FACULDADE SENAI PARAÍBA

Curso Superior de Automação Industrial

# Programação Orientada a Objetos

**Prof. Daniel Brandão**

SENAI

# Exercício 1

**Crie uma classe Principal e uma classe de teste que contenham métodos para as seguintes rotinas:**

A. Imprima todos os números pares entre 1 e 100;
B. Imprima todos os múltiplos de 3 entre 1 e 100;

# Exercício 2

- Crie uma classe chamada **Pessoa**. A classe terá os atributos **“nome”, “sexo” e “idade”.** Deve haver o método “maiorDeIdade()”, que retorna **true** se a pessoa for maior de idade, ou **false** caso contrário. Considere a maioridade aos 18 anos. Na sequência, escreva uma classe que instancie um objeto tipo **Pessoa** e verifique se o mesmo é ou não maior de idade, imprimindo o resultado via console.

# Exercício 3

- Uma locadora possui dois tipos de **veículos**: **utilitário** e de **passeio**. O veículo, independente do tipo, deve ser identificada pelo modelo, marca, ano, preço de locação e quantidade de dias de locação. Para calcular o preço da locação e construir as entidades apresentadas, utilize como regra as definições a seguir:
  - Veículos utilitários possuem um valor fixo acrescentado de **R$ 40,00**;
  - Veículos de passeio possuem um valor fixo com desconto de **R$ 20,00**;
  - Teste a classe, imprimindo um teste com a locação sendo calculada por dias contratados, do utilitário e o de passeio.

# Exercício 4

Crie uma classe chamada **Fatura** para que uma loja de suprimentos de informática possa utilizá-la para representar uma fatura de um item vendido na loja.

- Uma Fatura deve incluir quatro partes das informações como variáveis de instância – o **número** (tipo String), a **descrição** (tipo String), a **quantidade comprada** de um item(tipo int) e o **preço por item** (tipo double).
- Além disso, forneça um método chamado **getValorFatura** que calcula o valor da fatura (isso é, multiplica a quantidade pelo preço por item) e depois retorna o valor como um double.
- Se o valor não for positivo, ele deve ser configurado como 0. Se o preço por item não for positivo ele deve ser configurado como 0.0

# Exercício 5

- Implemente uma classe chamada **Aluno**. Essa classe deve possuir 4 atributos: **nome, cpf, matricula** e **e-mail**. Faça um teste criando objetos da classe **Aluno**. Altere e exiba no Console os valores armazenados nos atributos desse objeto. Crie uma nova classe chamada **TestaAluno** trazendo como retorno os atributos de 2 alunos diferentes.

# Exercício 6

- Implemente uma classe para definir os objetos que representarão os **clientes** de um **banco**. Essa classe deve possuir 5 atributos: **nome, rg, cpf, telefone e número da conta**. Faça um teste criando objetos da classe **Cliente**. Altere e exiba no Console os valores armazenados nos atributos desses objetos. Crie uma nova classe chamada **TestaCliente** trazendo como retorno os atributos de cada cliente.

# Exercício 7

- Implemente uma classe para definir os objetos que representarão os **funcionários** de uma empresa. Eles devem possuir 5 atributos: **nome, matricula, CPF, salário e vale refeição**. Essa classe deverá ter um método que **calcule o salario bruto** do funcionário, que será a soma do **salário** com o **vale refeição**. Faça um teste criando objetos da classe **Funcionário**. Altere e exiba no Console os valores armazenados nos atributos desses objetos. Crie uma nova classe chamada **TestaFuncionario** trazendo como retorno os atributos com seus respectivos valores.

Os Exercícios devem ser entregues via e-mail,
até Segunda-feira (18/03)

professordanielbrandao@gmail.com

Programas:

- **Eclipse** (acima da versão Mars – www.eclipse.org
- **Java JDK** (www.oracle.com)

# Dúvidas?