

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:

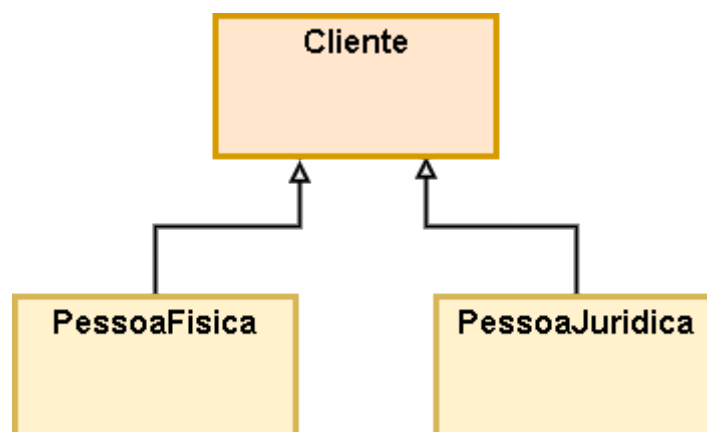
1. Dê continuidade aos mesmos 2 exercícios da lista anterior
2. Leia o enunciado do exercício com atenção
3. Observe as indicações de Entrada e Saída esperadas em cada exercício
4. Observe com atenção os desenhos e diagramas inseridos nos exercícios para facilitar a compreensão
5. Utilize o Cookbook, 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



ATENÇÃO: *Esta Atividade é uma continuação da Atividade anterior (JAVA-11 – POO: Fundamentos). Selecione os mesmos exercícios da lista anterior e implemente o que se pede.*

Atividade 01

Utilizando os conceitos de Herança e Polimorfismo, da Programação Orientada a Objetos, crie as Classes **PessoaFisica** e **PessoaJuridica**, como Heranças da **Classe Cliente** (criada na lista de exercícios anterior), com os seus respectivos Métodos e Atributos. Na sequência, utilize a Classe **TestaCliente** para instanciar **dois objetos da Classe PessoaFisica** e **dois objetos da Classe PessoaJuridica**, e apresente as informações destes 4 Objetos na tela.



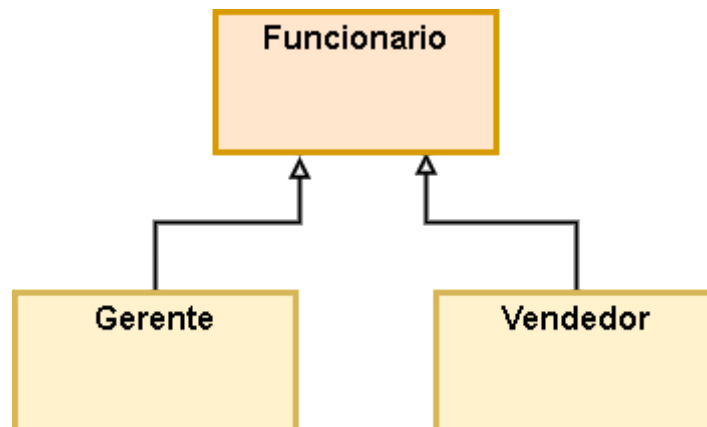
Boas práticas:

- 1) Crie a Classe **PessoaFisica** como uma herança da Classe **Cliente** e **defina pelo menos 1 Atributo** relevante, a pessoa física, a sua escolha;
- 2) Lembre-se de escolher Atributos que descrevam características específicas e exclusivas de uma pessoa física;
- 3) Crie o **Método Construtor com parâmetros** na Classe **PessoaFisica**;
- 4) Crie os **Métodos Get e Set para todos os Atributos** da Classe **PessoaFisica**;
- 5) Crie o **Método visualizar()** na Classe **PessoaFisica**, que consiga exibir todos os dados do Objeto;
- 6) **Instancie 2 Objetos da Classe PessoaFisica na Classe TestaCliente**;
- 7) Utilize o Método **visualizar()** para exibir os dados dos 2 Objetos Instanciados.
- 8) Crie a Classe **PessoaJuridica** como uma herança da Classe **Cliente** e **defina pelo menos 1 Atributo** relevante, a pessoa jurídica, a sua escolha;
- 9) Lembre-se de escolher Atributos que descrevam características específicas e exclusivas de uma pessoa jurídica;
- 10) Crie o **Método Construtor com parâmetros** na Classe **PessoaJuridica**;
- 11) Crie os **Métodos Get e Set para todos os Atributos** da Classe **PessoaJuridica**;

