0 e 100);
- modificar o número de golos na n-ésima jornada;

- consultar o número de golos na n-ésima jornada;
- a) Construa a classe Jogador de forma a responder às especificações anteriores.
- b) Construa para a classe Jogador o método toString.
- c) Para a classe Jogador construa um método que devolva o total de golos marcados por um jogador.
- d) Construa uma classe de teste que permita verificar os dois construtores, e os métodos de instância definidos.