

Linguagem de Programação I

https://www.w3schools.com/java/java_classes.asp

Ementa

~~Introdução à linguagem orientada a Objetos.~~

~~Criando objetos. Instância de objeto.~~

~~Construtores.~~

Destruítores.

~~Encapsulamento.~~

~~Herança e Polimorfismo.~~

Aula 1

- ☐ Crie uma classe **Cachorro** que possua um atributo *raca*
- ☐ Crie um método *emitirSom* na classe **Cachorro**, do tipo static e que não retorne valor
- ☐ Crie um método *correr*, sem retorno
- ☐ Crie uma classe **Testes**, com um método *main* e dentro dele, crie uma instância da classe **Cachorro** e chame seus métodos

Aula 2

- ☐ Crie um projeto Java
- ☐ Crie um pacote nesse projeto chamado *br.com.objetos*
- ☐ Crie uma classe **Aluno** com os atributos: *nome*, *genero* e *idade*
- ☐ Crie uma classe de testes, contendo o método *main*, crie dois objetos da classe **Aluno** e exiba no console o nome de cada um deles

Aula 3

- ☐ Crie uma classe **Mensagem** contendo os atributos *data* (literal), *autor* (literal) e *texto* (literal)

- ☐ Crie um método estático *configurar*, que exiba uma mensagem “Configuração realizada” no console
- ☐ Crie um método que receba três parâmetros (*data*, *autor* e *texto*) que devem ser copiados para os atributos do objeto da classe.
- ☐ Crie um método *enviar* que recupere os valores *data*, *autor* e *texto* do objeto e mostre no console uma mensagem que concatene os três valores na forma: “Olá <autor>, sua mensagem com o texto <texto> foi enviada em <data>”

Aula 4

- ☐ Modifique a classe anterior, fazendo com que o método que receba os três parâmetros e os transfere para os atributos de classe se torne o construtor da classe.
- ☐ Crie o construtor padrão da classe *Mensagem*

Aula 5

- ☐ Altere os atributos da classe **Mensagem** usando o modificador apropriado para fazê-los privados.
- ☐ Crie os métodos de encapsulamento para a classe **Mensagem**
- ☐ Crie uma classe de **Testes** que contenha o método *main* e nela, crie instâncias da classe **Mensagem** usando seu construtor que recebe parâmetros. Depois exiba o valor desses atributos no console usando os métodos de encapsulamento.

Aula 6

- ☐ Crie uma classe **Animal** que obedeça à seguinte descrição: possua os atributos *nome* (String), *comprimento* (float), *número de patas* (int), *cor* (String), *ambiente* (String) e *velocidade média* (float).
- ☐ Crie um método construtor que receba por parâmetro os valores iniciais de cada um dos atributos de classe.
- ☐ Encapsule os atributos.
- ☐ Crie um método *verDados*, sem parâmetros e do tipo *void*, que, quando chamado, imprime na tela os dados do animal.

- ☐ Crie uma classe **Peixe** que herde da classe **Animal** e obedeça à seguinte descrição: possua um atributo *caracteristica* (String)
- ☐ Crie um método construtor que receba por parâmetro os valores iniciais de cada um dos atributos (incluindo os atributos da classe **Animal**).
- ☐ Encapsule os atributos da classe **Peixe**.
- ☐ Crie um método *dadosPeixe* sem parâmetros e do tipo *void*, que, quando chamado, imprime na tela os dados do peixe (incluindo os dados do **Animal**).
- ☐ Crie uma classe de testes para criar instâncias de **Peixe**, dando valores aos seus atributos e chamando o método *dadosPeixe*.

Aula 7

- ☐ Crie uma hierarquia de classes para representar os diferentes tipos de **funcionários** de um escritório que tem os seguintes cargos: **gerente**, **assistente**, **vendedor**. Escreva uma classe base chamada **Funcionario** que contenha os atributos: *nome* (String), *matricula* (String) e *salarioBase* (double). Use encapsulamento e forneça um construtor que receba os valores correspondentes a serem armazenados nos respectivos atributos.
- ☐ A classe **Funcionario** deverá ser estendida pelas classes representativas dos tipos de funcionários: **Gerente**, **Assistente** e **Vendedor**.
- ☐ Crie uma classe **Teste** com um método *main* que cria um objeto de cada tipo de funcionário. Calcule então a soma dos salários dos três funcionários e imprima no console o valor total.

Aula 8

- ☐ Crie uma classe chamada **Ingresso** que possui um *valor* em reais e um método *imprimeValor*.
- ☐ Crie uma classe **VIP**, que herda **Ingresso** e possui um valor adicional.
- ☐ Crie um método que retorne o valor do ingresso **VIP** (com o adicional incluído).
- ☐ Crie uma classe **Normal**, que herda **Ingresso** e possui um método que imprime: "Ingresso Normal".
- ☐ Crie uma classe **CamaroteInferior** (que possui a localização do ingresso e métodos para acessar e imprimir esta localização) e uma classe **CamaroteSuperior**, que é mais cara (possui valor adicional). Esta última possui

um método para retornar o valor do ingresso. Ambas as classes herdam a classe **VIP**.

