

INTRODUÇÃO À PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO

Atividade #1

Nome: Rafael Nilton Brito Santiago

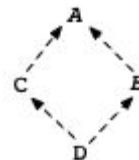
Diamond Problem

O Problema do diamante é referente aos problemas ocasionados ao se utilizar herança múltipla. Sendo este o motivo pelo qual a linguagem java não optou por suportar herança múltipla, pois muitas vezes eram gerados maiores problemas do que melhores soluções, de uma maneira geral é pelo fato de não saber como se comportar quando encontrar em uma classe herdada dois métodos herdados de classes diferentes com o mesmo nome.

Vamos supor que você precise criar duas classes B e C.

Ambas as classes, vão precisar utilizar um método `foo()` que você irá implementar. Então é criado uma classe A(uma classe pai). No qual, onde as classes B e C, herdarem de A para evitar ter que reescrever o método. No caso de precisar criar uma nova classe que herde das duas. Uma classe D, por exemplo, que herde das duas B e C por qualquer motivo. Ao chamar o método da classe superior (herdada) ou um `base.foo()`, de qual classe irá ser chamada? O problema acima vai gerar uma certa confusão no desenvolvedor e criar uma complexidade muito alta na hora da execução com relação a escolha do método pai. Este problema pode ser resolvido facilmente com Interface. Pois com a Interface, não existe implementação a ser herdada. Mas sim, um contrato de que a classe deve possuir sua própria implementação.

```
1 interface IB
2 {
3     void foo();
4 }
5
6 interface IC
7 {
8     void foo();
9 }
10
11 class D : IB, IC
12 {
13     public void foo()
14     {
15         // Your code
16     }
17 }
```



Fontes:

<http://blog.victormonte.com/heranca-multipla-e-o-problema-do-diamante/>

INTRODUÇÃO À PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO

Atividade #1

Nome: Rafael Nilton Brito Santiago

<http://www.lambdafaq.org/what-about-the-diamond-problem/>