

## ATIVIDADE PRÁTICA – JAVA

JAVA-11 – POO: Fundamentos

### Instruções gerais:

1. Utilize o Eclipse ou o STS para desenvolver os algoritmos.
2. Ao concluir os exercícios, envie todos os códigos criados no Eclipse ou no STS para o Repositório criado na sua conta pessoal do Github, em uma pasta identificada com o tema da sessão
3. Caso seja solicitado, adicione o **link do Github** ou os **prints** dos arquivos .JAVA, indicados pelo instrutor na Plataforma Canvas.

**Mantenha as entregas das Atividades em dia na Plataforma Canvas**

## EXERCÍCIOS

### Boas práticas:

1. Resolva 2 exercícios da lista
2. Leia o enunciado do exercício com atenção
3. Observe as indicações de Entrada e Saída esperadas em cada exercício
4. Observe com atenção os desenhos e diagramas inseridos nos exercícios para facilitar a compreensão
5. Utilize o Cookbook, os Vídeos da Plataforma e os Códigos guia como referências para a resolução dos exercícios
6. Caso ainda fique alguma dúvida, consulte os instrutores da sua turma pelo Discord

## Atividade 01

Utilizando os conceitos de Classe, Objeto e Métodos, da Programação Orientada a Objetos, crie a **Classe Cliente** com os seus respectivos Métodos e Atributos. Na sequência, crie uma Classe chamada **TestaCliente**, instancie **dois objetos da Classe Cliente** e apresente as informações destes 2 Objetos no console.

### Boas práticas:

- 1) Crie a Classe Cliente e **defina pelo menos 5 Atributos** relevantes ao Objeto Cliente, a sua escolha;
- 2) Lembre-se de escolher Atributos genéricos, que descrevam as características gerais de qualquer Cliente;
- 3) Crie o **Método Construtor com parâmetros**, contendo todos os Atributos definidos na Classe Cliente nos argumentos do Método;
- 4) Crie os **Métodos Get e Set para todos os Atributos da Classe Cliente**;
- 5) Crie o **Método visualizar()**, com a função de listar todos os Atributos de um Objeto da Classe Cliente;
- 6) Crie a Classe **TestaCliente** no mesmo pacote da Classe Cliente, contendo o Método **main()** e **instancie 2 Objetos da Classe Cliente**;
- 7) Utilize o Método **visualizar()** para exibir os dados dos 2 Objetos Instanciados.

## Atividade 02

Utilizando os conceitos de Classe, Objeto e Métodos, da Programação Orientada a Objetos, crie a **Classe Funcionario** com os seus respectivos Métodos e Atributos. Na sequência, crie uma Classe chamada **TestaFuncionario**, instancie **dois objetos da Classe Funcionario** e apresente as informações destes 2 Objetos no console.

### Boas práticas:

- 1) Crie a Classe Funcionario e **defina pelo menos 5 Atributos** relevantes ao Objeto Funcionario, a sua escolha;
- 2) Lembre-se de escolher Atributos genéricos, que descrevam as características gerais de qualquer Funcionario;
- 3) Crie o **Método Construtor com parâmetros**, contendo todos os Atributos definidos na Classe Funcionario nos argumentos do Método;
- 4) Crie os **Métodos Get e Set para todos os Atributos da Classe Funcionario**;
- 5) Crie o **Método visualizar()**, com a função de listar todos os Atributos de um Objeto da Classe Funcionario;
- 6) Crie a Classe **TestaFuncionario** no mesmo pacote da Classe Funcionario, contendo o Método main() e **instancie 2 Objetos da Classe Funcionario**;
- 7) Utilize o Método visualizar() para exibir os dados dos 2 Objetos Instanciados.

## Atividade 03

Utilizando os conceitos de Classe, Objeto e Métodos, da Programação Orientada a Objetos, crie a **Classe Produto** com os seus respectivos Métodos e Atributos, que descreva os produtos de uma **Loja de Games**. Na sequência, crie uma Classe chamada **TestaGame**, instancie **dois objetos da Classe Produto** e apresente as informações destes 2 Objetos no console.

