

Diseño de clases en Java

Ricardo Pérez López

IES Doñana, curso 2020/2021

Generado el 31 de marzo de 2021 a las 14:07:00

Índice general

1. Definición de clases	1
1.1. Sintaxis básica	2
1.2. Visibilidad de una clase	2
1.2.1. Visibilidad predeterminada	2
1.2.2. Visibilidad pública	3
1.3. Visibilidad de un miembro de una clase	3
1.3.1. Pública	4
1.3.2. Privada	4
1.3.3. Predeterminada	4
2. Miembros de instancia	4
2.1. Campos	4
2.1.1. Acceso y modificación	4
2.1.2. Campos finales	4
2.2. Métodos	4
2.2.1. Referencia <code>this</code>	5
2.2.2. Constructores y destructores	5
2.2.3. Accesores y mutadores	5
2.2.4. Sobrecarga	5
2.2.5. Ámbito de un identificador	5
2.2.6. Resolución de identificadores	5
3. Miembros estáticos	5
3.1. Métodos estáticos	5
3.2. Campos estáticos	5
3.3. Campos estáticos finales	5

1. Definición de clases

1.1. Sintaxis básica

```
<clase> ::= [<modif_acceso_clase>] class <nombre> {  
    <miembro>*  
}  
  
<nombre> ::= identificador  
<miembro> ::= <variable> | <método>  
<variable> ::= [<modif_acceso_miembro>] [static] <decl_variable>  
<método> ::= [<modif_acceso_miembro>] [static] <decl_método>  
<modif_acceso_clase> ::= public  
<modif_acceso_miembro> ::= public | private | protected
```

La definición de una clase es una construcción sintáctica que define su propio ámbito y que está formada por un bloque de **declaraciones de miembros**, cada una de las cuales puede declarar una **variable** (también llamada **campo**) o un **método**.

A su vez, cada miembro puede ser **de instancia** o puede ser **estático**.

Ejemplo

```
public class Hola {  
    public int x = 4;  
    private String nombre;  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public void setNombre(String nom) {  
        nombre = nom;  
    }  
}
```

1.2. Visibilidad de una clase

Una clase siempre pertenece siempre a un paquete, que es el paquete en el que se ha definido.

El uso y definición de paquetes en Java lo estudiaremos con más profundidad posteriormente.

En relación a los paquetes en las que se definen, las clases pueden tener dos tipos de visibilidades:

- **Visibilidad *predeterminada* (por defecto o default)**: la clase sólo es accesible desde el interior del paquete en el que se ha definido.
- **Visibilidad *pública***: la clase es accesible desde cualquier paquete.

Para indicar la visibilidad que debe tener una clase, se puede usar un **modificador de acceso**.

1.2.1. Visibilidad predeterminada

Cuando no se utiliza ningún *modificador de acceso* al definir la clase, ésta se define con visibilidad predeterminada.

En un archivo fuente pueden definirse tantas clases con visibilidad predeterminada como se desee. Además, en ese caso el archivo fuente puede tener cualquier nombre (por supuesto, siempre con extensión `.java`).

1.2.2. Visibilidad pública

Para definir una clase con visibilidad pública, se usa el *modificador de acceso* **public** en la definición de la clase.

En un archivo fuente pueden definirse muchas clases, pero sólo una de ellas puede ser pública.

Además, el archivo fuente debe llamarse igual que la (única) clase pública que contiene.

1.3. Visibilidad de un miembro de una clase

Cada miembro de una clase puede tener uno de estos cuatro tipos de visibilidades:

- **Visibilidad *privada*:** el miembro sólo es accesible desde el interior de la clase en la que se ha definido.
- **Visibilidad *predeterminada (por defecto o default)*:** el miembro es accesible desde el interior de la clase en la que se ha definido y también desde otras clases que pertenezcan al mismo *paquete*.
- **Visibilidad *protegida*:** el miembro es accesible desde el interior de la clase en la que se ha definido, también desde otras clases que pertenezcan al mismo paquete y también desde sus subclases (aunque se hayan definido en paquetes distintos).
- **Visibilidad *pública*:** el miembro es accesible desde el interior de la clase en la que se ha definido y también desde cualquier otra clase (siempre que la clase en sí también sea accesible).

La visibilidad es un mecanismo de **encapsulación** que impide que ciertos miembros puedan ser accedidos (o incluso conocidos) fuera de la clase en la que se han definido, o fuera del paquete que contiene la clase en la que se ha definido.

El siguiente cuadro resume las cuatro visibilidades y desde dónde se puede acceder a un miembro definido con una determinada visibilidad en una determinada clase:

Visibilidad	La propia clase	Otras clases del mismo paquete	Subclases de la clase	Otras clases de cualquier paquete
Privada	Sí	No	No	No
Predeterminada	Sí	Sí	No	No
Protegida	Sí	Sí	Sí	No
Pública	Sí	Sí	Sí	Sí

La visibilidad que queremos que tenga un determinado miembro se indica mediante los llamados **modificadores de acceso**.

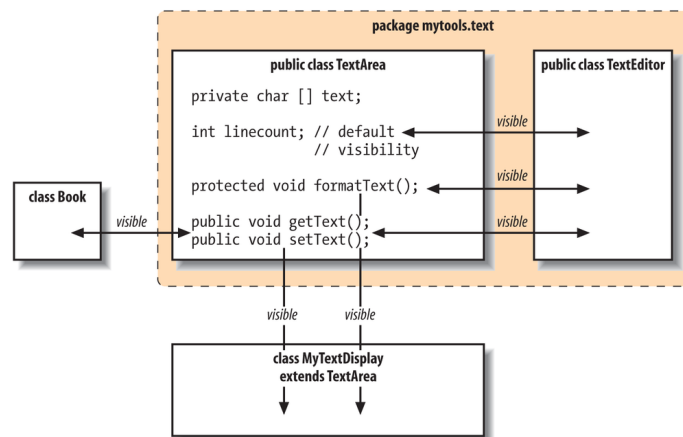
Los **modificadores de acceso** son palabras clave que acompañan a la declaración de un miembro y

que sirven para indicar la visibilidad deseada para ese miembro.

La forma de indicar que se desea que un miembro tenga visibilidad predeterminada es no usar ningún modificador de acceso en su declaración.

Para el resto de visibilidades existe un **modificador de acceso** que puede ir acompañando a la **declaración de cada miembro**:

Visibilidad	Modificador de acceso
Pública	public
Privada	private
Protegida	protected
Predeterminada	(ninguno)



Visibilidades en Java

1.3.1. Pública

1.3.2. Privada

1.3.3. Predeterminada

2. Miembros de instancia

2.1. Campos

2.1.1. Acceso y modificación

2.1.2. Campos finales

2.2. Métodos

2.2.1. Referencia `this`**2.2.2. Constructores y destructores****2.2.3. Accesores y mutadores****2.2.4. Sobrecarga****2.2.5. Ámbito de un identificador****2.2.6. Resolución de identificadores****3. Miembros estáticos****3.1. Métodos estáticos****3.2. Campos estáticos****3.3. Campos estáticos finales**