Полиморфизъм

(Много форми)

Позволява интерфейс или метод да бъде използван по много начини. Реализира се чрез наследяване, методи и интерфейси. Един метод или обект, може да се държи по различен начин в зависимост къде се ползва.

Видове:

1. Компилационен (Compile-time)/Статичен полиморфизъм – реализира се с претоварване на методи (method Overload)/Динамичен полиморфизъм същият метод има различни сигнатури
2. Полиморфизам по време на изпълнението на програмата (Runtime polymorphism) – реализира се чрез препокриване на методите (Override), като важи за методи в НАСЛЕДЯВАЩИ класове (ако ми се наложи да ползвам метод от родителския клас в

Претоварването на методи (Compiled-time polymorphism) позволява да дефинираме еднакви методи но с различни параметри в зависимост от

Статичният полиморфизъм, обхваща методи в един и същи клас.

public double add(double x, double y) {  
 return x + y;  
}  
  
public double add(double x, double y, double z) {  
 return x + y + z;  
}  
  
public int add(int x, int y){  
 return x + y;  
}

Препокриване на методи (Runtime polymorphism) се случва когато подклас(дъщерен) дефинира метод, съдържащ се в родителския (със същата сигнатура), но има различен изход.

public class Animal {  
  
 public void makesSound(){  
 System.*out*.println("Makes sound");  
 }  
}

Следва в дъщерните класове:

public class Dog extends Animal {  
  
 @Override  
 public void makesSound(){  
 System.*out*.println("Barks");  
 }  
}

public class Cat extends Animal {  
  
 @Override  
 public void makesSound() {  
 System.*out*.println("Meow");  
 }  
}

А в MAIN метода =>

public class Main {  
 public static void main(String[] args) {  
  
 Calculator calculator = new Calculator();  
  
 System.*out*.println(calculator.add(10.3, 20.4));  
 System.*out*.println(calculator.add(1.2,2.2,3.2));  
 System.*out*.println(calculator.add(1,2));  
  
 Animal myAnimal; // Полиморфично поле/променлива  
 myAnimal = new Dog();  
 myAnimal.makesSound(); // Barks (метода е преизползван)  
 myAnimal = new Cat();  
 myAnimal.makesSound(); // Meow (метода е преизползван)  
 }  
}

Интерфейсите са друг начин за реализация на полиморфизъм. Класовете могат да имплементират един и същ интерфейс и да предоставят различа реализация.