Capítulo 14. herencias múltiples

permite la definición de una clase a partir de la definición de otra ya existente.

La herencia permite compartir automáticamente métodos y datos entre clases, subclases y objetos.  
Hay dos tipos de herencia: *Herencia Simple* y *Herencia Múltiple*.

La primera indica que se pueden definir nuevas clases solamente a partir de una clase inicial mientras que la segunda indica que se pueden definir nuevas clases a partir de dos o más clases iniciales.

En la herencia, las clases derivadas “heredan” los datos y las funciones miembro de las clases base, pudiendo las clases derivadas redefinir estos comportamientos (polimorfismo) y añadir comportamientos nuevos propios de las clases derivadas.

“protected”. Cualquier cosa que tenga visibilidad protected se comportará como pública en la clase Base y en las que componen la jerarquía de herencia, y como privada en las clases que NO sean de la jerarquía de la herencia.

existen 3 clases de herencia que se diferencian en el modo de manejar la visibilidad de los componentes de la clase resultante:

Herencia publica (class Derivada: public Base ) : Con este tipo de herencia se respetan los comportamientos originales de las visibilidades de la clase Base en la clase Derivada.

Herencia privada (clase Derivada: private Base) : Con este tipo de herencia todo componente de la clase Base, será privado en la clase Derivada (ojo! siempre será privado aunque ese dato fuese público en la clase Base) • Herencia protegida (clase Derivada: protected Base) : Con este tipo de

herencia, todo componente publico y protegido de la clase Base, será protegido en la clase Derivada, y los componentes privados, siguen siendo privados.

Herencia Múltiple La herencia múltiple es el mecanismo que permite al programador hacer clases derivadas a partir, no de una sola clase base, sino de varias.