1.Herança é um recurso que permite um reaproveitamento de código onde a classe filha não precisa declarar atributos e funções que sua classe pai tenha, a filha herda tais atributos e métodos.Filhos herdam de seus pais algumas características(atributos) e as vezes formas de agir(métodos) .

2.Polimorfismo quer dizer varias formas.Um método com a mesma assinatura pode ter diferentes implementações ao mesmo tempo, só sendo decidida qual, em tempo de compilação ou execução.

3.

**public** **class** Forma {

**float** area;

**public** Forma() {

// **TODO** Auto-generated constructor stub

}

}

**public** **class** Retangulo **extends** Forma {

**float** base;

**float** altura;

**public** Retangulo(**float** b,**float** h) {

// **TODO** Auto-generated constructor stub

**super**();

base = b;

altura = h;

}

}

**public** **class** Quadrado **extends** Retangulo {

**public** Quadrado(**float** lado) {

// **TODO** Auto-generated constructor stub

**super**(lado, lado);

}

}

3.

**public** **class** Forma {

**float** area;

**public** Forma(){

}

}

**public** **class** Retangulo **extends** Forma {

**float** base;

**float** altura;

**public** Retangulo(**float** b,**float** h) {

// **TODO** Auto-generated constructor stub

**super**();

base = b;

altura = h;

}

}

**public** **class** Quadrado **extends** Retangulo {

**public** Quadrado(**float** lado)

{

**super**(lado,lado);

}

}

**public** **class** Principal {

**static** Quadrado *q1*;

**public** Principal() {

// **TODO** Auto-generated constructor stub

}

**public** **static** **void** main(String[] args) {

*q1* = **new** Quadrado(2);

System.***out***.println("Quadrado de lados "+*q1*.altura+" e "+*q1*.base);

}