



Ecl-Emma-Jacoco

Mario Groneick, Marcel Brockscothen

Kurs: Softwareschnittstellen und Softwarequalität

Betreuer: Prof. Dr. Ursula Oesing

24.7.2018

Inhaltsverzeichnis

1 Ecl-Emma-Jacoco	1
1.1 Installation von Ecl-Emma-Jacoco	1
1.2 Nutzung von Ecl-Emma-Jacoco	1
1.3 Theoretische Testabdeckung und Ecl-Emma	2
2 Bistro-Verwaltungs-App	2
3 Testabdeckung der Bistro-Verwaltungs-App	3
4 Fazit	4

1 Ecl-Emma-Jacoco

Ecl-Emma ist ein Eclipse Plugin um die Testbedeckung in Eclipse zu messen. Seit Version 2.0 basiert Ecl-Emma auf der Jacoco Testabdeckungs Bibliothek für Java. Davor basierte es auf der Emma Bibliothek von Vlad Roubtsov. Das Ecl-Emma-Plugin unterstützt sowohl Junit als auch TestNG als Testframework.

1.1 Installation von Ecl-Emma-Jacoco

EclEmma ist als direktes Eclipse Plugin verfügbar und kann über den Marketplace installiert werden. Zusätzlich lässt sich Jacoco auch über Maven de-

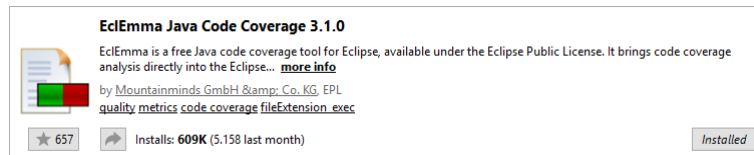


Abbildung 1: EclEmma im Marketplace

pendencies als Plug-In einbinden, dabei ist dann zu beachten, dass nur Jacoco eingebunden worden ist und nicht Ecl-Emma.

1.2 Nutzung von Ecl-Emma-Jacoco

Nach der Installation kann EclEmma direkt „out-of-box“ genutzt werden um die Testüberdeckung zu messen oder man kann es weiter Konfigurieren, so lassen sich Ordner aus der Messung exkludieren oder inkludieren. EclEmma funktioniert sowohl mit JUnit-tests als auch mit TestNG-tests und zusätzlich werden externe Testframeworks von EclEmma ebenfalls berücksichtigt. Dies wurde mit dem JavaFX UI-Testframework „testFx“ getestet.

Um die Abdeckung zu starten, muss man die Tests dann über den neuen Button in der Toolbar „Coverage“ oder „Rechtsklick → Coverage As ...“ ausführen. Nachdem die Tests durchgelaufen sind wird der Code, der getestet worden ist, farblich gekennzeichnet, Rot, Gelb oder Grün. Rot gekennzeichnete Zeilen wurden von dem Test nicht getroffen und gelten entsprechend als nicht

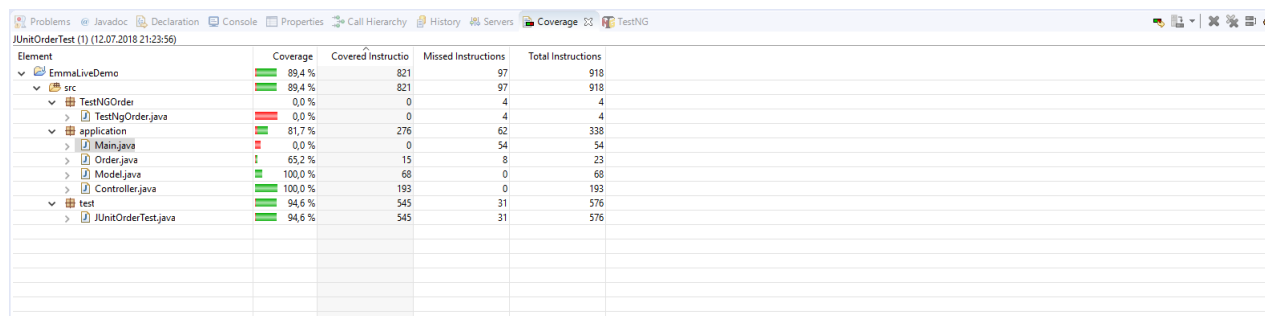
getestet. Gelbe Kennzeichnungen sind Bedingte Anweisungen, bei dem der Test keine vollständige Abdeckung der möglichen Bedingungen bietet. Grün gekennzeichnete Zeilen sind vollständig von dem Test abgedeckt.

```
private float price;

public Order(String name, float price){
    this.name = name;
    this.price = price;
}
public String getName() {
    return name;
}
public void setName(String name) {
    this.name = name;
}
public float getPrice() {
    return price;
}
public void setPrice(float price) {
    this.price = price;
}
```

Abbildung 2: Inline Text Highlightning von Ecl-Emma

Um eine bessere Übersicht für das Projekt zu haben, bietet das Plugin zusätzlich noch eine neue View an, die, wenn sie nicht bei der Installation hinzugefügt worden ist, über „Window → Show View → Other... → Coverage“ angezeigt werden lassen kann.



