CS 580 Client-Server Programming
Fall Semester, 2000
Doc 11 Testing Hidden Methods

Contents
Testing and Hidden Methods/State ........................................... 3
How to Test Hidden Methods Directly? ...................................... 5
   Method 1: Relax the protection level ..................................... 5
   Method 2: Use inner classes ............................................. 6
   Method 3: Use reflection ............................................. 7

References

XP mailing list

Inner Class lecture notes at: http://www.eli.sdsu.edu/courses/fall98/cs596/notes/nested/nested.html

Copyright ©, All rights reserved.
2000 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA.
OpenContent (http://www.opencontent.org/opl.shtml) license defines the copyright on this document.
How NASA Navigates Space Craft

1. Send the craft in the general direction of the target
2. As you get closer to the target, adjust the craft's direction
3. Repeat 2 until you reach the target

The arguments on the slide 4 are some course corrections that you may need to consider later on.
Testing and Hidden Methods/State

Issues:

• How does one test hidden methods?

• Direct access to an object's state can reduce the time needed to write a test
Testing Hidden Methods One Position
Don't Do it

Pro:

- Can not test everything
- Clients of an object only care if public interface works correctly
- Testing public interface also tests hidden methods
- Hidden methods are more likely to change, requiring changes to the tests

The basic idea is to work smarter not harder. One can not completely test each class, and one does not have infinite time to write tests one should write the most effective tests possible. Tests of the public interface of class will also test hidden methods of the class. Bugs in hidden methods that never affect the public methods are not a problem. Since the most important thing is that the public interface works correctly, concentrate your tests on the public interface.

Con:

- The closer the test is to the code it tests the easier the test
- Bugs in hidden methods can make it hard to debug public methods

My experience is that the more code I write without tests, the more time I spend on finding and correcting bugs. How many times have you spent hours (days?) tracking down a bug that turned out to be a simple bug in some simple untested method, which would have been easy to test? The argument that one can not test everything and must make effective use of ones testing time is correct. Given the differences in programmer skill level, programmer experience, etc., everyone has to work out their own solution to this. The XP solution is to try to test everything that could possibly break. Since most students are not used to testing, you have to fight the habit of not testing and testing after you have completely finished coding. Given the current state of affairs in commercial software, the industry has a lot to learn about testing.
How to Test Hidden Methods Directly?
Method 1: Relax the protection level

In Java one can

- Make the method package level access
- Place the test class in the same package as the tested code

Pro:

- Makes it possible to test the hidden method
- Clients outside the package can not access the method

Con:

- Clients in the package may then use the method
- Requires organizational discipline to avoid using the method

You should comment the method to inform the clients that the method is not to be used
How to Test Hidden Methods Directly?
Method 2: Use inner classes

import junit.framework.TestCase;

public class Foo {
    private int value;

    private void bar() {
        value = 10;
    }

    public static class FooTest extends TestCase {
        public FooTest(String name) {
            super(name);
        }

        public void testBar() {
            Foo a = new Foo();
            a.bar();
            assert( 10 == a.value );
        }
    }
}

Pro:
• Provides access to all methods/fields
• Test does not have to be shipped with production code
• Test stays with tested class

Con:
• Source files are bit harder to read
• Must remove all inner $class files from production code
• Test not with other test classes
How to Test Hidden Methods Directly?
Method 3: Use reflection

Pro:
• Java reflection provides access to all methods/fields of a class
• Does not require any changes to tested class

Con:
• Reflection can be slow
• Reflection is cumbersome to use
• Requires setting permission files

See http://www.eli.sdsu.edu/courses/fall98/cs596/notes/reflection/reflection.html for more information about reflection