**Atividade: Criação de uma Classe "Livro"**

**Objetivo:** Criar uma classe em PHP para representar um **Livro** com atributos e métodos simples. A atividade ajudará você a entender como funcionam as classes e os objetos em PHP.

**Passos:**

1. **Criação da classe Livro:**
   * A classe deve ter os seguintes atributos:
     + titulo (string)
     + autor (string)
     + ano\_publicacao (int)
     + preco (float)
   * Defina todos os atributos como **privados** (encapsulamento).
2. **Criação do Construtor:**
   * Crie um construtor que inicialize os atributos do livro quando o objeto for criado.
3. **Métodos da Classe:**
   * Crie os seguintes métodos:
     + **getTitulo()**: Retorna o título do livro.
     + **getAutor()**: Retorna o autor do livro.
     + **getAnoPublicacao()**: Retorna o ano de publicação.
     + **getPreco()**: Retorna o preço.
     + **setPreco($preco)**: Permite modificar o preço do livro.
4. **Instanciação e Utilização:**
   * Crie ao menos dois objetos Livro e defina seus atributos usando o construtor.
   * Utilize os métodos para acessar e exibir as informações de cada livro.
   * Altere o preço de um dos livros utilizando o método setPreco() e exiba o novo preço.

**Atividade: Sistema de Animais com Herança**

**Objetivo:** Criar um sistema simples de classes que utiliza herança para representar diferentes tipos de **animais**.

**Passos:**

1. **Classe Animal:**
   * Crie uma classe Animal com os seguintes atributos:
     + nome (string)
     + idade (inteiro)
   * Crie um método fazerSom(), mas ele não precisa fazer nada na classe Animal. O método será sobrescrito nas classes filhas.
2. **Classe Cachorro (herdando de Animal):**
   * A classe Cachorro deve herdar a classe Animal e sobrescrever o método fazerSom(), para que ele exiba a mensagem "Au au!".
3. **Classe Gato (herdando de Animal):**
   * A classe Gato também deve herdar de Animal e sobrescrever o método fazerSom(), para que ele exiba a mensagem "Miau!".
4. **Instanciando os objetos:**
   * Crie objetos das classes Cachorro e Gato, passando o nome e a idade dos animais. Chame o método fazerSom() para cada objeto.
5. **Exibir as informações:**
   * Exiba o nome, idade e o som de cada animal utilizando os métodos apropriados.

**Atividade: Gerenciamento de Funcionários com Visibilidade**

**Objetivo:** Criar uma classe Funcionario com atributos de diferentes níveis de visibilidade (público, protegido e privado), e demonstrar como controlar o acesso a esses atributos.

**Passos:**

1. **Classe Funcionario:**
   * A classe deve ter os seguintes atributos:
     + **nome** (público): pode ser acessado diretamente.
     + **salario** (protegido): só pode ser acessado dentro da classe e suas subclasses.
     + **senha** (privado): só pode ser acessado dentro da própria classe Funcionario.
   * Crie um método **exibirInformacoes()** para exibir o nome do funcionário e seu salário.
2. **Métodos Getters e Setters:**
   * Crie um **getter** para o atributo salario (já que ele é protegido e não pode ser acessado diretamente).
   * Crie um **setter** para o atributo senha (como senha é privada, deve haver um método que permita alterá-la de maneira controlada).
3. **Método aumentarSalario():**
   * Crie um método público aumentarSalario($percentual) que aumenta o salário do funcionário com base em um percentual fornecido como parâmetro.
4. **Instanciando e Testando a Classe:**
   * Crie um objeto da classe Funcionario e faça a alteração de seu salário e senha. Exiba as informações do funcionário.
   * Teste a tentativa de acesso aos atributos diretamente, tanto válidos quanto inválidos (ex.: tentativas de acessar salario diretamente, ou modificar senha diretamente).

**Atividade: Sistema de Cadastro de Produtos com Construtor**

**Objetivo:** Criar uma classe Produto que utiliza um construtor para inicializar os atributos de um produto, como nome, preço e quantidade em estoque. O construtor deve ser usado para garantir que os valores iniciais sejam definidos no momento da criação do objeto.

**Passos:**

1. **Classe Produto:**
   * A classe deve ter os seguintes atributos:
     + **nome** (string): O nome do produto.
     + **preco** (float): O preço do produto.
     + **quantidade** (inteiro): A quantidade do produto em estoque.
2. **Construtor:**
   * O construtor deve receber três parâmetros (nome, preço e quantidade) e inicializar os atributos correspondentes da classe.
3. **Métodos:**
   * Crie um método **exibirInformacoes()** para exibir o nome, preço e quantidade do produto.
   * Crie um método **atualizarQuantidade($quantidade)** para atualizar a quantidade de produtos em estoque.
   * Crie um método **calcularValorTotal()** que retorna o valor total do estoque, ou seja, preço do produto multiplicado pela quantidade disponível.
4. **Instanciando e Testando a Classe:**
   * Crie objetos da classe Produto, passando valores para o construtor.
   * Utilize os métodos para exibir as informações e atualizar o estoque.
   * Calcule o valor total do estoque de cada produto.