

## Questão 01

### a) Classe:

É uma estrutura que define um conjunto de propriedades (atributos) e comportamentos (métodos) que descrevem um determinado tipo de objeto. Ela funciona como um modelo ou plano para a criação de objetos. Uma classe é uma abstração de um conceito ou entidade do mundo real e é usada para criar instâncias específicas desse conceito, chamadas de objetos.

### b) Objeto:

Um objeto é uma instância de uma classe. Ele representa uma entidade específica que possui características e comportamentos definidos pela classe à qual pertence. Os objetos são criados a partir de uma classe e possuem estado (atributos) e comportamento (métodos) próprios. Cada objeto é único e pode interagir com outros objetos por meio de mensagens.

### c) Método e atributo:

Métodos e atributos são componentes de uma classe que definem seu comportamento e estado, respectivamente. Um método é um bloco de código que realiza uma determinada ação ou operação quando é chamado. Ele pode receber parâmetros e retornar um valor. Já um atributo é uma variável que armazena informações relacionadas ao objeto. Ele representa o estado do objeto e pode ser acessado e modificado pelos métodos da classe.

### d) Encapsulamento:

Encapsulamento é um princípio da programação orientada a objetos que consiste em esconder os detalhes internos de uma classe e fornecer uma interface externa para interagir com ela. Isso significa que os atributos de uma classe devem ser protegidos e acessados apenas por meio de métodos públicos. O encapsulamento ajuda a garantir a integridade dos dados e permite que a implementação interna de uma classe seja modificada sem afetar o restante do código que a utiliza.

### e) Visibilidade de atributos e métodos (public, private e protected):

A visibilidade de atributos e métodos em uma classe determina quais partes do código podem acessá-los. Existem três níveis de visibilidade:

**Public:** Atributos e métodos públicos são acessíveis a partir de qualquer classe. Eles podem ser usados por outras classes e seus objetos.

**Private:** Atributos e métodos privados são acessíveis apenas dentro da própria classe. Eles não podem ser acessados por outras classes.

**Protected:** Atributos e métodos protegidos são acessíveis dentro da mesma classe, classes do mesmo pacote e subclasses. Eles fornecem um nível intermediário de visibilidade.

.

f) Herança:

Herança é um conceito fundamental na programação orientada a objetos que permite criar novas classes a partir de classes existentes. A classe que é estendida é chamada de classe base ou superclasse, e a nova classe criada é chamada de classe derivada ou subclasse. A subclasse herda os atributos e métodos da superclasse, permitindo reutilização de código e facilitando a criação de hierarquias de classes.

g) Polimorfismo:

Polimorfismo é a capacidade de um objeto ser referenciado de diferentes formas e responder de maneiras diferentes aos mesmos métodos. Isso permite tratar objetos de diferentes classes de forma uniforme, desde que compartilhem a mesma interface ou classe base. O polimorfismo é alcançado por meio de herança e sobrescrita de métodos.

h) Classes abstratas:

Uma classe abstrata é uma classe que não pode ser instanciada diretamente, mas serve como uma superclasse para outras classes. Ela fornece uma estrutura básica e pode conter métodos abstratos, que são declarados sem implementação e devem ser implementados pelas classes derivadas. As classes abstratas são úteis quando se deseja definir um conjunto comum de características e comportamentos para suas subclasses, mas não faz sentido criar objetos da classe abstrata em si.

i) Construtores:

Um construtor é um método especial em uma classe que é executado automaticamente quando um objeto dessa classe é criado. Ele é usado para inicializar os atributos do objeto e prepará-lo para uso. O construtor tem o mesmo nome da classe e pode receber parâmetros para definir os valores iniciais dos atributos. Em Java, é possível ter vários construtores na mesma classe, desde que tenham parâmetros diferentes, o que é chamado de sobrecarga de construtores.

j) Get e Set:

Os métodos get e set, também conhecidos como métodos de acesso e modificação, são usados para obter e definir o valor dos atributos de um objeto, respectivamente. O método get retorna o valor do atributo e o método set atualiza o valor do atributo com base no argumento fornecido. Esses métodos são usados para controlar o acesso aos atributos de uma classe e garantir a consistência dos dados. Eles permitem que os atributos sejam acessados e modificados apenas por meio de métodos, em vez de permitir acesso direto aos atributos.

k) Sobrecarga de métodos:

A sobrecarga de métodos ocorre quando uma classe tem dois ou mais métodos com o mesmo nome, mas com parâmetros diferentes. Isso permite que métodos com funcionalidades semelhantes, mas que operam em diferentes tipos de dados

ou com diferentes números de argumentos, sejam definidos na mesma classe. A decisão sobre qual método executar é feita com base nos argumentos fornecidos durante a chamada do método.

#### l) Sobrescrita de métodos:

A sobrescrita de métodos ocorre quando uma classe filha redefine um método que já está definido na classe pai. A classe filha fornece uma implementação diferente do método com base nas necessidades específicas da classe. Isso permite que a classe filha substitua o comportamento da classe pai para esse método específico. A sobrescrita de métodos é um conceito fundamental da herança e do polimorfismo.

#### m) Palavras reservadas (super, this e final):

A palavra "super" é usada em uma classe filha para se referir aos membros da classe pai. Ela é usada para invocar o construtor da classe pai, acessar métodos ou atributos da classe pai e evitar a ambiguidade quando uma classe possui um método com o mesmo nome do método da classe pai.

A palavra-chave "this" é usada para se referir ao objeto atual em um contexto de classe. Ela pode ser usada para acessar os atributos e métodos do próprio objeto e também para chamar construtores sobrecarregados na mesma classe.

A palavra-chave "final" é usada para declarar uma constante, tornar uma classe não extensível (ou seja, não pode ser usada como superclasse).

#### n) Relação de objetos: "ter", "usar" e "ser":

Na programação orientada a objetos, os objetos podem se relacionar de diferentes maneiras:

Relação de "ter": Um objeto pode ter outros objetos como seus atributos. Essa relação é chamada de composição ou agregação. Um objeto contém outros objetos como parte de sua estrutura interna.

Relação de "usar": Um objeto pode usar outros objetos para realizar uma determinada tarefa. Essa relação é chamada de associação. Um objeto pode invocar métodos de outro objeto para obter algum resultado ou executar uma operação específica.

Relação de "ser": A relação de "ser" é estabelecida por meio da herança. Uma classe filha é uma classe pai. A classe filha herda os atributos e métodos da classe pai e pode adicionar ou modificar seu comportamento.