1. Indrodução à Programação Orientada a Objetos (POO)

ⓐ A Programação orientada a objetos é um Paradigma de Programação que serve para facilitar a organização do desenvolvimento de software em torno de objetos por meio do encapsulamento, herança, Polimorfismo, e abstração. E esses objetos servem tanto para representar entidades do mundo real quanto entidades abstratas.

ⓑ Em POO a classe é como se fosse um molde o qual você define seus atributos e métodos de um objeto, permitindo organizar o código do agrupar os dados e comportamentos que são da mesma entidade. Objeto mada mais é do que uma ~~instanci~~ instância de uma classe e serve para representar algo, tornando o código de melhor compreensão. A abstração serve para representar entidades de forma simplificado com foco no essencial e sem detalhes irrelevantes, tornando mais fácil o entendimento. Encapsulamento é algo que envolve agrupar os dados e métodos relacionados a esses dados em uma única coisa para restringir o acesso direto aos atributos, protegendo assim a integridade dos dados. A herança, como o nome já diz, permite que uma classe herde atributos e métodos de outra classe, assim reaproveitando o código. O polimorfismo serve para que objetos de diferentes classes façam coisas diferentes a partir do mesmo método, isso serve para deixar o código mais flexível.

2. Polimorfismo

ⓐ O Polimorfismo é um dos Pilares de POO que permite que os objetos de classes diferentes sejam tratados de forma igual através de uma interface, ou seja, significa que um único método pode ter funções diferentes dependendo de que classe invoca ele.

ⓑ Overloading e Overriding:

I. Polimorfismo de sobrecarga (overloading): Acontece quando dois ou mais métodos têm o mesmo nome, mas têm diferentes assinaturas dentro da mesma classe.

II. Polimorfismo de Sobrescrita (overriding): Permite que uma classe reque um método já existente de outra classe e modifique ele

3. Classes Abstratas

a) Uma classe abstrata é uma classe que não pode ser diretamente instanciada, significa que não é possível criar objetos a partir dela, ela é mais como um "modelo" para outras classes e definindo uma interface comum que as outras classes devem seguir.

b) Deve ser declarada com uma palavra-chave que é "abstract" antes de definir a classe. Pode conter métodos sem implementação (abstratos) que são declarados sem corpo e devem ser completados pelas outras classes. As outras classes são obrigadas a implementar esses métodos abstratos, a menos que essas classes também sejam abstratas.

c) Diferenças: classes normais podem ser instanciadas para criar objetos diretamente, enquanto classes abstratas não podem. Classes abstratas são moldes para outras classes, enquanto classes comuns representam entidades concretas.

4. Interfaces

a) Uma interface nada mais é do que uma entidade que declara vários métodos que uma classe deve implementar, estabelecendo um tipo de contrato de classes com o mundo exterior, e quando uma classe utiliza uma interface, ela tem que dar as funcionalidades especificadas por essa interface utilizada.

b) As interfaces ajudam a estabelecer contratos entre as classes, porque atuam como os próprios contratos para definir métodos que as classes vão implementar, isso serve para que classes diferentes sejam utilizadas de forma intercambiável e se implementem à mesma interface, logo, isso promove flexibilidade e interoperabilidade entre os componentes do software.

c) Interface só pode declarar métodos abstratos, enquanto a classe abstrata pode implementar tanto os métodos abstratos quanto os métodos implementados. Uma classe só pode herdar uma única classe abstrata, mas pode herdar várias interfaces. Métodos da interface são públicos, abstratos e não podem ter outros tipos de acesso, enquanto as classes abstratas permitem outros modificadores de acesso pra seus métodos e atributos.

5. Comparação entre Classes Abstratas e Interfaces

a) As duas servem para criação de contratos e definição de métodos em comuns para classes diferentes, mas na prática são bem diferentes. Por exemplo na questão da herança, classes podem herdar de várias interfaces, mas não podem herdar de mais de uma classe abstrata.

b)

- Interfaces: Quando for necessário que classes que não têm hierarquia em comum precisam implementar um ou mais métodos em comum, ou seja, quando uma classe precisa de herdar de mais de um lugar.
- Classe Abstrata: Quando várias classes precisam de comportamentos e atributos em comum.

c)

- Vantagens da interface: as classes podem usar várias interfaces sem problema.
- Desvantagens: Não pode implementar métodos propriamente e isso pode levar à duplicação de código
- Vantagens da classe abstrata: Reduz a duplicação de código por poder implementar métodos.
- Desvantagens: Uma classe pode herdar de uma única classe abstrata

6.

d) Em resumo, os principais pontos abordados foram:

- Definição de POO, classe, objeto e outras coisas importantes em POO;
- Definição, importância e os dois tipos de Polimorfismo;
- Definição, importância, características e regras de classes abstratas
- Diferenças entre classes comuns e abstratas;
- Definição e importância de interfaces;
- Diferenças entre interfaces e classes abstratas, vantagens, desvantagens e quando usar cada uma

6 -

b) Todos os três são conceitos importantíssimos para POO. O Polimorfismo serve para que diferentes objetos possam responder a métodos de maneira específica, as classes abstratas são modelos que permitem que outras classes herdem seus comportamentos e atributos e que possam modificar e implementar seus métodos para usar do seu modo por meio do Polimorfismo de sobrescrita, e as interfaces definem contratos de métodos entre as classes tornando tudo mais flexível