5.D. Encapsulación, control de acceso y visibilidad.

1. Encapsulación, control de acceso y visibilidad.

Dentro de la Programación **Orientada a Objetos** ya has visto que es muy importante el concepto de **ocultación**, la cual ha sido lograda gracias a la **encapsulación** de la información dentro de las clases. De esta manera una clase puede ocultar parte de su contenido o restringir el acceso a él para evitar que sea manipulado de manera inadecuada. Los **modificadores de acceso** en Java permiten especificar el **ámbito de visibilidad** de los miembros de una clase, proporcionando así un mecanismo de accesibilidad a varios niveles.

Acabas de estudiar que cuando se definen los miembros de una clase (atributos o métodos), e incluso la propia clase, se indica (aunque sea por omisión) un modificador de acceso. En función de la visibilidad que se desee que tengan los objetos o los miembros de esos objetos se elegirá alguno de los modificadores de acceso que has estudiado. Ahora que ya sabes cómo escribir una clase completa (declaración de la clase, declaración de sus atributos y declaración de sus métodos), vamos a hacer un repaso general de las opciones de **visibilidad** (**control de acceso**) que has estudiado.

Los modificadores de acceso determinan si una clase puede utilizar determinados miembros (acceder a atributos o invocar miembros) de otra clase. Existen dos niveles de control de acceso:

1. **A nivel general (nivel de clase)**: visibilidad de la propia clase.

2. **A nivel de miembros**: especificación, miembro por miembro, de su nivel de visibilidad.

En el caso de la clase, ya estudiaste que los niveles de visibilidad podían ser:

- **Público** (modificador public), en cuyo caso la clase era visible a cualquier otra clase (cualquier otro fragmento de código del programa).

- **Privada al paquete** (sin modificador o modificador "por omisión"). En este caso, la clase sólo será visible a las demás clases del mismo paquete, pero no al resto del código del programa (otros paquetes).

En el caso de los miembros, disponías de otras dos posibilidades más de niveles de accesibilidad, teniendo un total de cuatro opciones a la hora de definir el control de acceso al miembro:

- **Público** (modificador `public`), igual que en el caso global de la clase y con el mismo significado (miembro visible desde cualquier parte del código).

- **Privado al paquete** (sin modificador), también con el mismo significado que en el caso de la clase (miembro visible sólo desde clases del mismo paquete, ni siquiera será visible desde una subclase salvo si ésta está en el mismo paquete).

- **Privado** (modificador `private`), donde sólo la propia clase tiene acceso al miembro.

- **Protegido** (modificador `protected`)

Para saber más

Puedes echar un vistazo al artículo sobre el control de acceso a los miembros de una clase Java en los manuales de Oracle (en inglés):

Controlling Access to Members of a Class.

Autoevaluación

Si queremos que un atributo de una clase sea accesible solamente desde el código de la propia clase o de aquellas clases que hereden de ella, ¿qué modificador de acceso deberíamos utilizar?

- ○ private.
- ○ protected.
- ○ public.
- ○ Ninguno de los anteriores.