

CS 683 Emerging Technologies: Embracing Change
Spring Semester, 2001
Doc 15 Code Smells
Contents

Some Testing	2
A Short Catalog of Test Ideas	4
Any Object.....	5
Strings	6
Numbers.....	7
Collections.....	8
Linked Structures (trees, graphs, etc.).....	9
Equality Testing of Objects.....	10

References

Refactoring: Improving the Design of Existing Code, Fowler, Addison-Wesley, 1999

Testing for Programmers: A tutorial for OOPSLA 2000, Brian Marick,
<http://www.testing.com/writings/half-day-programmer.pdf>

Used here with permission from Brian Marick

A Short Catalog of Test Ideas, Brian Marick,
<http://www.testing.com/writings/short-catalog.pdf>

Used here with permission from Brian Marick

Copyright©, All rights reserved. 2001 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

Some Testing Fowler on Testing¹

"Make sure all tests are fully automatic and that they check their own results"

"It is better to write and run incomplete tests than not to run complete tests"

"Don't let the fear that testing can't catch all bugs stop you from writing the tests that will catch most bugs"

"Trying to write too many tests usually leads to not writing enough"

"Run your tests frequently"

"When you get a bug report, start by writing a unit test that exposes the bug"

Think of the boundary conditions and concentrate your tests there

¹ Fowler Chapter 4, pp. 89-102

Programming Errors

Programmers tend to make the same errors many times

Keep a list or catalog of your errors

A Short Catalog of Test Ideas

Tests develop catalogs of commonly found errors in programs

Since errors are often repeated, this helps testers find common errors

As programmers such a catalog:

- Suggests tests to uncover errors
- Help avoid errors when writing code

If we know these are common errors, we can keep them in mind while coding

The following catalog is from Brian Marick

<http://www.testing.com/writings/short-catalog.pdf>

The catalog is used here with permission

Any Object

Test nil(null) references and pointers to objects

In Java/Smalltalk

Does the code handle correctly variables & parameters that are null(nil)

Java

```
String firstName = null
```

Smalltalk

```
| firstName |  
firstName := nil.
```

Strings

Test the empty string

Does the code do the correct thing when string variables/parameters are the empty string

In Java/Smalltalk an empty string is not the same as a null(nil) reference to a string

Java

```
String firstName = "";  
String secondName = new String();
```

Smalltalk

```
| firstName secondName |  
firstName := ''.   
secondName := String new
```

Numbers

Test the code using:

- 0
- The smallest number

Often numbers are used in a context with a valid range

The smallest number refers to the smallest valid number in the range

- Just below the smallest number
- The largest number
- Just above the largest number

Example

`int planetIndex;`//Represents the I'th planet from the Sun

Numbers to test

0	Below the smallest
1	Smallest
9	Largest (Pluto is still considered a planet)
10	Above the largest

Collections

Test the code using:

- An empty collection
- A collection with one element
- The largest possible collection

Not the largest possible collection allowed by the language/hardware

The largest possible collection the system will encounter

If this is not possible use a collection with more than one element

- A collection with duplicate elements

Linked Structures (trees, graphs, etc.)

Test the code using:

- An empty structure
- Minimal non-empty structure
- A circular structure
- A structure with depth greater than one

The test must make the code reach the lowest depth

If the structure in the context has a maximally deep use that level

Equality Testing of Objects

Objects have two meanings of equality

- Pointer Identical
Two object references point to the same memory location
- Equal
The fields of the two objects have the same value

Java

- ==
Tests if two object references are pointer identical
- equals()
Tests if two objects are equal

If this method is not implemented in a class it defaults to ==

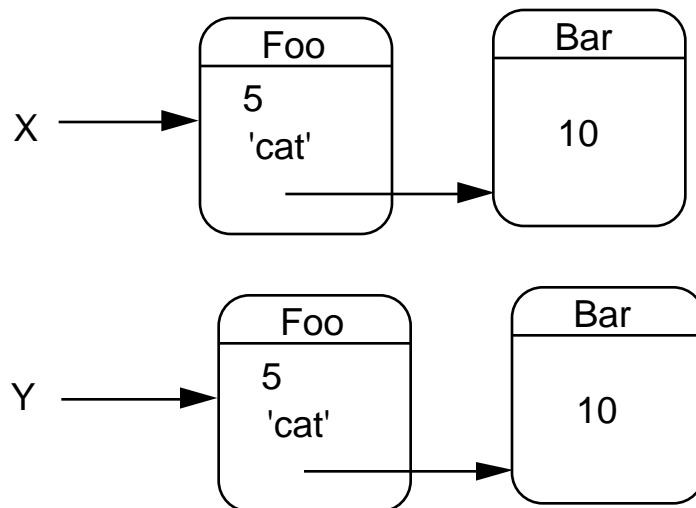
Smalltalk

- ==
Tests if two object references are pointer identical
- =
Tests if two objects are equal

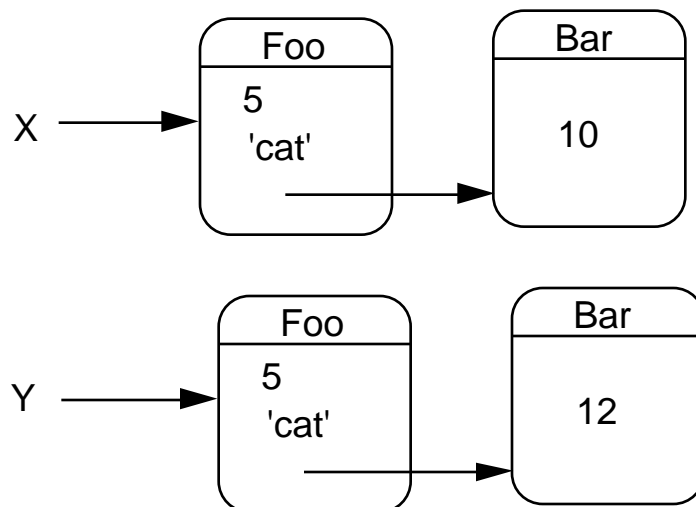
If this method is not implemented in a class it defaults to ==

Test the code with objects equal but not identical

Lack of pointer identity should extend as far down as is meaningful to the code



Test the code with objects different at the lowest level



"Trying to write too many tests usually leads to not writing enough ... You get many benefits from testing even if you do a little testing ..."

Fowler