Element	Coverage	Covered Instruction	Missed Instructions	Total Instructions
JUnitOrderTest (1) (12.07.2018 21:23:56)				
▼ EmmaLiveDemo	89,4 %	821	97	918
▼ src	89,4 %	821	97	918
▼ TestNGOrder	0,0 %	0	4	4
> TestNgOrder.java	0,0 %	0	4	4
▼ application	81,7 %	276	62	338
> Main.java	0,0 %	0	54	54
> Order.java	65,2 %	15	8	23
> Model.java	100,0 %	68	0	68
> Controller.java	100,0 %	193	0	193
▼ test	94,6 %	545	31	576
> JUnitOrderTest.java	94,6 %	545	31	576

Abbildung 3: Coverage Overview in Eclipse

1.3 Theoretische Testabdeckung und Ecl-Emma

2 Bistro-Verwaltungs-App

Um Ecl-Emma vorzustellen wurde im Rahmen des Moduls eine Demo entwickelt. Die Demo ist eine Bistro-Verwaltungs App in der, der Nutzer vorgege-

bene Menüs auswählen kann, sich einen eigenen Salat zusammenstellen kann aus mehreren Rubriken. Zusätzlich kann der Nutzer den Gesamtbetrag der Bestellung berechnen lassen als auch Trinkgeld, in Höhe von 10% des Gesamtbetrags, geben.

3 Testabdeckung der Bistro-Verwaltungs-App

Für die Demo wurden zwei Testklassen geschrieben, eine mit dem JUnit Framework, die andere mit TestNG. Beide Testklassen decken unterschiedliche Bereiche der Demo ab. In der JUnit Testklasse wird zusätzlich zu dem JUnit Framework noch das externe Framework „testFX“ benutzt um UI Tests von einer JavaFX Anwendung durchzuführen. Die jeweiligen Testklassen haben eine Testabdeckung von 53% für den JUnit Test und eine 19,2% Abdeckung für TestNG. Hierbei ist wie schon erwähnt zu beachten, dass Ecl-Emma keine Pfadüberdeckung durchführt sondern ausschließlich Zweigabdeckung.

JUnitOrderTest (1) (13.07.2018 12:50:41)

Element	Coverage	Covered Instructio	Missed Instructions	Total Instructions
▼ EmmaLiveDemo	53,0 %	281	249	530
▼ src	53,0 %	281	249	530
▼ application	53,0 %	281	249	530
> Main.java	0,0 %	0	54	54
> Salat.java	0,0 %	0	16	16
> salatAuswahl.java	0,0 %	0	5	5
> Order.java	65,2 %	15	8	23
> Model.java	77,3 %	68	20	88
> Controller.java	57,6 %	198	146	344

Abbildung 4: Testabdeckung mit JUnit Tests

TestNgOrder (13.07.2018 12:50:16)

Element	Coverage	Covered Instructio	Missed Instructions	Total Instructions
▼ EmmaLiveDemo	19,2 %	102	428	530
▼ src	19,2 %	102	428	530
▼ application	19,2 %	102	428	530
> Controller.java	0,0 %	0	344	344
> Main.java	0,0 %	0	54	54
> Salat.java	0,0 %	0	16	16
> salatAuswahl.java	100,0 %	5	0	5
> Order.java	65,2 %	15	8	23
> Model.java	93,2 %	82	6	88

Abbildung 5: Testabdeckung mit TestNG Tests

Merged (13.07.2018 13:00:20)

Element	Coverage	Covered Instructio	Missed Instructions	Total Instructions
▼ EmmaLiveDemo	57,7 %	306	224	530
▼ src	57,7 %	306	224	530
▼ application	57,7 %	306	224	530
> Main.java	0,0 %	0	54	54
> Salat.java	0,0 %	0	16	16
> salatAuswahl.java	100,0 %	5	0	5
> Order.java	65,2 %	15	8	23
> Model.java	100,0 %	88	0	88
> Controller.java	57,6 %	198	146	344

Abbildung 6: JUnit Test Abdeckung und TestNG Abdeckung gemerged

Mit dem Ecl-Emma Plug-in lassen sich die beiden Abdeckungen mergen um

eine Übersicht über den Gesamtabdeckungsgrad der beiden Testklassen zu bekommen. Der Abdeckungsgrad des Projekts nach dem Merge durch Ecl-Emma liegt bei 57.7%.

4 Fazit

Literatur

- [1] *Software Qualität*, Dirk Hoffman (2006)
- [2] https://de.wikipedia.org/wiki/Kontrollflussorientierte_Testverfahren Zugriff 16.6.2018
- [3] <https://www.jacoco.org/index.html> Zugriff 16.6.2018
- [4] <https://www.jacoco.org/jacoco/trunk/doc/counters.html> Zugriff 16.6.2018
- [5] <https://code2flow.com/> Zugriff 16.6.2018