

Usare il tool emma

info3

angelo gargantini @ unibg

Cosa è emma

- Emma è un tool (free) per la valutazione della copertura di codice java
- <http://emma.sourceforge.net/>
- Ci sono altri tool
 - cobertura:<http://cobertura.sourceforge.net/>

Come funziona

1. scrivi e compila il tuo codice java come al solito
 - in eclipse
 - fai nuovo progetto (ad esempio **emmaprj**)
 - fai nuova classe Prova (ad esempio dal menu file, new class)
 - Usa il package di default
 - compila e fai run normalmente

Come funziona

1. download emmaxx-lib.zip dal sito di emma o dal sito del corso ed estrai **emma.jar** salvandolo nella directory dove ci sono i file *.class da testare
 - in eclipse, nella directory che contiene tutto il progetto **emmaprj** cerca la directory con nome **classes**
 - **tipo Documents and settings/<nome utente>/workspace/emmaprj/li dentro ci sono i tuoi *.class**
2. apri una finestra DOS e fai cd fino alla directory dove hai messo emma.jar
 - se fai dir dovresti vedere emma.jar e il file .class
 - usa tab per auto completion

Usare emma

- esegui emma → coverage.txt

```
java -cp emma.jar emmarun -cp . Prova
```

- *per avere output in html (in dir. coverage)*

```
java -cp emma.jar emmarun -r html -cp . Prova
```

- per avere il sorgente marcato

```
java -cp emma.jar emmarun -r html -sp . -cp . Prova
```

Con i package e separati i class

- Se usi un package **pack**
 - i tuoi file java hanno all'inizio **package pack**
 - i *.java si trovano nella sotto directory **src/pack**
 - i *.class si trovano nella sotto directory **bin/pack**
 - Devi seguire i comandi
`java -cp emma.jar emmarun -cp .
pack.Esercizio`

- EMMA supports class, **method, line, and basic block coverage**
- EMMA does not do branch or path coverage. However, EMMA reports **fractional** line coverage to help you visualize untested branches in code.

Struttura esempi

- nei nostri esempi avremo
 - il metodo da testare con nome foooo
 - nel main metteremo le chiamate al metodo foooo corrispondenti ai casi di test che abbiamo trovato

Esempio

- Scrivi un metodo che prende un intero x e un array di interi e se x è 0 restituisce il primo intero (o 0 se l'array è vuoto), altrimenti se x non è 0, restituisce l'intero più grande nell'array se è maggiore di x , altrimenti restituisce x
- scrivi il grafo, trova i casi di test per copertura dei comandi, dei branch, decisioni e condizioni
- prova poi con emma

Esercizio

- scrivi un metodo **azzera** che dati due interi a e b e un array di interi, azzera tutti gli elementi dell'array compresi tra a e b , quelli minori di a li mette a -1 e quelli maggiori di b a $+2$
- opzionale: scrivi un metodo che dato un array di interi ne stampa il contenuto e usalo nel `main` prima e dopo la chiamata del metodo `azzera`
- scrivi il grafo, trova i casi di test per copertura dei comandi, dei branche, decisioni e condizioni, MCDC
- prova poi con emma

Esercizio 2

- Scrivi un metodo **BISESTILE** che prende in ingresso un numero **x** restituisce **true** se **x** è un anno bisestile e **false** altrimenti
- Fai il digramma di flusso di controllo del metodo, trova i percorsi per lo **statement coverage e decision e branch coverage**
- Trova i casi di test e provali in EMMA
- Quali altri casi di test servono per avere l'**MCDC**?

Con Junit e Emma

Si può fare

```
java -cp emma.jar emmarun -r html -sp . -cp
./:/home/garganti/local/eclipse/plugins/org.junit_3.8.1/junit.jar:/home/garganti/local/eclipse/plugins/org.eclipse.jdt.junit_3.1.1/junitsupport.jar:/home/garganti/local/eclipse/plugins/org.eclipse.jdt.junit.runtime_3.1.0/junitruntime.jar -r html org.eclipse.jdt.internal.junit.runner.RemoteTestRunner -version 3 -port 49039 -classNames ProvaTest
```