

Polimorfisme en Java

Què és el polimorfisme?

Les referències a objectes de les classes derivades es poden convertir (de fet, Java ho fa automàticament) al tipus de la classe base. Això permet dissenyar algorismes polimòrfics, que poden operar de manera genèrica sobre qualsevol objecte de qualsevol classe derivada. En aquest cas, el lligam entre la referència i l'objecte es produeix de forma tardana, en temps d'execució, i no en temps de compilació.

Exercici resolt Polimorf1: Granota i Amfibi

Crear una classe Amfibi amb l'atribut nom i deriveu-ne la classe Granota. Definir els mètodes d'accés a l'atribut a la classe Amfibi. Crear un programa de test per instanciar un objecte Granota i definir una referència Amfibi que apunti a l'objecte Granota anterior. Modificar el nom amb cada una de les referències i, per a cada cas, llegir el valor amb cada referència. Analitzar el resultat.

A continuació tenim el codi demanat.

```
/**
 * Polimorf1.java
 * Exemple de herencia i polimorfisme
 * @author Jose Moreno
 * @version
 */
public class Polimorf1 {
    public static void main(String[] args) {
        Granota unaGranota = new Granota();
        Amfibi unAmfibi = unaGranota; //referencia a l'objecte unaGranota
        //mètode setNom() de la classe base
        unAmfibi.setNom("Eusebi");
        System.out.println("Nom de l'amfibi: "+unAmfibi.getNom());
        System.out.println("Nom de la granota: "+unaGranota.getNom());
        //mètode setNom() des de la classe derivada
        unaGranota.setNom("Carlota");
        System.out.println("Nom de l'amfibi: "+unAmfibi.getNom());
        System.out.println("Nom de la granota: "+unaGranota.getNom());
    }
}

class Amfibi {
    String nom;
    public Amfibi() {
        System.out.println("Nou objecte Amfibi ");
    }
    public String getNom() {
        return nom;
    }
    public void setNom(String nom){
        this.nom = "Amfibi->" + nom;
    }
}

class Granota extends Amfibi {
    public Granota() {
        super();
        System.out.println("Nou objecte Granota ");
    }
}
```

```
}
```

La sortida del programa anterior ve a continuació. El mètode és membre de la classe base, però és accessible des de la classe derivada, perquè no s'hi ha redefinit. Comprovem que el lligam entre la invocació i el mètode es fa de forma dinàmica en temps d'execució, tenint en compte el tipus de dada referenciada, no el tipus base de la referència.

```
Nou objecte Amfibi
Nou objecte Granota
Nom de l'amfibi: Amfibi->Eusebi
Nom de la granota: Amfibi->Eusebi
Nom de l'amfibi: Amfibi->Carlota
Nom de la granota: Amfibi->Carlota
```

Exercici resolt Polimorf2: Granota i Amfibi

Modificar l'exemple anterior redefinint el mètode setNom a la classe Granota. Analitzar el resultat.

A continuació tenim el codi demanat.

```
/**
 * Polimorf2.java
 * Exemple de herencia i polimorfisme
 * @author Jose Moreno
 * @version
 */
public class Polimorf2 {
    public static void main(String[] args) {
        Granota unaGranota = new Granota();
        Amfibi unAmfibi = unaGranota; //referencia a l'objecte unaGranota
        //mètode setNom() de la classe base
        unAmfibi.setNom("Eusebi");
        System.out.println("Nom de l'amfibi: "+unAmfibi.getNom());
        System.out.println("Nom de la granota: "+unaGranota.getNom());
        //mètode setNom() des de la classe derivada
        unaGranota.setNom("Carlota");
        System.out.println("Nom de l'amfibi: "+unAmfibi.getNom());
        System.out.println("Nom de la granota: "+unaGranota.getNom());
    }
}

class Amfibi {
    String nom;
    public Amfibi() {
        System.out.println("Nou objecte Amfibi ");
    }
    public String getNom() {
        return nom;
    }
    public void setNom(String nom){
        this.nom = "Amfibi->"+nom;
    }
}

class Granota extends Amfibi {
    public Granota() {
        super();
        System.out.println("Nou objecte Granota ");
    }
}
```

```

    }
    public String getNom() {
        return nom;
    }
    public void setNom(String nom){
        this.nom = "Granota->" + nom;
    }
}

