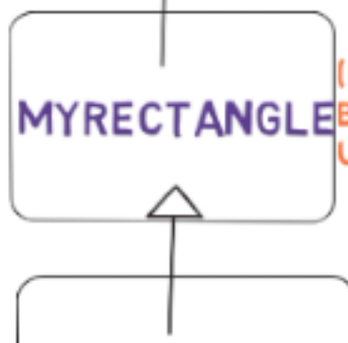
```
public interface IShape {

    public String introduceYourself();
```

(INTERFACE)

```
public String introduceYourself() {
    return "I am a rectangle";
}
```



(BASE CLASS, BUT NOT AN ABSTRACT BASE CLASS, BECAUSE IT HAS NO UNIMPLEMENTED FUNCTIONS)

```
public String introduceYourself() {
    return "I am a square";
}
```

(DERIVED CLASS)

ALL GOOD SO FAR..

BUT NOW SAY YOU DECIDE THAT THE ISHAPE INTERFACE NEEDS A NEW METHOD **GETCOLOR**

IT WOULD SEEM LIKE WE HAVE 2 OPTIONS, NEITHER OF WHICH IS GREAT

OPTION #1: ADD A GETCOLOR METHOD TO THE INTERFACE ISHAPE AND ADD IMPLEMENTATIONS TO ALL EXISTING CLASSES THAT IMPLEMENT ISHAPE

OPTION #2: CREATE A NEW INTERFACE, ISHAPE2, WHICH HAS THE GETCOLOR METHOD, AND HAVE ISHAPE2 EXTEND ISHAPE. EXISTING CLASSES WOULD BE UNCHANGED, BUT NEW IMPLEMENTATIONS WOULD BE OF ISHAPE2, AND NOT OF ISHAPE

BREAKING BACKWARD COMPATIBILITY IS A TERRIBLE IDEA, ALWAYS

OPTION 1 SUCKS, BECAUSE ALL OF THE CLASSES THAT IMPLEMENTED ISHAPE, SUCH AS MYSQUARE, MYCIRCLE,... WILL NOW STOP WORKING (UNLESS THEY EACH GET THEIR OWN IMPLEMENTATION OF THE GETCOLOR METHOD)

THESE 2 INTERFACES DRIVE ERRATIC BEHAVIOUR

OPTION 2 SUCKS, BECAUSE SOMES SHAPES WOULD IMPLEMENT ISHAPE (AND NOT SUPPORT THE GETCOLOR FUNCTION), WHILE SOME OTHER SHAPES WOULD ARBITRARILY SUPPORT THIS FUNCTION

DEFAULT INTERFACE METHODS TO THE RESCUE!

JAVA NOW HAS A WAY FOR INTERFACES TO CONTAIN A DEFAULT IMPLEMENTATION OF METHODS

THIS MIGHT SEEM COUNTER-INTUITIVE AT FIRST, BECAUSE WE TYPICALLY THINK OF INTERFACES AS BEING METHOD SIGNATURES WITH NO METHOD IMPLEMENTATIONS..

..BUT IN SITUATIONS WHERE THE PROGRAMMER WOULD LIKE TO ADD TO EXISTING INTERFACES AND STILL KEEP BACKWARD COMPATIBILITY, THIS NEW FEATURE IS A GODSEND

USING DEFAULT METHODS IS INCREDIBLY SIMPLE, JUST MARK THE METHOD WITH THE DEFAULT KEYWORD

NEW AND PRE-EXISTING ISHAPE CLASSES WILL ALL SATISFY THE SAME INTERFACE AND BEHAVE UNIFORMLY

```
public interface IShape {

    public double getArea();
    public double getPerimeter();

    default String getColor() {
        return "Color.Red";
    }

}
```

NONE OF THE CLASSES IMPLEMENTING ISHAPE NEED TO CHANGE, UNLESS THEY WISH TO