

7.E. Interfaces.

1. Interfaces.

1.3. Definición de interfaces.

La declaración de una interfaz en Java es similar a la declaración de una clase, aunque con algunas variaciones:

- Se utiliza la palabra reservada `interface` en lugar de `class`.
- Puede utilizarse el modificador `public`. Si incluye este modificador la interfaz debe tener el mismo nombre que el archivo `.java` en el que se encuentra (exactamente igual que sucedía con las clases). Si no se indica el modificador `public`, el acceso será por omisión o "de paquete" (como sucedía con las clases).
- Todos los miembros de la interfaz (atributos y métodos) son `public` de manera implícita. No es necesario indicar el modificador `public`, aunque puede hacerse.
- Todos los atributos son de tipo `final` y `public` (tampoco es necesario especificarlo), es decir, constantes y públicos. Hay que darles un valor inicial.
- Todos los métodos son abstractos también de manera implícita (tampoco hay que indicarlo). No tienen cuerpo, tan solo la cabecera.

Como puedes observar, una interfaz consiste esencialmente en una lista de atributos finales (constantes) y métodos abstractos (sin implementar). Su sintaxis quedaría entonces:

```
[public] interface <NombreInterfaz> {  
  
    [public] [final] <tipo1> <atributo1>= <valor1>;  
  
    [public] [final] <tipo2> <atributo2>= <valor2>;  
  
    ...  
  
    [public] [abstract] <tipo_devuelto1> <nombreMetodo1> ([lista_parámetros]);  
  
    [public] [abstract] <tipo_devuelto2> <nombreMetodo2> ([lista_parámetros]);  
  
    ...  
  
}
```

Si te fijas, la declaración de los métodos termina en punto y coma, pues no tienen cuerpo, al igual que sucede con los métodos abstractos de las clases abstractas.

El ejemplo de la interfaz `Depredador` que hemos visto antes podría quedar entonces así:

```
public interface Depredador {  
  
    void localizar (Animal presa);  
  
    void cazar (Animal presa);  
  
    ...  
  
}
```

Serán las clases que implementen esta interfaz (León, Leopardo, Cocodrilo, Rana, Lagarto, Hombre, etc.) las que definan cada uno de los métodos por dentro.

Autoevaluación

Los métodos de una interfaz en Java tienen que ser obligatoriamente declarados como `public` y `abstract`. Si no se indica así, se producirá un error de compilación. ¿Verdadero o Falso?

- ☐ Verdadero
☐ Falso

Ejercicio resuelto

Crea una interfaz en Java cuyo nombre sea **Imprimible** y que contenga algunos métodos útiles para mostrar el contenido de una clase:

1. Método **devolverContenidoString**, que crea un **String** con una representación de todo el contenido público (o que se decida que deba ser mostrado) del objeto y lo devuelve. El formato será una lista de pares "nombre=valor" de cada atributo separado por comas y la lista completa encerrada entre llaves: "{<nombre_atributo_1>=<valor_atributo_1>, ..., <nombre_atributo_n>=<valor_atributo_n>}".
2. Método **devolverContenidoArrayList**, que crea un **ArrayList** de **String** con una representación de todo el contenido público (o que se decida que deba ser mostrado) del objeto y lo devuelve.
3. Método **devolverContenidoHashtable**, similar al anterior, pero en lugar devolver en un **ArrayList** los valores de los atributos, se devuelve en una **Hashtable** en forma de pares (nombre, valor).

Solución:

Se trata simplemente de declarar la interfaz e incluir en su interior esos tres métodos:

```
public interface Imprimible {  
  
    String devolverContenidoString ();  
  
    ArrayList devolverContenidoArrayList ();  
  
    Hashtable devolverContenidoHashtable ();  
  
}
```

El cómo se implementarán cada uno de esos métodos dependerá exclusivamente de cada clase que decida implementar esta interfaz.