5.A. Introducción a las clases.

2. Estructura y miembros de una clase.

2.2. Cabecera de una clase.

En general, la declaración de una clase puede incluir los siguientes elementos y en el siguiente orden:

Modificadores tales como `public`, `abstract` o `final`.

El nombre de la clase (con la primera letra de cada palabra en mayúsculas, por convenio).

El nombre de su clase padre (superclase), si es que se especifica, precedido por la palabra reservada `extends` ("extiende" o "hereda de").

Una lista separada por comas de interfaces que son implementadas por la clase, precedida por la palabra reservada `implements` ("implementa").

El cuerpo de la clase, encerrado entre llaves {}.

La sintaxis completa de una cabecera (los cuatro primeros puntos) queda de la forma:

```
[modificadores]
```

```
class <NombreClase> [extends <NombreSuperClase>][implements
```

```
<NombreInterface1>] [[implements <NombreInterface2>] ...] {
```

En el ejemplo anterior de la clase `Punto` teníamos la siguiente cabecera:

```
class Punto {
```

En este caso no hay modificadores, ni indicadores de herencia, ni implementación de interfaces. Tan solo la palabra reservada `class` y el nombre de la clase. Es lo mínimo que puede haber en la cabecera de una clase.

La herencia y las interfaces las verás más adelante. Vamos a ver ahora cuáles son los modificadores que se pueden indicar al crear la clase y qué efectos tienen. Los modificadores de clase son:

```
[public] [final | abstract]
```

Veamos qué significado tiene cada uno de ellos:

- Modificador `public`. Indica que la clase es visible (se pueden crear objetos de esa clase) desde cualquier otra clase. Es decir, desde cualquier otra parte del programa. Si no se especifica este modificador, la clase sólo podrá ser utilizada desde clases que estén en el mismo paquete. El concepto de paquete lo veremos más adelante. Sólo puede haber una clase `public` (clase principal) en un archivo .java. El resto de clases que se definan en ese archivo no serán públicas.

- Modificador `abstract`. Indica que la clase es abstracta. Una clase abstracta no es instanciable. Es decir, no es posible crear objetos de esa clase (habrá que utilizar clases que hereden de ella). En este momento es posible que te parezca que no tenga sentido que esto pueda suceder (si no puedes crear objetos de esa clase, ¿para qué la quieres?), pero puede resultar útil a la hora de crear una jerarquía de clases. Esto lo verás también más adelante al estudiar el concepto de herencia.

- Modificador `final`. Indica que no podrás crear clases que hereden de ella. También volverás a este modificador cuando estudies el concepto de herencia. Los modificadores `final` y `abstract` son excluyentes (sólo se puede utilizar uno de ellos).

Todos estos modificadores y palabras reservadas las iremos viendo poco a poco, así que no te preocupes demasiado por intentar entender todas ellas en este momento.

En el ejemplo anterior de la clase `Punto` tendríamos una clase que sería sólo visible (utilizable) desde el mismo paquete en el que se encuentra la clase (modificador de acceso por omisión o de paquete, o `package`). Desde fuera de ese paquete no sería visible o accesible. Para poder utilizarla desde cualquier parte del código del programa bastaría con añadir el atributo public: `public class Punto`.

Autoevaluación

Si queremos poder instanciar objetos de una clase desde cualquier parte de un programa, ¿qué modificador o modificadores habrá que utilizar en su declaración?

- private.
- public.
- abstract.
- Ninguno de los anteriores.