1. **Explique os conceitos abaixo:**

a. **Objeto:** é uma instância de uma classe. Um objeto é uma entidade do mundo real que tem uma identidade e um conjunto de propriedades que o caracterizam. Em programação orientada a objetos, um objeto é criado a partir de uma classe e pode ser manipulado através de métodos.

b. **Classe:** é uma estrutura que abstrai um conjunto de objetos com características similares. Uma classe define o comportamento de seus objetos através de métodos e os estados possíveis destes objetos através de atributos. Em outras palavras, uma classe descreve os serviços providos por seus objetos e quais informações eles podem armazenar.

c. **Herança** :é um mecanismo para o compartilhamento de métodos e atributos entre classes e subclasses, permitindo a criação de novas classes através da programação das diferenças entre a nova classe e a classe-pai. Na herança simples, uma classe pode herdar atributos e métodos de apenas uma superclasse. Na herança múltipla, uma classe pode herdar atributos e métodos de várias superclasses.

d. **Encapsulamento:** é o princípio pelo qual os dados e as operações são encapsulados em um módulo chamado classe que pode ser visto como um padrão para criar objetos (instanciação). O encapsulamento permite esconder os detalhes internos da implementação do objeto, protegendo-o contra modificações externas.

e. **Polimorfismo:** é o princípio pelo qual duas ou mais classes derivadas de uma mesma superclasse podem invocar métodos que têm a mesma identificação, assinatura, mas comportamentos distintos, especializados para cada classe derivada, usando para tanto uma referência a um objeto do tipo da superclasse.

2.**O que são métodos mágicos no PHP? Dê exemplos**.

Métodos mágicos são métodos especiais que sobrescrevem o comportamento padrão do PHP quando certas operações são realizadas em um objeto. Alguns exemplos de métodos mágicos incluem:

- `__construct()`: usado para inicializar uma classe quando um objeto é criado.
- `__toString()`: usado para converter um objeto em uma string.
- `__get()`: usado para acessar uma propriedade inacessível.
- `__set()`: usado para definir o valor de uma propriedade inacessível.
- `__call()`: usado para interceptar chamadas a métodos inexistentes.

3. Explique a diferença de *public*, *protected* ou *private*.

- `public`: Atributos e métodos públicos são acessíveis de qualquer lugar, dentro ou fora da classe. Eles podem ser acessados diretamente por outras classes ou objetos.
- `protected`: Atributos e métodos protegidos são acessíveis apenas dentro da classe em que foram definidos e em suas subclasses. Eles não podem ser acessados diretamente por outras classes ou objetos.

- `private`: Atributos e métodos privados são acessíveis apenas dentro da classe em que foram definidos. Eles não podem ser acessados diretamente por outras classes ou objetos.

4. Pesquise sobre as vantagens e benefícios da programação orientada a objetos. Cite as referências.

A programação orientada a objetos (POO) é um paradigma de programação que utiliza o conceito de objetos para representar elementos do mundo real. Alguns dos benefícios da POO incluem:

- Reutilização de código: A POO permite que o código seja organizado em módulos reutilizáveis, chamados de classes. Essas classes podem ser usadas em diferentes partes do programa, evitando a duplicação de código e tornando o desenvolvimento mais eficiente.
- Modularidade: A POO permite que o código seja dividido em módulos independentes, cada um com sua própria responsabilidade. Isso torna o código mais fácil de entender, manter e modificar.
- Abstração: A POO permite que os objetos sejam abstraídos do mundo real, simplificando a complexidade do problema a ser resolvido. Isso torna o código mais fácil de entender e manter.
- Polimorfismo: O polimorfismo é a capacidade de um objeto ser tratado como outro objeto. Isso permite que diferentes objetos sejam tratados de forma semelhante, simplificando o código e tornando-o mais flexível.
- Herança: A herança é um mecanismo para o compartilhamento de métodos e atributos entre classes e subclasses, permitindo a criação de novas classes através da programação das diferenças entre a nova classe e a classe-pai.

Referências:

1. Vantagens e Desvantagens da POO - DevMedia
2. Programação orientada a objetos: o que é e por que é importante - Gorila Code
3. Introdução a Programação Orientada a Objetos (POO) - DIO
4. Vantagens da Programação Orientada a Objetos (POO) - DIO