- ☐ Crie a classe **Imovel**, que possui um endereço e um preço.
- ☐ Crie uma classe **Novo**, que herda **Imovel** e possui um adicional no preço. Crie métodos de acesso e impressão deste valor adicional.
- ☐ Crie uma classe **Velho**, que herda **Imovel** e possui um desconto no preço. Crie métodos de acesso e impressão para este desconto.

Aula 9

- ☐ Dadas as classes abaixo, crie 3 construtores diferentes para a classe Filho:

```
public class Pai {
    private int i;
    private double d;
    private boolean b;

    public Pai(int ii, double dd) {
        i = ii;
        d = dd;
        b = false;
    }

    public Pai(int ii) {
        i = ii;
        d = 0.;
        b = false;
    }
}

public class Filho extends Pai {
    private int k;
}
```

- ☐ Crie uma superclasse **Pessoa** e uma subclasse **Aluno** que herde de **Pessoa**. A pessoa tem *nome* e *ano* de nascimento. O aluno tem *nota*. Escreva construtores para ambas as classes e métodos de acesso (get) para cada atributo. Escreva um método *mostraDados()* em **Pessoa**, que devolva um String contendo os valores dos atributos, um em cada linha, com título, e sobrescreva-o em **Aluno**.
- ☐ Um sistema de cadastro de clientes de uma empresa guarda informações sobre nomes, endereços e telefones. Além disso, caso o cliente seja uma pessoa física, seu número de CPF é armazenado e, caso o cliente seja uma pessoa jurídica, seu CNPJ e nome de fantasia da empresa devem ser guardados. A

qualquer momento deve ser possível imprimir todos dados de um determinado cliente. Implemente as classes necessárias em Java para a modelagem descrita utilizando os conceitos de polimorfismo.

Aula 10

- ☐ Crie uma Classe Pessoa, contendo os atributos encapsulados, com seus respectivos seletores (getters) e modificadores (setters), e ainda o construtor padrão e pelo menos mais duas opções de construtores conforme sua percepção. Atributos: String nome; String endereço; String telefone;
- ☐ Considere, como subclasse da classe Pessoa (desenvolvida no exercício anterior) a classe Fornecedor. Considere que cada instância da classe Fornecedor tem, para além dos atributos que caracterizam a classe Pessoa, os atributos valorCredito (correspondente ao crédito máximo atribuído ao fornecedor) e valorDivida (montante da dívida para com o fornecedor). Implemente na classe Fornecedor, para além dos usuais métodos seletores e modificadores, um método obterSaldo() que devolve a diferença entre os valores dos atributos valorCredito e valorDivida. Depois de implementada a classe Fornecedor, crie um programa de teste adequado que lhe permita verificar o funcionamento dos métodos implementados na classe Fornecedor e os herdados da classe Pessoa.
- ☐ Considere, como subclasse da classe Pessoa, a classe Empregado. Considere que cada instância da classe Empregado tem, para além dos atributos que caracterizam a classe Pessoa, os atributos codigoSetor (inteiro), salarioBase (vencimento base) e imposto (porcentagem retida dos impostos). Implemente a classe Empregado com métodos seletores e modificadores e um método calcularSalario. Escreva um programa de teste adequado para a classe Empregado.
- ☐ Implemente a classe Administrador como subclasse da classe Empregado. Um determinado administrador tem como atributos, para além dos atributos da classe Pessoa e da classe Empregado, o atributo ajudaDeCusto (ajudas referentes a viagens, estadias, ...). Note que deverá redefinir na classe Administrador o método herdado calcularSalario (o salário de um administrador é equivalente ao salário de um empregado usual acrescido das ajuda de custo). Escreva um programa de teste adequado para esta classe.
- ☐ Implemente a classe Operario como subclasse da classe Empregado. Um determinado operário tem como atributos, para além dos atributos da classe

Pessoa e da classe Empregado, o atributo valorProducao (que corresponde ao valor monetário dos artigos efetivamente produzidos pelo operário) e comissao (que corresponde à porcentagem do valorProducao que será adicionado ao vencimento base do operário). Note que deverá redefinir nesta subclasse o método herdado calcularSalario (o salário de um operário é equivalente ao salário de um empregado usual acrescido da referida comissão). Escreva um programa de teste adequado para esta classe.

- ☐ Implemente a classe Vendedor como subclasse da classe Empregado. Um determinado vendedor tem como atributos, para além dos atributos da classe Pessoa e da classe Empregado, o atributo valorVendas (correspondente ao valor monetário dos artigos vendidos) e o atributo comissao (porcentagem do valorVendas que será adicionado ao vencimento base do Vendedor). Note que deverá redefinir nesta subclasse o método herdado calcularSalario (o salário de um vendedor é equivalente ao salário de um empregado usual acrescido da referida comissão). Escreva um programa de teste adequado para esta classe.