Duplicate method

**Name:** Duplicate method

**Description:** There is another identical implementation of this method. Extract the code to a common superclass or delegate to improve sharing.

**ID:** cs/duplicate-method

**Kind:** problem

**Severity:** recommendation

**Precision:** high

Methods should not be duplicated at more than one place in the program. Duplicating code makes it harder to update should a change need to be made. It also makes the code harder to read.

**Recommendation**

Determining how to address this issue requires some consideration. If the duplicate methods are in the same class then it is normally possible to just remove one and replace all references to that method by references to the other method. If the methods are in different classes then there might be a need to create a superclass that contains the method, which both classes inherit. If it is not logical to create a superclass the method could be moved into a separate utility class.

**Example**

In this example the Toolbox and the Window class both have the same move method. In this case it would be logical to put this method as well as the x and y properties into a new superclass that Toolbox and Window extend.

```csharp
1 class Toolbox
2 {
3     private int x;
4     private int y;
5     public void move(int x, int y)
6     {
7         this.x = x;
8         this.y = y;
9     }
10     // ...
11 }
12 class Window
13 {
14     private int x;
15     private int y;
16     public void move(int x, int y)
17     {
18         this.x = x;
19         this.y = y;
20     }
21     // ...
22 }
```

**Fixing Using a Superclass**

The example could be easily fixed by moving the x and y properties as well as the move method to a parent class. Note that the x and y properties have to be changed to protected if they are accessed from the Toolbox and Window classes.
class Container {
    protected int x;
    protected int y;
    public void move(int x, int y) {
        this.x = x;
        this.y = y;
    }
}

class Toolbox : Container {
    // ...
}

class Window : Container {
    // ...
}

References