

## Use Interface Where Possible

**Use Interface Where Possible** refactoring delegates execution of the specified methods, derived from a base class/interface, to an instance of an ancestor class or an inner class, implementing the same interface.

### Example

Before	After
<pre data-bbox="177 412 983 672">// File Class.java public class Class implements Interface { public void publicMethod() {     ... } public void hiddenMethod() {     ... } }</pre>	<pre data-bbox="983 412 1519 672">// File Class.java UNCI public class Class impl public void publicMethod() {     ... } public void hiddenMethod() {     ... } }</pre>
<pre data-bbox="177 672 983 857">// File Interface.java public interface Interface { int     CONSTANT=0; void publicMethod(); }</pre>	<pre data-bbox="983 672 1519 857">// File Interface.java public interface Interfa int     CONSTANT=0; void publicMethod(); }</pre>
<pre data-bbox="177 857 983 1258">// File AnotherClass.java public class AnotherClass { Class firstClass; Class secondClass; public void method() {     firstClass.publicMethod();     firstClass.hiddenMethod();     secondClass.publicMethod();     if (secondClass         instanceof Class) {         ...     }     ... } }</pre>	<pre data-bbox="983 857 1519 1258">// File AnotherClass.j public class AnotherCla Class firstClass; Interface secondInterface; public void method() {     firstClass.publicMethod();     firstClass.hiddenMethod();     secondInterface.publicMethod();     if (secondInterface         instanceof Interface) {         ...     }     ... } }</pre>

## To use interface where possible

1. Select a class whose methods should be delegated to its parent class or interface. To do that, place the caret on this class in the editor or in one of the views (Project, Commander).
2. On the main menu or on the context menu of the selection, choose **Refactor | Use Interface Where Possible**.
3. In the [Use Interface Where Possible](#) dialog box, select the parent object, which will replace the usages of the current class.
4. To replace the current class name in `instanceof` statements, check the option **Use interface/superclass in instanceof**.

If you use `instanceof` statements and leave this check box unselected, you may receive erroneous code, such as:

```
if (secondInterface instanceof Class)
```

This code will compile, but may produce undesired results.

5. To review the intended changes and make final corrections prior to the refactoring, click **Preview**. To continue without preview, click **Refactor**.
6. The **Rename variables** dialog appears. It lists the occurrences of the class, that may be replaced by the superclass or the interface selected. Select the usages you want to replace, and (optionally) specify new names for each of them.
7. Click **OK** to continue. If you have previously clicked the **Preview** button, the **Preview** window appears now.

## See Also

### Reference:

- [Use Interface Where Possible Dialog](#)

### Web Resources:

- [Developer Community](#) 