**Interfaces en evolución**

Considere una interfaz que ha desarrollado llamada DoIt:

public interface DoIt {

void doSomething(int i, double x);

int doSomethingElse(String s);

}

Suponga que, en un momento posterior, desea agregar un tercer método DoIt, de modo que la interfaz ahora se convierta en:

public interface DoIt {

void doSomething(int i, double x);

int doSomethingElse(String s);

boolean didItWork(int i, double x, String s);

}

Si realiza este cambio, todas las clases que implementan la DoItinterfaz anterior se interrumpirán porque ya no implementarán la interfaz anterior. Los programadores que confían en esta interfaz protestarán en voz alta.

Intente anticipar todos los usos de su interfaz y especifíquelo completamente desde el principio. Si desea agregar métodos adicionales a una interfaz, tiene varias opciones. Podría crear una DoItPlus interfaz que se extienda DoIt:

public interface DoItPlus extends DoIt {

boolean didItWork(int i, double x, String s);

}

Ahora los usuarios de su código pueden elegir continuar usando la interfaz anterior o actualizar a la nueva interfaz.

Alternativamente, puede definir sus nuevos métodos como métodos predeterminados. El siguiente ejemplo define un método predeterminado llamado didItWork:

public interface DoIt {

void doSomething(int i, double x);

int doSomethingElse(String s);

**default boolean didItWork(int i, double x, String s) {**

**// Method body**

**}**

}

Tenga en cuenta que debe proporcionar una implementación para los métodos predeterminados. También podría definir nuevos métodos estáticos para las interfaces existentes. Los usuarios que tienen clases que implementan interfaces mejoradas con nuevos métodos predeterminados o estáticos no tienen que modificarlos o recompilarlos para acomodar los métodos adicionales.