

WPA - Ladení a testování webové aplikace

Miroslav Blaško, Bogdan Kostov

October 15, 2014

Obsah

1 Maven

2 JUnit

Základní koncepty

- artefakt (artefact)
- koordináty projektu (project coordinates)
 - groupId
 - artifactId
 - version
- závislost (dependency)
- repozitář (repository)
- životní cyklus (lifecycle)
- fáze (phase)
- cíl (goal)
- plugin

Vybrané příkazy

- Základní :
 - mvn clean – smazání target adresáře
 - mvn package – zabalení zkompilevaného kódu do distribuovatelného formátu (např. WAR, JAR)
 - mvn install – nainstalování artefaktu do lokálního repozitáře
- Ladení příkazů a závislostí:
 - mvn -X *PŘÍKAZ* – zobrazení ladících hlášek
 - mvn dependency:tree – zobrazení stromu závislostí
- Testování:
 - mvn test – spuštění všech testů
 - mvn -Dtest=*cz.cvut.SomeClassTest* surfire:test – spuštění konkrétního testu
 - mvn -DskipTests *PŘÍKAZ* – přeskočení testů (např. mvn -DskipTests package)
 - mvn cobertura:cobertura – generování stránek pro analýzu pokrytí testami

Další informace

- `HOME_DIR/.m2/repository` – adresář lokálního repozitáře
- Hledání artefaktů v hlavních repozitářích
 - <http://search.maven.org/>
 - <http://mvnrepository.com/>

jUnit I.

- Testovací framework
- Od verze 4 využívá anotace pro deklaraci testovacích případů
- Třída reprezentuje sadu testů (anotované metody pak slouží k jejich výkonu)
 - *@BeforeClass* — statická inicializační metoda, provede se při inicializaci testovací sady
 - *@Before* — metoda se provede před každým jednotlivým testem
 - *@Test* — testovací případ
 - *@After* — metoda se provede po každém jednotlivém testu
 - *@AfterClass* — metoda se provede po provedení všech testů

JUnit II.

- Pro kontrolu splnění dílčích invariantů používáme statické metody třídy *Assert* (obvykle je importujeme do testovací sady všechny pomocí deklarace *import static org.junit.Assert.**)

```
@Test
public void testBookDeletedWhenUserRemoved() {
    Long userId = addUser();
    String title = "Bob a Bobek, kralici z
        klobouku";
    bookService.addBook(title, userId);
    assertEquals(1, bookService.getAllBooks().
        size());
    userService.deleteUser(userId);
    assertEquals(0, bookService.getAllBooks().
        size());
}
```

JUnit III.

- Je vhodné také použít knihovnu *Cobertura*, jež generuje report pokrytí kódu testy (řádky, větve)
- IDE *Netbeans* umí generovat kostru testů (pravé tlačítko na soubor v Záložce `projects/Tools/Create JUnit Tests`)