

## Summary 7 - The *this* keyword

- What is this?

Non-static methods are always applied to an object. We call that object the *implicit parameter*. For example:

```
String father = "Frank Zappa";
String son = "Moon Unit Zappa";
father.replace('a', '8');
son.replace('o', '6');
```

Here, *father* is the implicit parameter in the first call to the *replace* method and *son* is the implicit parameter in the second call.

Consider the code for *replace* in the definition of the String class:

```
public class String {
    ...
    public String replace(char oldChar, char newChar) {
        blah, blah, blah
    }
}
```

Where I have written *blah, blah, blah*, there is presumably a loop that goes through THIS String (i.e., the one for which we are replacing all occurrences of oldChar with newChar). That is, there has to be some way to refer to the implicit parameter. That's what the *this* keyword is!

To repeat: inside a (necessarily non-static) method in a class definition, the keyword *this* refers to the implicit parameter upon which method is acting.

Similarly, inside a constructor, the keyword *this* refers to the object being constructed.

As a matter of style, in our class definitions, we write *this.foo* to refer to the field foo and *this.blah()* to call non-static methods of the class. Doing so has two advantages: it reinforces the fact that foo is THIS object's field and blah is applied to THIS implicit parameter, and it allows our IDE Eclipse to show us the possible fields/methods that THIS object can act upon.

- Example (from JavaEyes)

```
public StartButton(String title, AnimatedPanel panel) {
    super(title);

    this.squarePanel = panel;
    this.hasBeenPressed = false;

    this.setBackground(StartButton.DEFAULT_COLOR);
    this.addActionListener(this);
}
```

By writing *this.squarePanel* we can easily tell that *squarePanel* is a field ("this" StartButton's square panel).

By writing *this.setBackground(...)* we can easily tell that it is THIS StartButton's background whose color we are setting.

Test your understanding: Do you understand the last use of the *this* keyword in the above example?

- For further study:
  - *Big Java*, section 3.8 *Implicit and Explicit Method Parameters*
  - This summary's *author*: David Mutchler
  - See also the Summary on:
    - *StaticMethodsAndFields*