**PROBLEMA DO DIAMANTE**

**Herança múltipla**

Basicamente o uso de herança faz com que uma classe herde atributos e métodos de outra classe existente. Em algumas linguagens também há a possibilidade de se herdar de mais de uma classe, a chamada herança múltipla. Em C++ e Python são algumas das linguagens que suportam esse recurso, já em Java não temos.

**O problema da herança múltipla**

O uso de herança múltipla pode gerar uma grande dor de cabeça para os desenvolvedores.

Imagine uma classe Carro que implementa um método chamado obterNumeroDeRodas que retorna o valor 4. Outra classe chamada Moto que também implementa um método chamada obterNumeroDeRodas com a mesma assinatura, mas que retorna o valor 2. Agora você cria outra classe chamada Quadriciclo que faz uso da herança múltipla, herdando de Carro e Moto. Então quando o método obterNumeroDeRodas for chamado, qual método retornará? O da classe Carro ou o da classe Moto?

Essa situação gera uma confusão e faz com que o resultado seja imprevisível. Esse problema é chamado "Diamond problem" (Problema Diamante).

**SOLUCIONANDO**

Uma forma de contornar esse problema seria o uso de Interface. Nesse caso, não haveria implementação dos métodos, mas sim um contrato que toda classe que implementar as interfaces deverão assinar.Assim não há reuso de código. Para resolver esse impasse o Java 8 disponibiliza o chamado método default, que possibilita a implementação dos métodos em interfaces.