**Atividades Revisão**

1- O que é orientação a objetos e quais são seus princípios fundamentais?

R = Orientação a Objetos (OO): É um paradigma de programação baseado no conceito de "objetos", que podem conter dados na forma de campos, também conhecidos como atributos, e código na forma de procedimentos, também conhecidos como métodos. Os princípios fundamentais da orientação a objetos são:

Abstração: Permite modelar entidades do mundo real como objetos na programação, abstraindo detalhes irrelevantes e enfatizando características importantes.

Encapsulamento: Oculta a implementação interna dos objetos, permitindo o acesso somente por meio de uma interface bem definida.

Herança: Permite que uma classe herde atributos e métodos de outra classe, promovendo a reutilização de código e estabelecendo relações entre classes.

Polimorfismo: Permite que objetos de diferentes classes sejam tratados de forma uniforme, com base em uma interface comum, facilitando a extensibilidade e flexibilidade do código.

2-Quais são as vantagens da orientação a objetos em comparação com paradigmas de programação procedural?

R = Reutilização de código: Através de conceitos como herança e composição, o código pode ser reutilizado em diferentes partes do programa.

Abstração e encapsulamento: Permitem uma melhor organização e modularidade do código, facilitando sua compreensão e manutenção.

Facilidade de manutenção: Devido à sua estrutura modular e encapsulada, a manutenção do código orientado a objetos é mais fácil e menos propensa a erros.

Modelagem do mundo real: A OO permite modelar entidades do mundo real de forma mais precisa, tornando o código mais intuitivo e próximo do problema real que está sendo resolvido.

3- Explique o conceito de encapsulamento e como ele é implementado em linguagens de programação orientadas a objetos.

R = É um dos princípios fundamentais da programação orientada a objetos, que consiste em ocultar os detalhes de implementação de um objeto e expor apenas uma interface para interagir com ele. Em linguagens orientadas a objetos, como Java, isso é implementado através do uso de modificadores de acesso (como public, private, protected) para controlar o acesso aos membros de uma classe.

4- Explique o conceito de classe e objeto em programação orientada a objetos, e como eles estão relacionados.

R = Em programação orientada a objetos, uma classe é uma estrutura que define um tipo de objeto. Ela descreve os atributos e comportamentos que os objetos desse tipo terão. Um objeto, por sua vez, é uma instância de uma classe, ou seja, uma entidade específica que foi criada com base na estrutura definida pela classe.

5- Qual é a importância dos construtores e métodos em uma classe em programação orientada a objetos?

R = Os construtores são métodos especiais usados para inicializar objetos quando eles são criados. Eles têm o mesmo nome da classe e não têm tipo de retorno. Os métodos, por sua vez, são funções definidas dentro de uma classe que executam operações específicas sobre os objetos dessa classe, permitindo a interação e manipulação dos dados do objeto.

6- Quais são as diferenças entre métodos estáticos e métodos de instância em uma classe?

R = Métodos estáticos pertencem à classe em si, enquanto métodos de instância pertencem a instâncias individuais da classe. Métodos estáticos podem ser invocados sem criar uma instância da classe, enquanto métodos de instância precisam ser chamados em uma instância específica da classe.

7- Como são tratados os conceitos de acoplamento e coesão na orientação a objetos e por que são importantes?

R = O acoplamento refere-se ao grau de interdependência entre módulos ou classes, enquanto a coesão refere-se à medida em que os elementos de um módulo ou classe estão relacionados entre si. Na orientação a objetos, é importante reduzir o acoplamento e aumentar a coesão para facilitar a manutenção e a extensibilidade do código.

8- Qual é o objetivo do método main?

R = O método `main` é o ponto de entrada para a execução de um programa Java. Ele indica onde a execução do programa deve começar e geralmente contém o código para iniciar a aplicação, criar objetos, chamar métodos e assim por diante.

9- Como ocorre a passagem de parâmetros em métodos de uma classe em programação orientada a objetos?

R = Em programação orientada a objetos, os parâmetros são passados para métodos da mesma forma que em outras linguagens de programação. Eles podem ser passados por valor ou por referência, dependendo do tipo de dado e da linguagem de programação específica.

10- Qual é a diferença entre os modificadores de acesso "public", "private" e "protected" em uma classe?

R = public: Os membros são acessíveis de qualquer lugar.

private: Os membros só são acessíveis dentro da própria classe.

protected: Os membros são acessíveis dentro da própria classe e por subclasses.

11-Qual é a importância da palavra-chave "this" em programação orientada a objetos e como ela é utilizada?

R = A palavra-chave `this` em Java refere-se à instância atual do objeto em que o código está sendo executado. É usada para acessar membros de instância (variáveis e métodos) da própria classe. Isso é útil quando há ambiguidade entre nomes de variáveis locais e membros de instância.