

A herança é um conceito fundamental na Programação Orientada a Objetos, permitindo que uma classe herde atributos e métodos de outra classe. Usaremos o mesmo cenário dos veículos, mas agora focaremos exclusivamente na herança.

1. Superclasse Veiculo:

A superclasse Veiculo terá atributos básicos que serão comuns a todos os veículos e um método para mostrar essas informações.

```
public class Veiculo {  
  
    private String marca;  
    private String modelo;  
  
    public Veiculo(String marca, String modelo) {  
        this.marca = marca;  
        this.modelo = modelo;  
    }  
  
    public void exibirInfo() {  
        System.out.println("Marca: " + marca + ", Modelo: " + modelo);  
    }  
}
```

2. Subclasse Moto:

A subclasse Moto herda de Veiculo e tem um atributo adicional cilindradas.

```
public class Moto extends Veiculo {  
  
    private int cilindradas;  
  
    public Moto(String marca, String modelo, int cilindradas) {  
        super(marca, modelo);  
        this.cilindradas = cilindradas;  
    }  
  
    @Override  
    public void exibirInfo() {  
        super.exibirInfo();  
        System.out.println("Cilindradas: " + cilindradas);  
    }  
}
```

3. Subclasse Carro:

A subclasse Carro herda de Veiculo e tem um atributo adicional portas.

```
public class Carro extends Veiculo {
```

```

private int portas;

public Carro(String marca, String modelo, int portas) {
    super(marca, modelo);
    this.portas = portas;
}

@Override
public void exibirInfo() {
    super.exibirInfo();
    System.out.println("Número de portas: " + portas);
}
}

```

4. No método main, crie instâncias das subclasses e chame seus métodos:

```

public class TesteHeranca {

    public static void main(String[] args) {
        Moto minhaMoto = new Moto("Honda", "CBR", 1000);
        Carro meuCarro = new Carro("Toyota", "Corolla", 4);

        System.out.println("Info da Moto:");
        minhaMoto.exibirInfo();

        System.out.println("\nInfo do Carro:");
        meuCarro.exibirInfo();
    }
}

```

Ao executar o programa TesteHeranca, você observará que ambas as subclasses Moto e Carro herdam o método exibirInfo() da superclasse Veiculo, mas também têm sua própria implementação do método, graças à sobrescrita (override). Isso é uma demonstração do poder da herança, onde as subclasses herdam características da superclasse, mas também podem adicionar ou modificar comportamentos conforme necessário.