

## ESQUEMA AULA PRÁTICA #3

### ❑ **Objetos e classes de teste**

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

### **1 - Pretende-se implementar a classe Contador que foi estudada nas aulas teóricas.**

**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 uma valor dado pelo utilizador;
- incrementar o Contador de uma unidade;
- incrementar o Contador de uma valor dado pelo utilizador;
- decrementar o Contador de uma unidade;
- decrementar o Contador de uma valor dado pelo utilizador;
- consultar o valor do Contador;
- implemente ainda um método para mostrar um Contador sob a gorma de texto, (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.

2 – 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 espectáculo tem ainda um **custo** que corresponde ao preço do bilhete para assistir ao espectáculo.

- 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.

*Para explorar:*

3- O programa (1) que se segue, está construído numa classe, Tempo. Contém uma função main, que constitui o código onde terá início a execução, e nessa função são invocadas as outras funções que estão definidas na mesma classe, getHoras e periodoDia. A classe contém ainda a declaração de um conjunto de variáveis (saudacoes, nome, horas, minutos) que são globais às funções definidas na classe.

A classe LocalDateTime existe pré-definida na linguagem Java, e é usada neste programa. Apesar de o programa conter alguns elementos que apenas estudaremos mais para a frente durante este curso, tente explorá-lo.

1) Adaptado de “A small cup of Java” de Pedro Guerreiro

```
import java.time.LocalDateTime;

public class Tempo {

    private static String[] saudacoes =
        {"Bom dia", "Boa tarde", "Boa noite"};

    private static String nome = "Escreva aqui o seu nome";
    private static int horas;
    private static int minutos;

    public static void getHoras( ){
        LocalDateTime dataComputador = LocalDateTime.now();
        horas = dataComputador.getHour();
        minutos = dataComputador.getMinute();
    }
    private static int periodoDia(int h){
        return (h+20) /8 % 3;
    }
    public static void main(String args[]){
        getHoras ( ) ;
        System.out.println(saudacoes[periodoDia(horas)] + ", " + nome);
        System.out.println("Passam " + minutos + " minutos das " + horas + " horas." ) ;
    }
}
```

- Depois de analisar o programa, tente modificá-lo para que:

- i) Apresente as horas de um modo gramaticalmente correto (1 hora, 1 minuto! Passam, faltam! Horas exactas?)
- ii) Refaça o exercício anterior usando o operador ternário.
- iii) Apresente também a data.
- iv) Refaça o exercício anterior mostrando o mês por extenso.
- v) Indique o número de dias que faltam até ao fim-de-semana...
- vi) Apresente a listagem das datas das 13 próximas Sextas-feiras 13.