- 12) Crie o **Método visualizar()** na Classe PessoaJuridica, que consiga exibir todos os dados do Objeto;
- 13) **Instancie 2 Objetos da Classe PessoaJuridica na Classe TestaCliente;**
- 14) Utilize o Método visualizar() para exibir os dados dos 2 Objetos Instanciados.

Atividade 02

Utilizando os conceitos de Herança e Polimorfismo, da Programação Orientada a Objetos, crie as Classes **Gerente** e **Vendedor** como heranças da **Classe Funcionario** (criada na lista de exercícios anterior), com os seus respectivos Métodos e Atributos. Na sequência, utilize a Classe **TestaFuncionario** para instanciar **dois objetos da Classe Gerente** e **dois objetos da Classe Vendedor**, e apresente as informações destes 4 Objetos na tela.



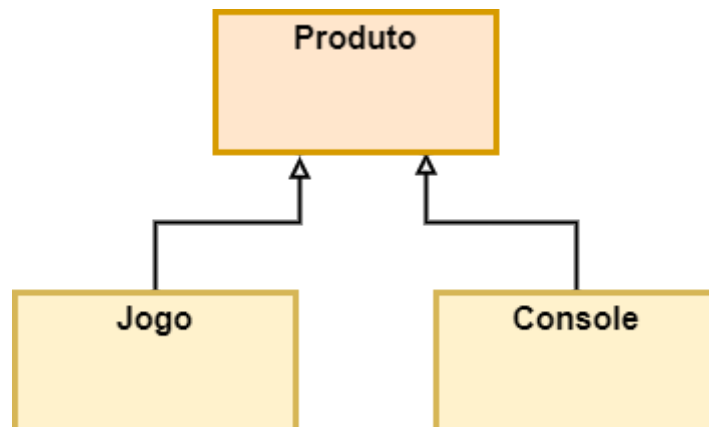
Boas práticas:

- 1) Crie a Classe Gerente como uma herança da Classe Funcionario e **defina pelo menos 1 Atributo** relevante ao Gerente, a sua escolha;
- 2) Lembre-se de escolher Atributos que descrevam características específicas e exclusivas de um Gerente;
- 3) Crie o **Método Construtor com parâmetros** na Classe Gerente;
- 4) Crie os **Métodos Get e Set para todos os Atributos** da Classe Gerente;
- 5) Crie o **Método visualizar()** na Classe Gerente, que consiga exibir todos os dados do Objeto;
- 6) **Instancie 2 Objetos da Classe Gerente na Classe TestaFuncionario;**
- 7) Utilize o Método visualizar() para exibir os dados dos 2 Objetos Instanciados.
- 8) Crie a Classe Vendedor como uma herança da Classe Funcionario e **defina pelo menos 1 Atributo** relevante ao vendedor, a sua escolha;
- 9) Lembre-se de escolher Atributos que descrevam características específicas e exclusivas de um vendedor;
- 10) Crie o **Método Construtor com parâmetros** na Classe Vendedor;
- 11) Crie os **Métodos Get e Set para todos os Atributos** da Classe Vendedor;

- 12) Crie o **Método visualizar()** na Classe Vendedor, que consiga exibir todos os dados do Objeto;
- 13) **Instancie 2 Objetos da Classe Vendedor na Classe TestaFuncionario;**
- 14) Utilize o Método visualizar() para exibir os dados dos 2 Objetos Instanciados.