```

La sortida del programa anterior ve a continuació. El mètode que s'executa és sempre el de Granota. El lligam entre la invocació i el mètode es fa de forma tardana i dinàmica, segons la classe de l'objecte (derivat) que fa la invocació, tot i que la referència està definida del tipus base.

```

Nou objecte Amfibi
Nou objecte Granota
Nom de l'amfibi: Granota->Eusebi
Nom de la granota: Granota->Eusebi
Nom de l'amfibi: Granota->Carlota
Nom de la granota: Granota->Carlota

```

Exercici resolt Polimorf3: Tipus de referència superclasse a la subclasse

Crear una classe Base i derivateu-ne una classe Derivat, amb constructors sense paràmetres i que anunciïn la instanciació. Definir un mètode f() a les dues classes, que anunciï el nom de la classe i del mètode. Programar una classe principal per instanciar un objecte de cada classe. Invocar el mètode f() de cada classe, assigneu la classe derivada a la classe base i provar una altra vegada a invocar els mètodes. Analitzar la sortida.

A continuació tenim el codi demanat.

```

/**
 * Polimorf3.java
 * Exemple de herencia i polimorfisme
 * @author Jose Moreno
 * @version
 */
public class Polimorf3 {
    public static void main(String[] args) {
        Derivat deriv = new Derivat();
        Base bas = new Base();
        bas.f();
        bas = deriv; //referencia a l'objecte deriv
        deriv.f();
        bas.f();
    }
}
class Base {
    public Base() {
        System.out.println("Nou objecte Base ");
    }
    public void f() {
        System.out.println("Base.f()");
    }
}
class Derivat extends Base {
    public Derivat() {

```

```

        super();
        System.out.println("Nou objecte Derivat ");
    }
    public void f() {
        System.out.println("Derivat.f()");
    }
}

```

La sortida del programa anterior ve a continuació. Els objectes s'instancien en l'ordre correcte. A l'última línia comprovem que el mètode que s'invoça quan bas apunta a deriv és el de classe Derivat. Això implica que s'ha efectuat el lligam de forma dinàmica analitzant el tipus de dades referenciada, amb independència del tipus base de la referència.

```

Nou objecte Base
Nou objecte Derivat
Nou objecte Base
Base.f()
Derivat.f()
Derivat.f()

```

Exercici resolt Polimorf4: Tipus de referència subclasse a la superclasse

Modificar l'exemple anterior per provar si es produeix lligam dinàmic en assignar la classe base a la derivada.

A continuació tenim el codi demanat.

```

/**
 * Polimorf4.java
 * Exemple de herencia i polimorfisme
 * @author Jose Moreno
 * @version
 */
public class Polimorf4 {
    public static void main(String[] args) {
        Derivat deriv = new Derivat();
        Base bas = new Base();
        deriv.f();
        deriv = bas; //error en temps de compilacio
        deriv.f();
        bas.f();
    }
}
class Base {
    public Base() {
        System.out.println("Nou objecte Base ");
    }
    public void f() {
        System.out.println("Base.f()");
    }
}
class Derivat extends Base {
    public Derivat() {
        super();
        System.out.println("Nou objecte Derivat ");
    }
    public void f() {
        System.out.println("Derivat.f()");
    }
}

```

```
}  
}
```

En aquest cas, es produeix un error en temps de compilació. No és permès utilitzar un tipus derivat per referenciar un tipus base.

Podem provar de fer un canvi de tipus (type cast) forçat.

```
/**  
 * Polimorf4b.java  
 * Exemple de herència i polimorfisme  
 * @author Jose Moreno  
 * @version  
 */  
public class Polimorf4b {  
    public static void main(String[] args) {  
        Derivat deriv = new Derivat();  
        Base bas = new Base();  
        deriv.f();  
        deriv = (Derivat)bas; //excepcio en temps d'execucio  
        deriv.f();  
        bas.f();  
    }  
}  
class Base {  
    public Base() {  
        System.out.println("Nou objecte Base ");  
    }  
    public void f() {  
        System.out.println("Base.f()");  
    }  
}  
class Derivat extends Base {  
    public Derivat() {  
        super();  
        System.out.println("Nou objecte Derivat ");  
    }  
    public void f() {  
        System.out.println("Derivat.f()");  
    }  
}
```

Ara el programa compila sense errors. Tot i això, obtenim un error en temps d'execució, quan es detecta que s'està utilitzant una referència de tipus subclasse per accedir a un objecte que és de tipus superclasse.

```
Nou objecte Base  
Nou objecte Derivat  
Nou objecte Base  
Derivat.f()  
Exception in thread "main" java.lang.ClassCastException: Base  
at Polimorf4b.main(Polimorf4b.java:13)
```