

## 7.C. Herencia.

### 1. Herencia.

#### 1.2. Acceso a miembros heredados.

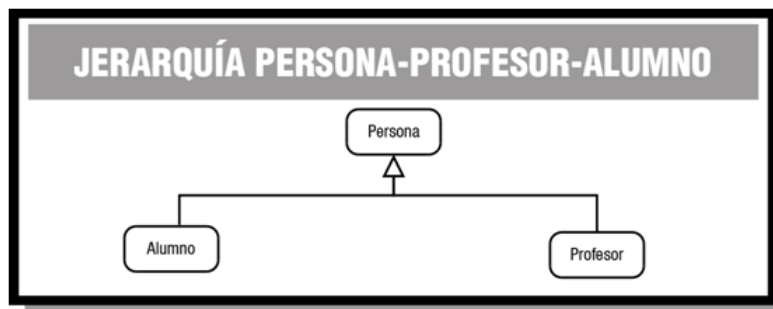
Como ya has visto anteriormente, **no es posible acceder a miembros privados de una superclase**. Para poder acceder a ellos podrías pensar en hacerlos **públicos**, pero entonces estarías dando la opción de acceder a ellos a cualquier objeto externo y es probable que tampoco sea eso lo deseable. Para ello se inventó el modificador **protected** (protegido) que permite el **acceso desde clases heredadas**, pero no desde fuera de las clases (estrictamente hablando, desde fuera del paquete), que serían como miembros privados.

En la unidad dedicada a la utilización de clases ya estudiaste los posibles modificadores de acceso que podía tener un miembro: **sin modificador** (acceso de paquete), **público**, **privado** o **protegido**. Aquí tienes de nuevo el resumen:

| Cuadro de niveles accesibilidad a los atributos de una clase |             |          |               |              |
|--|-------------|----------|---------------|--------------|
|  | Misma clase | Subclase | Mismo paquete | Otro paquete |
| Sin modificador (paquete)                                    | X           |          | X             | X            |
| public   | X           | X        | X             | X            |
| Private  | X           |          |               |              |
| Protected  | X           | X        | X             |              |

Si en el ejemplo anterior de la clase **Persona** se hubieran definido sus atributos como **private**:

```
public class Persona {  
    private String nombre;  
    private String apellidos;  
    ...  
}
```



Al definir la clase **Alumno** como heredera de **Persona**, no habrías tenido acceso a esos atributos, pudiendo ocasionar un grave problema de operatividad al intentar manipular esa información. Por tanto, en estos casos lo más recomendable habría sido declarar esos atributos como **protected** o bien sin modificador (para que también tengan acceso a ellos otras clases del mismo paquete, si es que se considera oportuno):

```
public class Persona {  
    protected String nombre;  
    protected String apellidos;  
    ...  
}
```

Sólo en aquellos casos en los que se desea explícitamente que un miembro de una clase no pueda ser accesible desde una clase derivada debería utilizarse el modificador **private**. En el resto de casos es recomendable utilizar **protected**, o bien no indicar modificador (acceso a

nivel de paquete).

### Ejercicio resuelto

Rescribe las clases **Alumno** y **Profesor** utilizando el modificador **protected** para sus atributos del mismo modo que se ha hecho para su superclase **Persona**

### Solución:

#### 1. Clase **Alumno**.

Se trata simplemente de añadir el modificador de acceso **protected** a los nuevos atributos que añade la clase.

```
public class Alumno extends Persona {  
    protected String grupo;  
    protected double notaMedia;  
    ...  
}
```

#### 2. Clase **Profesor**.

Exactamente igual que en la clase **Alumno**.

```
public class Profesor extends Persona {  
    protected String especialidad;  
    protected double salario;  
    ...  
}
```