Summary 9 – Interfaces and the implements keyword

- What is this?

If class B implements interface X, then class B must implement (i.e., have a definition of) all the methods specified in the interface X.

For example, consider the StringTransformable interface that is part of the WordGames project:

```
public interface StringTransformable {
   public String transform(String stringToTransform);
}
```

Each class in WordGames implements this interface, that is, each defines the transform method as specified in the StringTransformable interface. For example:

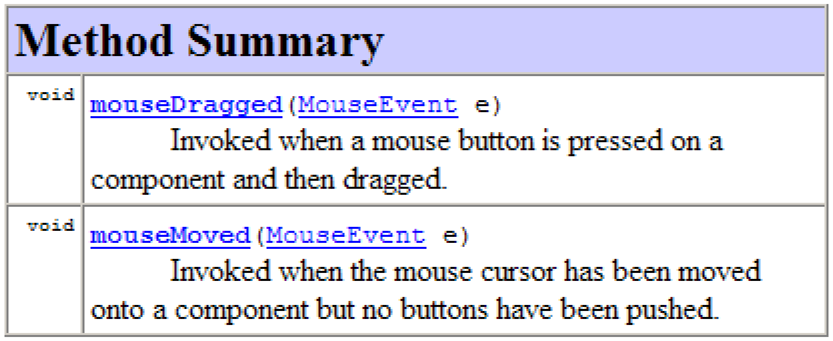
```
public class Shouter implements StringTransformable {

      @Override
      public String transform(String stringToTransform) {
            return stringToTransform.toUpperCase();
      }
}
```

Specifying an interface allows others to interface their code with yours. For example, the authors of the Swing package provided the MouseMotionListener interface (per the example to the right) so that you would know what methods their code will call when the mouse is dragged.

- Example

The MouseMotionListener interface is:



Method Summary	
void	mouseDragged(MouseEvent e) Invoked when a mouse button is pressed on a component and then dragged.
void	mouseMoved(MouseEvent e) Invoked when the mouse cursor has been moved onto a component but no buttons have been pushed.

So if the Eye class implements MouseMotionListener:

```
public class Eye extends JPanel
                 implements MouseMotionListener {
```

then the Eye class must implement the mouseDragged and mouseMoved methods specified in the MouseMotionListener interface:

```
public void mouseMoved(MouseEvent event) {
    Point mousePoint = new Point(event.getX(),
                                 event.getY());
    this.look(mousePoint);
}

public void mouseDragged(MouseEvent event) {
    // Do nothing
}
```

- For further study:
  - *Big Java*, section 9.1 *Using Interfaces for Code Reuse*
  - This summary's *author:* David Mutchler