



JaCoCo és EcIEmma

KÉSZÍTETTE: ADONICS NIKOLETT

Kódlefedettség

- ▶ A kódlefedettség egy szoftveres mérőszám, amellyel mérjük, hogy a kódunk hány sora fut le az automatizált tesztek során.
- ▶ A következő képlettel számítható ki százalékba:
$$\frac{\text{(végrehajtott kódsorok száma)}}{\text{(Az alkalmazás kódsorainak teljes száma)}} * 100.$$

Java Code Coverage Library

- ▶ A JaCoCo egy ingyenes Java kódlefedettségi könyvtár
 - ▶ Eclipse Public License feltételei szerint terjesztik
 - ▶ Az EcEmma csapata hozott létre
- ▶ Az EMMA helyettesítésére fejlesztették ki
- ▶ Kódlefedettségi jelentéseket készít
- ▶ Jól integrálható IDE-kkel: IntelliJ IDEA, Eclipse IDE
- ▶ Használat:
 - ▶ Egység- vagy integrációs tesztek lefedettségének mérése
 - ▶ Bármilyen teszt lefedettségének mérése.

Java Code Coverage Library

- ▶ Használható:
 - ▶ különféle IDE-k, build tool-ok, folyamatos integrációs szerverek
- ▶ Támogatja:
 - ▶ Utasítás lefedettség
 - ▶ Line lefedettség
 - ▶ Branch lefedettség
 - ▶ Method lefedettség
- ▶ Lefedettségi report: `jacoco.exec`
- ▶ Report: `target/site/jacoco`
- ▶ A JaCoCo report formátuma:
 - ▶ HTML, XML és CSV

A JaCoCo-val való kódlefedettség előnyei

- ▶ Átfogó HTML jelentéseket biztosít az egyszerű vizsgálathoz
- ▶ Könnyen kombinálható a Mavennel és további build és CI/CD eszközökkel
- ▶ Különböző lefedettségi mérőszámokat támogat (branch, line, method)
- ▶ Alkalmas minőség-ellenőrzésre és folyamatos integrációra

Mi a különbség a JUnit és a JaCoCo között?

JUnit

- ▶ Az egységtesztek végrehajtására és menedzselésére összpontosít
- ▶ Támogatja a tesztesetek írását és a várt eredmények ellenőrzését
- ▶ Fejlesztőközpontú

JaCoCo

- ▶ Egy kódlefedettségi eszköz a tesztelés mértékének nyomon követésére
- ▶ Lefedettségi jelentéseket készít a tesztelés nélküli kódterületek azonosítására
- ▶ Tesztelőcsoportok a kódlefedettség és -minőség felmérésére

Mind a JUnit, mind a JaCoCo integrálható build és elemző eszközökkel a tesztelési és kódminőségi folyamatok egyszerűsítése érdekében.

EclEmma

- ▶ Az EclEmma egy ingyenes Java kód lefedettségi eszköz az Eclipse számára
 - ▶ az Eclipse Public License feltételei szerint terjesztik
- ▶ Eredetileg az EclEmmát EMMA könyvtár ihlette
- ▶ Az EclEmma a JaCoCo kódlefedettségi könyvtáron alapul
- ▶ Jellemzők:
 - ▶ Gyors fejlesztési/tesztelési ciklus
 - ▶ Gazdag lefedettség-elemzés
 - ▶ Nem invazív

A JaCoCo használata kézzel

- ▶ tesztelesimodszerek-main-jacoco projekt
- ▶ Jacoco Agent és CLI letöltése: <https://www.jacoco.org/jacoco/>
 - ▶ jacoco-0.8.11 -> lib -> jacocoagent.jar és jacococli.jar
- ▶ Parancsok:
 - ▶ java
-javaagent:jacocoagent.jar=destfile=target/jacoco.exec,append=false
 - ▶ java -jar jacococli.jar report ./target/jacoco.exec --classfiles
./target/classes --html ./target/coverage/ --sourcefiles ./src/main/java

A JaCoCo használata build rendszerekben - Maven


- ▶ tesztesimodszerek-main-jacoco-build projekt
- ▶ Lefedettségi report: jacoco.exec
- ▶ Kimenet értelmezése:
 - ▶ eszközök és bővítmények, pl. Sonar Qube
 - ▶ jacoco:report -> target/site/jacoco




```
<plugin>
  <groupId>org.jacoco</groupId>
  <artifactId>jacoco-maven-
plugin</artifactId>
  <version>0.8.11</version>
  <executions>
    <execution>
      <goals>

<goal>prepare-agent</goal>
      </goals>
    </execution>
    <execution>
      <id>report</id>

<phase>prepare-package</phase>
      <goals>
        <goal>report</goal>
      </goals>
    </execution>
  </executions>
```

Report

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
 default	<div><div></div></div>	69%	<div><div></div></div>	70%	15	51	34	123	11	41	0	3
Total	131 of 434	69%	6 of 20	70%	15	51	34	123	11	41	0	3

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
 Kamra	<div><div></div></div>	36%	<div><div></div></div>	16%	11	18	29	49	8	15	0	1
 Torta	<div><div></div></div>	88%	<div><div></div></div>	90%	4	16	5	32	3	11	0	1
 Macska	<div><div></div></div>	100%	<div><div></div></div>	100%	0	17	0	42	0	15	0	1
Total	131 of 434	69%	6 of 20	70%	15	51	34	123	11	41	0	3

Kamra.java

```
1. public class Kamra {
2.
3.     private double területNegyzetmeterben; //m^2-ben
4.
5.     private int polcokSzama;
6.
7.     private double hasznosTerulet;
8.     private Macska macska;
9.     private Torta torta;
10.
11.     public Kamra(Macska macska, Torta torta, int polcokSzama) {
12.         this.teruletNegyzetmeterben = 5;
13.         if (polcokSzama < 0) polcokSzama = 0;
14.         this.polcokSzama = polcokSzama;
15.         this.macska = macska;
16.         this.torta = torta;
17.         this.hasznosTerulet = getHasznosTerulet();
18.     }
19.
20.     public void kinaldMegAMacskattortaval() {
21.         torta.tortaszeletElfogyasztasa();
22.         macska.eszik(10);
23.     }
24.
25.     public void etetMacska(int etelMennyiseg) {
26.         macska.eszik(etelMennyiseg);
27.         macska.setSimogatasokSzama(macska.getSimogatasokSzama()+1); // Macska boldogságának növelése
28.     }
29.
30.     public void bovites(int negyzetmeter) {
31.         területNegyzetmeterben += negyzetmeter;
32.         this.hasznosTerulet = getHasznosTerulet();
33.     }
34.
35.     public void informacioKiiras() {
36.         System.out.println("\nKamra információk:");
37.         System.out.println("Macska információ:");
38.         System.out.println("Név: " + macska.getNev());
39.         System.out.println("Súly: " + macska.getSuly());
40.         System.out.println("Barátságos: " + macska.isBaratsagos());
41.         System.out.println("Bolhák száma: " + macska.getBolhakSzama());
42.     }
43. }
```


EclEmma

- ▶ tesztelesimodszerek-main-eclemma projekt
- ▶ Telepítés:
 - ▶ Help -> Eclipse Marketplace -> EclEmma Java Code Coverage installation
- ▶ Futtatás:
 - ▶ Debug -> Coverage As -> Java Application