Atividade 03

Utilizando os conceitos de Herança e Polimorfismo, da Programação Orientada a Objetos, crie as Classes **Jogo** e **Console** (PS5, Xbox, etc.) como heranças da **Classe Produto** (criada na lista de exercícios anterior), com os seus respectivos Métodos e Atributos. Na sequência, utilize a Classe **TestaGame** para instanciar **dois objetos da Classe Jogo** e **dois objetos da Classe Console**, e apresente as informações destes 4 Objetos na tela.



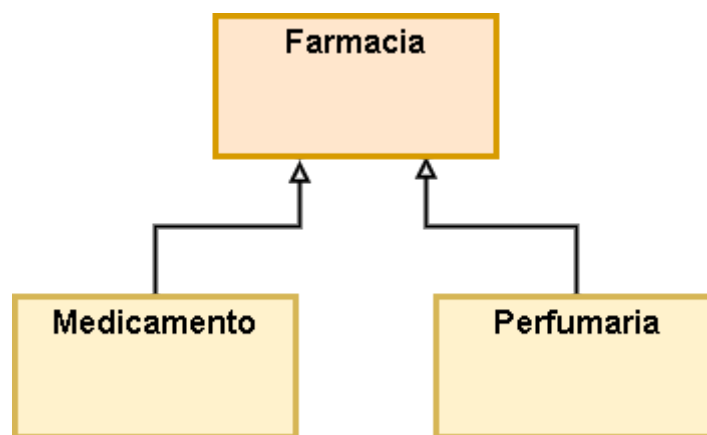
Boas práticas:

- 1) Crie a Classe Jogo como uma herança da Classe Game e **defina pelo menos 1 Atributo** relevante a um jogo qualquer, a sua escolha;
- 2) Lembre-se de escolher Atributos que descrevem características específicas e exclusivas de um Jogo;
- 3) Crie o **Método Construtor com parâmetros** na Classe Jogo;
- 4) Crie os **Métodos Get e Set para todos os Atributos** da Classe Jogo;
- 5) Crie o **Método visualizar()** na Classe Jogo, que consiga exibir todos os dados do Objeto;
- 6) **Instancie 2 Objetos da Classe Jogo na Classe TestaGame;**
- 7) Utilize o Método visualizar() para exibir os dados dos 2 Objetos Instanciados.
- 8) Crie a Classe Console como uma herança da Classe Game e **defina pelo menos 1 Atributo** relevante a um console qualquer, a sua escolha;
- 9) Lembre-se de escolher Atributos que descrevem características específicas e exclusivas de um console;
- 10) Crie o **Método Construtor com parâmetros** na Classe Console;
- 11) Crie os **Métodos Get e Set para todos os Atributos** da Classe Console;
- 12) Crie o **Método visualizar()** na Classe Console, que consiga exibir todos os dados do Objeto;

- 13) **Instancie 2 Objetos da Classe Vendedor na Classe TestaGame;**
- 14) Utilize o Método visualizar() para exibir os dados dos 2 Objetos Instanciados.

Atividade 04

Utilizando os conceitos de Herança e Polimorfismo, da Programação Orientada a Objetos, crie as Classes **Medicamento** e **Perfumaria** como heranças da **Classe Farmacia** (criada na lista de exercícios anterior), com os seus respectivos Métodos e Atributos. Na sequência, utilize a Classe **TestaFarmacia** para instanciar **dois objetos da Classe Medicamento** e **dois objetos da Classe Perfumaria**, e apresente as informações destes 4 Objetos na tela.



