Selecció dinàmica de mètodes

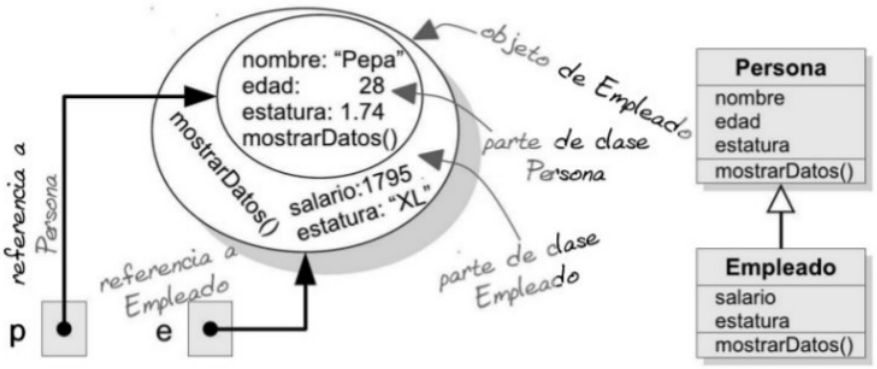
Quan definim una classe com a subclasse d'una altra, els objectes de la subclasse són també objectes de la superclasse.

Per exemple, un objecte Empleat serà, al mateix temps, un objecte de Persona, ja que posseeix totes les característiques de Persona, a més d'altres específics d'Empleat.

Per tant, podem referenciar un objecte Empleat usant una variable Persona.

Per exemple:

```
Empleado e = new Empleado();
Persona p = e;
```



És el mateix una variable Empleat que una variable Persona per a referenciar un objecte Empleat?

No. Hi ha una subtil, però important diferència.

En primer lloc, els atributs accessibles són els definits en la classe de la variable.

```
p.estatura   //atribut de Persona de tipo double
e.estatura   //atribut de Empleado de tipo String
```

Però, en canvi, amb els mètodes ocorre el contrari. S'executa la versió de l'objecte referenciat, és a dir, de la subclasse Empleat.

Per tant, sí que funciona el overriding(sobreescriptura).

```
Empleado e = new Empleado();
Persona p = e;
p.mostrarDatos();//método de Empleado
e.mostrarDatos();//método de Empleado
```

Això proporciona una de les eines més potents de què disposa Java per a exercir el polimorfisme: la selecció de mètodes en temps d'execució.

```
class Cliente extends Persona {
   ...
   @Override
   void mostrarDatos() {
      ...
   }
}
```

Si creem una variable de tipus Persona, amb ella podem referenciar tant objectes de classe Empleat, com Client, com a Persona.

Per a tots ells disposem del mètode mostrarDatos(), però s'executarà l'una o l'altra versió, segons l'objecte referenciat, que pot canviar en temps d'execució.

Així, la mateixa línia de codi p.mostrarDatos(), executarà mètodes diferents, segons la mena d'objecte referenciat.