1 - Na linguagem Java, a troca de mensagens entre objetos é realizada por meio de chamadas a métodos. Esse conceito é central na programação orientada a objetos (POO), pois permite que os objetos "conversem" entre si para executar ações ou trocar informações.

2 - Os sistemas orientados a objetos (OO) implementam sua solução com base em quatro pilares fundamentais: **abstração** , **encapsulamento** , **herança** e **polimorfismo** . Esses princípios permitem que os sistemas OO estruturem o código em classes e objetos que se comunicam por meio de mensagens (chamadas de métodos), promovendo a reutilização, a modularidade e a facilidade de manutenção. Vamos explorar cada um desses pilares e como eles são abordados para a implementação de soluções em sistemas OO.

**1. Abstração**

A abstração é o processo de simplificar sistemas complexos, focando apenas nas características essenciais dos objetos e ignorando os detalhes detalhados. Em um sistema OO, a abstração permite que classes sejam definidas de forma a representar conceitos do mundo real ou da lógica do sistema, destacando apenas os atributos e métodos necessários.

**2. Encapsulamento**

O encapsulamento significa agrupar dados e métodos que operam sobre esses dados em uma mesma unidade, ou seja, em uma classe. O encapsulamento também esconde os detalhes internos dos objetos, expondo apenas o que é necessário para o uso externo. Em Java, por exemplo, o encapsulamento é implementado usando modificadores de acesso como public, protected, e private.

**3. Herança**

A herança permite que uma classe ( classe filha) herde atributos e métodos de outra classe ( classe mãe). Isso promove a reutilização de código e permite criar configurações de classes, onde classes mais específicas herdam características de classes mais gerais.

**4. Polimorfismo**

O polimorfismo permite que objetos de classes diferentes tratem chamadas de método da mesma forma. Ele pode ser implementado de duas formas principais: sobrecarga de métodos (vários métodos com o mesmo nome, mas diferentes parâmetros) e sobrescrita de métodos (uma subclasse redefine um método da superclasse). Em Java, o polimorfismo é frequentemente utilizado em interfaces e classes abstratas, permitindo que um mesmo método funcione de maneira diferente, dependendo da classe do objeto..

**3 –** Nos exemplos apresentados, a troca de mensagens entre objetos das classes Carroe Pessoaocorre quando o objeto Pessoautiliza métodos do objeto Carropara executar ações. Isso é feito pela chamada de métodos Carrodentro de um método da classe Pessoa. Vamos revisar o trecho de código relevante e destacar onde essas trocas de mensagens acontecem.

4 - A **cardinalidade** é um conceito usado na modelagem de dados e na programação orientada a objetos (OO) para definir o número de relações possíveis entre entidades (ou objetos) em um sistema. Ela indica quantas instâncias de uma entidade podem estar associadas a instâncias de outra entidade. A cardinalidade é essencial para estruturar corretamente as relações e entender como diferentes partes de um sistema de interação.

### Tipos de Cardinalidade

Existem três tipos principais de cardinalidade:

1. **1:1 (Um para Um)**
   * **Definição** : Cada instância de uma entidade está associada a uma única instância de outra entidade.
   * **Exemplo** : Em um sistema de cadastro, uma pessoa pode ter um único número de identificação (CPF), e cada número de CPF está associado a uma única pessoa.
2. **1**

**(Um para Muitos)**

* + **Definição** : Uma instância de uma entidade está associada a várias instâncias de outra entidade, mas cada instância da segunda entidade está associada a uma única instância da primeira.
  + **Exemplo** : Uma biblioteca pode ter muitos livros, mas cada livro pertence a uma única biblioteca.

1. **Não**

**(Muitos para Muitos)**

* + **Definição** : Várias instâncias de uma entidade podem estar associadas a várias instâncias de outra entidade.
  + **Exemplo** : Em uma universidade, os alunos podem se matricular em várias disciplinas, e cada disciplina pode ter vários alunos matriculados.

**5 -** Aqui cinco objetos ao meu redor, com suas relações cardinais e como eles estão inter-relacionados:

1. **Computador e Monitor**
   * **Relação** : 1:1 (Um para Um)
   * **Descrição** : Um computador está conectado a um monitor específico. Cada monitor exibe a imagem de um único computador, criando uma relação direta.
2. **Mesa e Cadeiras**
   * **Relação** : 1

(Um para Muitos)

* + **Descrição** : Uma mesa tem várias cadeiras ao seu redor, mas cada cadeira está associada apenas a essa mesa específica, formando uma relação de um para muitos.

1. **Roteador e Dispositivos Conectados (como notebook e celular)**
   * **Relação** : 1

(Um para Muitos)

* + **Descrição** : Um roteador pode fornecer conexão a dispositivos, mas cada dispositivo está conectado a um único roteador no ambiente atual.

1. **Estante e Livros**
   * **Relação** : 1

(Um para Muitos)

* + **Descrição** : A estante contém vários livros, mas cada livro pertence apenas estante especificamente, caracterizando uma relação de um para muitos.

1. **Carregador e Dispositivos (celular, tablet)**
   * **Relação** : 1

(Um para Muitos)

* + **Descrição** : Um carregador pode ser usado para carregar vários dispositivos ao longo do tempo, mas cada dispositivo é carregado apenas por um carregador de cada vez.