ATIVIDADE DE RECUPERAÇÃO

Revisão de conceitos:

Classe: É um modelo ou estrutura que define os atributos e métodos de um objeto. Ela serve como um "molde" para a criação de objetos e pode ter um construtor para inicializar os atributos ao criar uma instância.

Objeto: É uma instância específica de uma classe. Cada objeto tem seus próprios valores para os atributos definidos pela classe e é criado com a ajuda do construtor.

Construtor: É um método especial usado para inicializar objetos, atribuindo valores iniciais aos seus atributos quando o objeto é criado.

Método: Funções dentro de uma classe que definem o comportamento dos objetos. Eles operam sobre os atributos e podem realizar ações no objeto.

Instância: É um objeto específico criado a partir de uma classe. Cada instância tem seus próprios valores para os atributos da classe.

Herança: Mecanismo em que uma classe filha herda atributos e métodos de uma classe pai. A subclasse pode sobrescrever ou adicionar novos comportamentos.

Exercício 1 - Sistema de Produtos em uma Loja

Crie um sistema em Java para gerenciar os produtos de uma loja, utilizando herança para representar os tipos de produtos e instâncias para criar objetos. O sistema deve:

Classe Base Produto:

Crie uma classe Produto com os seguintes atributos:

- nome (String)
- preco (double)

Implemente um construtor que inicializa esses atributos e um método exibirDetalhes() que exibe o nome e o preço do produto.

Subclasse ProdutoComDesconto (herda de Produto):

Crie uma subclasse ProdutoComDesconto que herda da classe Produto e possui um atributo adicional:

- desconto (double)

Implemente um construtor que inicializa os atributos da superclasse e o atributo desconto. Crie também um método exibirDetalhesComDesconto() que exibe o nome, o preço com desconto aplicado e o valor do desconto.

Instanciação e Teste:

Crie uma instância de Produto e uma de ProdutoComDesconto, inicialize seus atributos através do construtor e exiba seus detalhes utilizando os métodos exibirDetalhes() e exibirDetalhesComDesconto().

Exercício 2 - Sistema de Estacionamento

Crie um sistema em Java para gerenciar veículos em um estacionamento. Utilize herança para representar diferentes tipos de veículos. O sistema deve:

Classe Base Veiculo:

Crie uma classe Veiculo com os seguintes atributos:

- modelo (String)
- placa (String)
- ano (int)

Implemente um construtor que inicializa esses atributos e um método exibirInformacoes() que imprime o modelo, a placa e o ano do veículo.

Subclasses de Veiculo:

Crie as seguintes subclasses que herdam de Veiculo:

Carro (herda de Veiculo):

Atributos adicionais:

- numeroPortas (int)

Implemente um construtor que inicializa os atributos da superclasse e o atributo numeroPortas. Crie um método exibirInformacoesCarro() que exibe todas as informações do carro.

Moto (herda de Veiculo):

Atributos adicionais:

- cilindrada (int)

Implemente um construtor que inicializa os atributos da superclasse e o atributo cilindrada. Crie um método exibirInformacoesMoto() que exibe todas as informações da moto.

Instanciação e Teste:

Crie instâncias de Carro e Moto, inicialize seus atributos através do construtor e exiba as informações de cada veículo utilizando os métodos exibirInformacoes(), exibirInformacoesCarro() e exibirInformacoesMoto().

Exercício 3 - Sistema de Funcionários

Crie um sistema em Java para gerenciar funcionários em uma empresa, utilizando herança para representar diferentes tipos de cargos. O sistema deve:

Classe Base Funcionario:

Crie uma classe Funcionario com os seguintes atributos:

- nome (String)
- idade (int)
- salario (double)

Implemente um construtor que inicializa esses atributos e um método exibirInformacoes() que imprime o nome, idade e salário do funcionário.

Subclasses de Funcionario:

Crie as seguintes subclasses que herdam de Funcionario:

- Gerente (herda de Funcionario):

Atributos adicionais:

- departamento (String)

Implemente um construtor que inicializa os atributos da superclasse e o atributo departamento. Crie um método exibirInformacoesGerente() que exibe o nome, idade, salário e departamento do gerente.

Desenvolvedor (herda de Funcionario):

Atributos adicionais:

- linguagemProgramacao (String)

Implemente um construtor que inicializa os atributos da superclasse e o atributo linguagemProgramacao. Crie um método exibirInformacoesDesenvolvedor() que exibe o nome, idade, salário e linguagem de programação do desenvolvedor.

Instanciação e Teste:

Crie instâncias de Gerente e Desenvolvedor, inicialize seus atributos através do construtor e exiba as informações de cada funcionário utilizando os métodos exibirInformacoes(), exibirInformacoesGerente() e exibirInformacoesDesenvolvedor().

Exercício 4 - Sistema de Formas Geométricas

Crie um sistema em Java para calcular as áreas de diferentes formas geométricas, utilizando herança para representar as características específicas de cada tipo de forma. O sistema deve:

Classe Base Forma:

Crie uma classe Forma com o seguinte atributo:

- nome (String)

Implemente um construtor que inicializa o atributo nome e um método exibirNome() que imprime o nome da forma.

Subclasses de Forma:

Crie as seguintes subclasses que herdam de Forma:

Circulo (herda de Forma):

Atributos adicionais:

- raio (double)

Implemente um construtor que inicializa os atributos da superclasse e o atributo raio. Crie um método calcularArea() que retorna a área do círculo (use a fórmula π×raio2).

Retangulo (herda de Forma):

Atributos adicionais:

- largura (double)
- altura (double)

Implemente um construtor que inicializa os atributos da superclasse e os atributos largura e altura. Crie um método calcularArea() que retorna a área do retângulo (use a fórmula largura ×altura).

Quadrado (herda de Retangulo):

Como o quadrado é um caso especial de retângulo (onde largura = altura), herde a classe Retangulo. A classe Quadrado pode ter apenas um atributo adicional:

- lado (double)

Implemente um construtor que inicializa o atributo da superclasse e o atributo lado. Sobrescreva o método calcularArea() para calcular a área do quadrado (use a fórmula $lado^2$).

Instanciação e Teste:

Crie instâncias de Circulo, Retangulo e Quadrado, inicialize seus atributos através do construtor e exiba o nome e a área de cada forma utilizando os métodos exibirNome() e calcularArea().