

1. Criar uma classe abstrata `FiguraGeometrica`:

Defina atributos abstratos como `base` e `altura`.

Implemente um método abstrato `calcular_area()`.

Crie subclasses para representar figuras específicas (quadrado, retângulo, triângulo).

Implemente o método `calcular_area()` em cada subclasse de acordo com a fórmula específica da figura.

2. Criar uma classe abstrata `ContaBancaria`:

Defina atributos abstratos como `titular`, `saldo`, `taxa_juros`.

Implemente métodos abstratos

como `depositar()`, `sacar()`, `calcular_juros()`.

Crie subclasses para representar diferentes tipos de contas (corrente, poupança, investimento).

Implemente os métodos concretos em cada subclasse, considerando as regras específicas de cada tipo de conta.

3. Criar uma classe abstrata `Veiculo`:

Defina atributos abstratos como `marca`, `modelo`, `ano`, `cor`.

Implemente métodos abstratos

como `ligar()`, `desligar()`, `acelerar()`, `frear()`.

Crie subclasses para representar diferentes tipos de veículos (carro, moto, caminhão).

Implemente os métodos concretos em cada subclasse, considerando as características específicas de cada tipo de veículo.

4. Criar uma classe abstrata `Funcionario`:

Defina atributos abstratos como `nome`, `cargo`, `salario`.

Implemente métodos abstratos

como `calcular_pagamento()`, `tirar_ferias()`.

Crie subclasses para representar diferentes tipos de funcionários (CLT, PJ, freelancer).

Implemente os métodos concretos em cada subclasse, considerando as regras de pagamento e benefícios específicos de cada tipo de contrato.

5. Criar uma classe abstrata `Jogo`:

Defina atributos abstratos

como `nome`, `genero`, `plataforma`, `classificacao_indicativa`.

Implemente métodos abstratos

como `iniciar()`, `pausar()`, `salvar_jogo()`, `carregar_jogo()`.

Crie subclasses para representar diferentes tipos de jogos (RPG, aventura, estratégia).

Implemente os métodos concretos em cada subclasse, considerando as características e mecânicas específicas de cada tipo de jogo.