### Boas práticas:

- 1) Crie a Classe Produto e **defina pelo menos 5 Atributos** relevantes aos Produtos de uma Loja de Games, a sua escolha;
- 2) Lembre-se de escolher Atributos genéricos, que descrevam as características gerais de qualquer produto da Loja de Games;
- 3) Crie o **Método Construtor com parâmetros**, contendo todos os Atributos definidos na Classe Produto nos argumentos do Método;
- 4) Crie os **Métodos Get e Set para todos os Atributos da ClasseProduto**;
- 5) Crie o **Método visualizar()**, com a função de listar todos os Atributos de um Objeto da Classe Produto;
- 6) Crie a Classe **TestaGame** no mesmo pacote da Classe Game, contendo o Método **main()** e **instancie 2 Objetos da Classe Produto**;
- 7) Utilize o **Método visualizar()** para exibir os dados dos 2 Objetos Instanciados.

## Atividade 04

Utilizando os conceitos de Classe, Objeto e Métodos, da Programação Orientada a Objetos, crie a **Classe Farmacia** com os seus respectivos Métodos e Atributos, que descreva os produtos de uma **Farmácia**. Na sequência, crie uma Classe chamada **TestaFarmacia**, instancie **dois objetos da Classe Farmacia** e apresente as informações destes 2 Objetos no console.

### Boas práticas:

- 1) Crie a Classe Farmacia e **defina pelo menos 5 Atributos** relevantes aos Produtos de uma Farmacia, a sua escolha;
- 2) Lembre-se de escolher Atributos genéricos, que descrevam as características gerais de qualquer produto da Farmácia;
- 3) Crie o **Método Construtor com parâmetros**, contendo todos os Atributos definidos na Classe Farmacia nos argumentos do Método;
- 4) Crie os **Métodos Get e Set para todos os Atributos da Classe Farmacia**;
- 5) Crie o **Método visualizar()**, com a função de listar todos os Atributos de um Objeto da Classe Farmacia;
- 6) Crie a Classe **TestaFarmacia** no mesmo pacote da Classe Farmacia, contendo o Método main() e **instancie 2 Objetos da Classe Farmacia**;
- 7) Utilize o Método visualizar() para exibir os dados dos 2 Objetos Instanciados.

## Atividade 05

Utilizando os conceitos de Classe, Objeto e Métodos, da Programação Orientada a Objetos, crie a **Classe Ingresso** com os seus respectivos Métodos e Atributos, que descreva o ingresso de um evento qualquer. Na sequência, crie uma Classe chamada **TestaIngresso**, instancie **dois objetos da Classe Ingresso** e apresente as informações destes 2 Objetos no console.

### Boas práticas:

- 1) Crie a Classe Ingresso e **defina pelo menos 5 Atributos** relevantes ao Ingresso de um evento, a sua escolha;
- 2) Lembre-se de escolher Atributos genéricos, que descrevam as características gerais de qualquer evento;
- 3) Crie o **Método Construtor com parâmetros**, contendo todos os Atributos definidos na Classe Ingresso nos argumentos do Método;
- 4) Crie os **Métodos Get e Set para todos os Atributos da Classe Ingresso**;
- 5) Crie o **Método visualizar()**, com a função de listar todos os Atributos de um Objeto da Classe Ingresso;
- 6) Crie a Classe **TestaIngresso** no mesmo pacote da Classe Ingresso, contendo o Método main() e **instancie 2 Objetos da Classe Ingresso**;
- 7) Utilize o Método visualizar() para exibir os dados dos 2 Objetos Instanciados.

## Atividade 06

Utilizando os conceitos de Classe, Objeto e Métodos, da Programação Orientada a Objetos, crie a **Classe Curso** com os seus respectivos Métodos e Atributos, que descreva um Curso qualquer, de uma Plataforma EAD. Na sequência, crie uma Classe chamada **TestaCurso**, instancie **dois objetos da Classe Curso** e apresente as informações destes 2 Objetos no console.

### Boas práticas:

- 1) Crie a Classe Curso e **defina pelo menos 5 Atributos** relevantes ao Curso EAD, a sua escolha;
- 2) Lembre-se de escolher Atributos genéricos, que descrevam as características gerais de qualquer Curso EAD;
- 3) Crie o **Método Construtor com parâmetros**, contendo todos os Atributos definidos na Classe Curso nos argumentos do Método;
- 4) Crie os **Métodos Get e Set para todos os Atributos da Classe Curso**;
- 5) Crie o **Método visualizar()**, com a função de listar todos os Atributos de um Objeto da Classe Curso;
- 6) Crie a Classe **TestaCurso** no mesmo pacote da Classe Curso, contendo o Método **main()** e **instancie 2 Objetos da Classe Curso**;
- 7) Utilize o Método **visualizar()** para exibir os dados dos 2 Objetos Instanciados.