POLIMORFISMO - Craitson Luiz Mayer

Polimorfismo é o princípio pelo qual duas ou mais classes derivadas de uma mesma superclasse podem invocar métodos que têm a mesma identificação (assinatura) mas comportamentos distintos, especializados para cada classe derivada, usando para tanto uma referência a um objeto do tipo da superclasse. A decisão sobre qual o método que deve ser selecionado, de acordo com o tipo da classe derivada, é tomada em tempo de execução, através do mecanismo de ligação tardia.

Referencia

http://www.dca.fee.unicamp.br/cursos/PooJava/polimorf/index.html

https://www.caelum.com.br/apostila-java-orientacao-objetos/heranca-reescrita-e-polimorfismo/

Exemplos:

Exemplo 1 -

```
public class Animal{    public void sound(){
 System.out .println("Animal is making a sound");    } }
class Horse extends Animal{
@Override    public void sound(){
System.out .println("Neigh");    }
public static void main(String args[]){
Animal obj = new Horse();
obj.sound();    } }
```

Exemplo 2 -

```
public class Animal {    public void comer() {
   System.out .println( "Animal Comendo..." );
} }
public class Cao  extends Animal {
public void comer()
{       System.out .println( "Cão Comendo..." );
} }
public class Tigre extends Animal {
public void comer() {
 System.out .println( "Tigre Comendo..." );    } }

public class Test {
public void fazerAnimalComer( Animal animal ) {
  animal.comer();    }
public static void main( String[] args ) {
Test t = new Test();
t.fazerAnimalComer( new Animal() );
      t.fazerAnimalComer( new Cao() );
       t.fazerAnimalComer( new Tigre() );    } }
```

Resultado : Animal Comendo… Cão Comendo… Tigre Comendo…