

Práctica 2: Poner los atributos de la clase persona a private. Tratar de acceder desde la clase Alumno a esos atributos que antes heredaba. Toma captura de pantalla de los mensajes del IDE. Para ello, por ejemplo, puedes crear un método en alumno y llamar a this.nombre desde el método.

```
public class Persona {  
    private String nombre;  
    private String apellido;  
  
    public Persona(String nombre, String apellido) {  
        this.nombre = nombre;  
        this.apellido = apellido;  
    }  
}  
  
class Alumno extends Persona {  
    String curso;  
    double notaMedia;  
  
    public Alumno(String nombre, String apellido) {  
        super(nombre, apellido);  
    }  
  
    public String getNombre() {  
        return this.nombre;  
    }  
}
```

nombre has private access in Persona
(Alt-Enter shows hints)

En el caso de poner el atributo como privado en la clase Persona, nos encontraremos con un mensaje de error del IDE, que nos indica justamente eso.

```
public class Persona {  
    protected String nombre;  
    private String apellido;  
  
    public Persona(String nombre, String apellido) {  
        this.nombre = nombre;  
        this.apellido = apellido;  
    }  
}  
  
class Alumno extends Persona {  
    String curso;  
    double notaMedia;  
  
    public Alumno(String nombre, String apellido) {  
        super(nombre, apellido);  
    }  
  
    public String getNombre() {  
        return this.nombre;  
    }  
}
```

Si cambiamos la visibilidad a protected, public, o no especificamos (Esta última solo valida por estar en el mismo paquete.), no habrá ningún problema al llamar al atributo desde la clase Alumno.