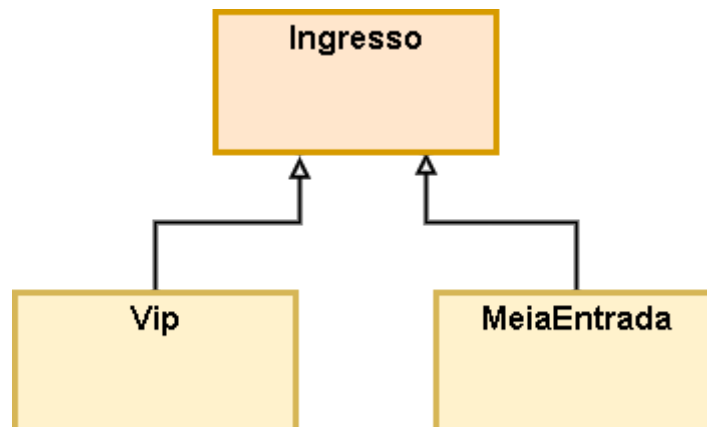
Boas práticas:

- 1) Crie a Classe Medicamento como uma herança da Classe Farmacia e **defina pelo menos 1 Atributo** relevante aos produtos da categoria Medicamento, a sua escolha;
- 2) Lembre-se de escolher Atributos que descrevam características específicas e exclusivas de um produto da categoria medicamento;
- 3) Crie o **Método Construtor com parâmetros** na Classe Medicamento;
- 4) Crie os **Métodos Get e Set para todos os Atributos** da Classe Medicamento;
- 5) Crie o **Método visualizar()** na Classe Medicamento, que consiga exibir todos os dados do Objeto;
- 6) **Instancie 2 Objetos da Classe Medicamento na Classe TestaFarmacia;**
- 7) Utilize o Método visualizar() para exibir os dados dos 2 Objetos Instanciados.
- 8) Crie a Classe Perfumaria como uma herança da Classe Farmacia e **defina pelo menos 1 Atributo** relevante aos produtos da categoria perfumaria, a sua escolha;
- 9) Lembre-se de escolher Atributos que descrevam características específicas e exclusivas de um produto da categoria perfumaria;
- 10) Crie o **Método Construtor com parâmetros** na Classe Perfumaria;
- 11) Crie os **Métodos Get e Set para todos os Atributos** da Classe Perfumaria;

- 12) Crie o **Método visualizar()** na Classe Perfumaria, que consiga exibir todos os dados do Objeto;
- 13) **Instancie 2 Objetos da Classe Perfumaria na Classe TestaFarmacia;**
- 14) Utilize o Método visualizar() para exibir os dados dos 2 Objetos Instanciados.

Atividade 05

Utilizando os conceitos de Herança e Polimorfismo, da Programação Orientada a Objetos, crie as Classes **Vip** e **MeiaEntrada** como heranças da **Classe Ingresso** (criada na lista de exercícios anterior), com os seus respectivos Métodos e Atributos. Na sequência, utilize a Classe **TestaIngresso** para instanciar **dois objetos da Classe Vip** e **dois objetos da Classe MeiaEntrada**, e apresente as informações destes 4 Objetos na tela.



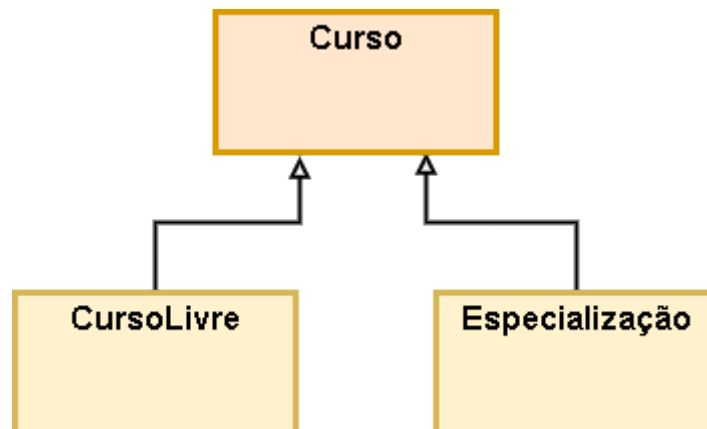
Boas práticas:

- 1) Crie a Classe Vip como uma herança da Classe Ingresso e **defina pelo menos 1 Atributo** relevante ao ingresso do tipo VIP, a sua escolha;
- 2) Lembre-se de escolher Atributos que descrevam características específicas e exclusivas de um ingresso VIP;
- 3) Crie o **Método Construtor com parâmetros** na Classe Vip;
- 4) Crie os **Métodos Get e Set para todos os Atributos** da Classe Vip;
- 5) Crie o **Método visualizar()** na Classe Vip, que consiga exibir todos os dados do Objeto;
- 6) **Instancie 2 Objetos da Classe Vip na Classe TestaIngresso;**
- 7) Utilize o Método visualizar() para exibir os dados dos 2 Objetos Instanciados.
- 8) Crie a Classe MeiaEntrada como uma herança da Classe Ingresso e **defina pelo menos 1 Atributo** relevante ao ingresso do tipo meia entrada, a sua escolha;
- 9) Lembre-se de escolher Atributos que descrevam características específicas e exclusivas de um ingresso do tipo meia entrada;
- 10) Crie o **Método Construtor com parâmetros** na Classe MeiaEntrada;
- 11) Crie os **Métodos Get e Set para todos os Atributos** da Classe MeiaEntrada;
- 12) Crie o **Método visualizar()** na Classe MeiaEntrada, que consiga exibir todos os dados do Objeto;

- 13) **Instancie 2 Objetos da Classe MeiaEntrada na Classe TestaIngresso;**
- 14) Utilize o Método visualizar() para exibir os dados dos 2 Objetos Instanciados.

Atividade 06

Utilizando os conceitos de Herança e Polimorfismo, da Programação Orientada a Objetos, crie as Classes **CursoLivre** (Curso rápido, com patrocinador) e **Especialização** (Curso longo, com certificado reconhecido pelo MEC) como heranças da **Classe Curso** (criada na lista de exercícios anterior), com os seus respectivos Métodos e Atributos. Na sequência, utilize a Classe **TestaCurso** para instanciar **dois objetos da Classe CursoLivre** e **dois objetos da Classe Especialização**, e apresente as informações destes 4 Objetos na tela.



Boas práticas:

- 1) Crie a Classe **CursoLivre** como uma herança da Classe **Curso** e **defina pelo menos 1 Atributo** relevante ao curso do tipo **CursoLivre**, a sua escolha;
- 2) Lembre-se de escolher Atributos que descrevam características específicas e exclusivas de um curso do tipo livre;
- 3) Crie o **Método Construtor com parâmetros** na Classe **CursoLivre**;
- 4) Crie os **Métodos Get e Set para todos os Atributos** da Classe **CursoLivre**;
- 5) Crie o **Método visualizar()** na Classe **CursoLivre**, que consiga exibir todos os dados do Objeto;
- 6) **Instancie 2 Objetos da Classe CursoLivre na Classe TestaCurso**;
- 7) Utilize o Método **visualizar()** para exibir os dados dos 2 Objetos Instanciados.
- 8) Crie a Classe **Especialização** como uma herança da Classe **Curso** e **defina pelo menos 1 Atributo** relevante ao curso do tipo **Especialização**, a sua escolha;
- 9) Lembre-se de escolher Atributos que descrevam características específicas e exclusivas de um curso de **Especialização**;
- 10) Crie o **Método Construtor com parâmetros** na Classe **Especialização**;
- 11) Crie os **Métodos Get e Set para todos os Atributos** da Classe **Especialização**;

- 12) Crie o **Método visualizar()** na Classe Especialização, que consiga exibir todos os dados do Objeto;
- 13) **Instancie 2 Objetos da Classe Especialização na Classe TestaCurso;**
- 14) Utilize o Método visualizar() para exibir os dados dos 2 Objetos Instanciados.