Problema do Diamante

A linguagem JAVA tem uma característica que possibilita uma determinada classe herdar comportamentos (métodos e atributos) de uma outra classe. Segue abaixo um exemplo que podemos realizar usando essa característica do JAVA.

public class Pessoa {

private String nome;

private String cpf;

}

Agora vamos criar a classe gerente que vai herdar de pessoa

public class Gerente extends Pessoa{

public Integer getBonificacao() {

Integer valorBonificacao = 0 ;

double bonusGerente = 0.5;

return valorBonificacao \* bonusGerente;

}

}

Com isso quando criamos um novo gerente ele agora herda os atributos de pessoa, ou seja nome e cpf, essa é a herança em JAVA.

Agora vamos criar outra situação, criamos uma nova classe

public class Funcionario extends Pessoa {

public Integer getBonificacao() {

Integer valorBonificacao = 0 ;

return valorBonificacao ;

}

}

Criamos a classe funcionário que tem um mesmo método getBonificacao() que já tem na classe gerente, até aqui tudo bem, agora vamos fazer uma situação que vamos mostrar o problema da herança múltipla em JAVA conhecido como Problema do diamante.

Criamos a classe vendedor

public class Vendedor extends Gerente, Funcionario(){

}

Lembrando que esse código não compila, estamos mostrando apenas para exemplificar o problema,

Vendodor vendedor = new Vendendor();

vendedor.getBonificacao();

Agora quando chamamos o método getBonificaca(), o JAVA não sabe qual deve ser chamado, se o método da classe gerente ou do funcionário, esse é um dos motivos que em JAVA não existe a herança múltipla, que podemos contornar usando interfaces.