Problema do diamante Java

Sabemos que herança é uma das características da Programação Orientada a Objetos (POO), e que ela possibilita que uma classe (Classe “filha”) herde os métodos e atributos de outra classe (Classe “Pai”). Em algum momento, algum programador pode ter tentado herdar de uma única classe várias classes diferentes, se deparado com um problema chamado de “herança múltipla”. O Java não suporta herança múltipla, em outras palavras, ele não deixa que você estenda dois objetos ao mesmo tempo. Diferentemente do C++ que permite esse mecanismo.

Nós nos perguntamos então, por que o Java não suporta herança múltipla. Ele irá se confundir podendo haver conflitos entre dois ou mais membros. Suponha que você tem dois membros com o mesmo nome mas tentam atender necessidades diferentes. Isso faz com que ocorra famoso Problema do Diamante.

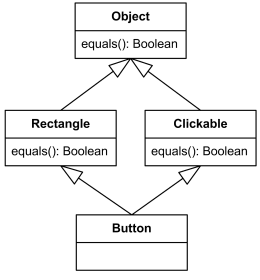


Figura 1: Diagrama de classes representando

o Problema do Diamante.

Então, como podemos resolver esse problema? O problema pode ser resolvido sobrescrevendo a classe que veio primeiro nas situações de conflito ou algo do gênero. O problema pode ser solucionável também com o uso das interfaces, pois com ela conseguimos os métodos de diferentes objetos.

Fontes: - <http://www.devmedia.com.br/heranca-conceitos-e-o-problema-diamante-parte-i/2697>

- <http://pt.stackoverflow.com/questions/42970/por-que-todo-mundo-odeia-heran%C3%A7a-m%C3%BAltipla-em-c-e-qual-sua-diferen%C3%A7a-para-mixin>

- <http://ptcomputador.com/P/java-programming/89666.html#.V7soCll7x88>