



Cobertura de código

David Alonso Ríos

Facultad de Informática
Universidade da Coruña

2008/2009

Motivación

- Cobertura: El grado en que el código fuente de un programa ha sido testeado.
- La calidad de un test no está en el número de líneas que lo compone ni en el número de tests ejecutados, sino en su porcentaje de cobertura.
- Como resultado:
 - Podemos estar seguros de que hemos probado partes importantes del sistema.
 - También somos informados de qué partes no estamos cubriendo.

Tipos de cobertura

- De líneas.
- De ramas: p. ej., ¿se han probado los dos casos de un if? (generalmente más importante que el de líneas)
- De funciones: ¿se ha ejecutado cada función?
- De condiciones: evaluar subexpresiones booleanas a cierto y falso (no necesariamente en ramas)
- ...

HAC LUCE Ejemplo

- Tenemos una función que nos dice si un año es bisiesto:
 - Un año es bisiesto si es divisible entre 4.
 - Sin embargo, los años divisibles entre 100 no son bisiestos.
 - Pero si el año es divisible entre 400 sí que es bisiesto.
- Probamos dos conjuntos de tests:
 - 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008.
 - 2001, 4, 100, 400.
- ¿Cuál tiene mayor cobertura?

Funcionamiento

- El software a testear es manipulado o es ejecutado dentro de un entorno especial.
- Se establece una relación entre cada método ejecutado y el código fuente.
- Finalmente, se genera un informe sobre la cobertura de líneas, ramas, etc.

Herramientas gratuitas para Java (i)



- Cobertura:
 - Tipos de cobertura:
 - Líneas.
 - Ramas.
 - Se puede ejecutar desde línea de comandos, ant, IDE.
 - Genera informes HTML y XML distinguiendo:
 - Clase.
 - Paquete.
 - Proyecto.
 - URL: <http://cobertura.sourceforge.net/>



Herramientas gratuitas para Java (ii)

- EMMA:
 - Tipos de cobertura:
 - Líneas.
 - Métodos.
 - Clases.
 - Se puede ejecutar desde ant, IDE.
 - Genera informes HTML, XML y texto distinguiendo:
 - Método.
 - Clase.
 - Paquete.
 - Proyecto.
 - URL: <http://emma.sourceforge.net/>

Ejemplo con el Cobertura (i)

Coverage Report - datetime

Package ^	# Classes	Line Coverage	Branch Coverage	Complexity
datetime	1	75% 	83% 	7

Classes in this Package ^	Line Coverage	Branch Coverage	Complexity
YearUtilities	75% 	83% 	7

Report generated by [Cobertura](#) 1.9 on 25/01/08 2:24.

Terminado

Ejemplo con el Cobertura (ii)

The screenshot shows a Mozilla Firefox browser window titled "Coverage Report - Mozilla Firefox". The address bar shows the user is logged in as "eduardo.mosqueira". The browser's navigation bar includes back, forward, and refresh buttons, along with a search bar containing "Google".

The main content area is divided into two panes. The left pane, titled "Packages", shows a tree view with "All" and "datetime". The "Classes" section below it lists "YearUtilities (75%)".

The right pane, titled "Coverage Report - datetime.YearUtilities", displays a summary table and the source code. The summary table is as follows:

Classes in this File	Line Coverage	Branch Coverage	Complexity
YearUtilities	75% 6/8	83% 5/6	7

The source code is displayed with line numbers 1 through 15. Line 3, "public class YearUtilities", is highlighted in red with a "0" in the margin, indicating 0% coverage. Line 9, "if (year%400 == 0)", is also highlighted in red with a "3" in the margin, indicating 33% coverage. Other lines are highlighted in green, indicating 100% coverage. The code is as follows:

```
1 package datetime;
2
3 0 public class YearUtilities
4 {
5     public static boolean isLeap(int year)
6     {
7 12         if (year%4 == 0)
8 9           if (year%100 == 0)
9 3             if (year%400 == 0)
10 0              return false;
11 3              else return false;
12 6              else return true;
13 3              else return false;
14     }
15 }
```

At the bottom of the report, it states: "Report generated by Cobertura 1.9 on 25/01/08 2:24." The browser's status bar at the bottom left shows "Terminado".