

**FACULDADE DE CIÊNCIAS SOCIAIS APLICADAS**  
**CURSO DE SISTEMAS DE INFORMAÇÃO**  
**DISCIPLINA: PARADIGMA DE LINGUAGEM DE PROGRAMAÇÃO – 2016.2**  
**PROFESSORA: DANIEL ABELLA**  
**Aluno (a):** Arthur Martins de Araújo

## ATIVIDADE 1 – Análise do problema do diamante

Em sala de aula, foi abordado o assunto sobre herança múltipla. O professor da disciplina de Paradigma de Linguagem de Programação (PLP), Daniel Abella, estava explicando a respeito de herança múltipla e o mesmo falou que não é possível criar uma classe em Java que herdasse de duas classes por exemplo. Vimos também que nas linguagens C e C++ é possível implementar herança múltipla, então ficou a dúvida, porque em JAVA não é possível implementar uma classe que herde de duas classes? Através da análise do problema do diamante, podemos chegar a resposta para a dúvida.

Basicamente o problema do diamante é o seguinte, você criou duas classes, B e C, onde você terá que implementar o método imprimir() nas duas classes, para que não você não reescreva o método, cria uma classe A (uma classe pai, com o método imprimir()) e faz com que as classes B e C herdem da classe A criada, fazendo com que não reescreva o método.

Daí o sistema cresce e é preciso criar uma nova classe, a D, e essa classe herde das duas classes B e C, por um motivo qualquer. Se chamar o método da classe superior (classe pai, herdada) ou um base.imprimir(), qual classe vai ser chamado? Esse problema irá gerar um problema e uma complexidade muito alta na hora da execução com relação a escolha do método que será chamado.

O problema foi resolvido com a utilização de Interface, ou seja, na Interface não há uma implementação a ser herdada, e sim, um contrato em que a classe que há implementa deve possuir sua própria implementação.