

7.E. Interfaces.

1. Interfaces.

1.6. Herencia de interfaces.

Las **interfaces**, al igual que las **clases**, también permiten la **herencia**. Para indicar que una **interfaz** hereda de otra se indica nuevamente con la palabra reservada **extends**. Pero en este caso sí se permite la **herencia múltiple de interfaces**. Si se hereda de más de una **interfaz** se indica con la lista de **interfaces** separadas por comas.

Por ejemplo, dadas las interfaces **InterfazUno** e **InterfazDos**:

```
public interface InterfazUno {  
  
    // Métodos y constantes de la interfaz Uno  
  
}
```

```
public interface InterfazDos {  
  
    // Métodos y constantes de la interfaz Dos  
  
}
```

Podría definirse una nueva **interfaz** que heredara de ambas:

```
public interface InterfazCompleja extends InterfazUno, InterfazDos {  
  
    // Métodos y constantes de la interfaz compleja  
  
}
```

Autoevaluación

En Java no está permitida la herencia múltiple ni para clases ni para interfaces. ¿Verdadero o Falso?

- ☐ Verdadero
☐ Falso

Ejercicio resuelto

Localiza en la API de Java algún ejemplo de interfaz que herede de una o varias interfaces (puedes consultar la documentación de referencia de la API de Java).

Solución:

Existen una gran cantidad de **interfaces** en la API de Java que heredan de otras **interfaces**. Aquí tienes un par de ejemplos:

- La interfaz `java.awt.event.ActionListener`, que hereda de `java.util.EventListener`:

```
§ public interface ActionListener extends EventListener
```

- La interfaz `org.omg.CORBA.Policy`, que hereda de `org.omg.CORBA.PolicyOperations`, `org.omg.CORBA.Object` y `org.omg.CORBA.portable.IDLEntity`:

```
§ public interface Policy extends PolicyOperations, Object, IDLEntity.
```

