
Ladění programů, testování jednotek (junit)

Obsah

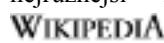
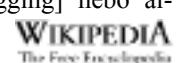
Ladění programů (debugging)	1
Ladění programů v Javě	1
Ještě lepší...	2
Běhové ověření podmínky - assert	3
Postup při práci s assert 	3
Ukázka použití assert (1)	3
Ukázka použití assert (2)	3
Ukázka použití assert (3)	4
Ukázka použití assert (4)	4
Testování jednotek nástrojem junit	4
Postup při práci s JUnit 	4
Ukázka použití JUnit (1)	5
Ukázka použití JUnit (2)	5
Ukázka použití JUnit (3)	5
Ukázka použití JUnit (4)	6
Návrh dle kontraktu	6
Postup při práci s jass 	6
Odkazy	7

Ladění programů (debugging)

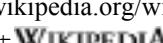
- Ladění programů s debuggerem jdb
- Nástroje ověřování podmínek za běhu - klíčové slovo assert
- Nástroje testování jednotek (tříd, balíků) - junit
- Pokročilé systémy dynamického ověřování podmínek - jass

Ladění programů v Javě

Je mnoho způsobů...

- kontrolní tisky - System.err.println(...) 
[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD%20n%C3%ADm%C3%AD%20na%20z%C3%A1znam%20b%C3%A9hu%20pg.:>]
- řádkovým debuggerem jdb 
[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD%20n%C3%ADm%C3%AD%20na%20z%C3%A1znam%20b%C3%A9hu%20pg.:>]
- integrovaným debuggerem v IDE
- pomocí speciálních nástrojů na záznam běhu pg.:
nejrůznější "loggery" - standardní poskytuje od JDK1.4 balík java.util.logging 
[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD%20n%C3%ADm%C3%AD%20na%20z%C3%A1znam%20b%C3%A9hu%20pg.:>] nebo alternativní a zdařilejší log4j 
[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD%20n%C3%ADm%C3%AD%20na%20z%C3%A1znam%20b%C3%A9hu%20pg.:>]

Ještě lepší...

- je používat systémy pro běhovou kontrolu platnosti podmínek:
 - vstupní podmínka metody (zda je volána s přípustnými parametry)
 - výstupní podmínka metody (zda jsou dosažené výstupy správné)
 - a podmínka kdekoliv jinde - např. invariant cyklu...
- K tomuto slouží jednak
 - standardní klíčové slovo (od JDK1.4) assert booleovský výraz 
[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD%20n%C3%ADm%C3%AD%20na%20z%C3%A1znam%20b%C3%A9hu%20pg.:>]
 - testovací nástroje typu JUnit 
[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD%20n%C3%ADm%C3%AD%20na%20z%C3%A1znam%20b%C3%A9hu%20pg.:>] (a varianty - HttpUnit 
[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD%20n%C3%ADm%C3%AD%20na%20z%C3%A1znam%20b%C3%A9hu%20pg.:>]) - s metodami assertEquals() 
[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD%20n%C3%ADm%C3%AD%20na%20z%C3%A1znam%20b%C3%A9hu%20pg.:>] - s metodami assertEquals()
[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD%20n%C3%ADm%C3%AD%20na%20z%C3%A1znam%20b%C3%A9hu%20pg.:>]
apod.
 - pokročilé nástroje na běhovou kontrolu invariantů, vstupních, výstupních a dalších podmínek - např. jass 
[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD%20n%C3%ADm%C3%AD%20na%20z%C3%A1znam%20b%C3%A9hu%20pg.:>] (Java with ASSer-tions), <http://csd.informatik.uni-oldenburg.de/~jass/>.

- Ukázka testu jednotky:Test třídy ChovatelPsu 
[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD_Search?search=ChovatelPsu]
[<http://www.fi.muni.cz/~tomp/java/ucebnice/javasrc/svet/chovatelstvi/psi/ChovatelPsuTest.java>]

Běhové ověření podmínky - assert

Postup při práci s assert

[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD_Search?search=assert]

Postup:

1. Napsat zdrojový program užívající klíčové slovo assert 
[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD_Search?search=assert] (pouze od verze Java2 v1.4 výše). Nepotřebujeme žádné speciální běhové knihovny, vše je součástí Javy; musíme ovšem mít překladové i běhové prostředí v1.4 a vyšší.
2. Přeložit jej s volbou -source 1.4
3. Spustit jej s volbou -ea (-enableassertions).
Aktivovat aserce lze i selektivně pro některé třídy (-ea název_třídy nebo -ea název_balíku... - tři tečky na konci!!!).
4. Dojde-li za běhu programu k porušení podmínky stanovené za assert, vznikne běhová chyba (AssertionError) a program skončí.

Ukázka použití assert (1)

Třída Zlomek používá assert k ověření, že zlomek není (chybou uživatele) vytvářen s nulovým jmenovatelem.

Za

assert 

[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD_Search?search=assert] uvedeme, co musí v daném místě za běhu programu platit.

Obrázek 1. Třídy AssertDemo, Zlomek

Ukázka použití assert (2)

Program přeložíme (s volbou `-source 1.4`):

Obrázek 2. Správný postup překladu AssertDemo

Ukázka použití assert (3)

Program spustíme (s volbou `-ea` nebo selektivním `-ea:NázevTřídy`):

Obrázek 3. Spuštění AssertDemo s povolením assert

Ukázka použití assert (4)

Spusťme-li bez povolení assert (bez volby `-ea` nebo naopak s explicitním zákazem: `-da`), pak program podmínky za assert neověřuje:

Obrázek 4. Spuštění AssertDemo bez povolení assert

Testování jednotek nástrojem junit

Postup při práci s JUnit

[<http://cs.wikipedia.org/wiki/Spec%C3%A1ln%C3%AD%Search?search=JUnit>]

Uvědomit si, že žádný nástroj za nás nevymyslí, JAK máme své třídy testovat. Pouze nám napomůže ke snadnějšímu sestavení a spuštění testu.

Postup:

1. Stáhnout si z <http://junit.org> poslední (stačí binární) distribuci testovacího prostředí.
2. Nainstalovat  [<http://cs.wikipedia.org/wiki/Spec%C3%A1ln%C3%AD%Search?search=JUnit>] (stačí rozbalit do samostatného adresáře).
3. Napsat testovací třídu/třídy - budě implementují rozhraní `junit.framework.Test`  [<http://cs.wikipedia.org/wiki/Spec%C3%A1ln%C3%AD%Search?search=junit.framework.Test>] ne-



[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD_Search?search=junit.framework.TestCase]

4. Testovací třída obsahuje metodu na nastavení testu (setUp), testovací metody (testNeco) a úklidovou metodu (tearDown).
5. Testovací třídu spustit v prostředí (řádkovém nebo GUI) - `junit.textui.TestRunner`

[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD_Search?search=junit.textui.TestRunner],
`junit.swingui.TestRunner`

[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD_Search?search=junit.swingui.TestRunner]
...
6. Testovač zobrazí, které testovací metody případně selhaly.

Ukázka použití JUnit (1)

Třída Zlomek zůstává zhruba jako v předchozím příkladu, přibývají však metody equals (porovnává dva zlomky, zda je jejich číselná hodnota stejná) a součet (sečítá dva zlomky, součet vrací jako výsledek).

Obrázek 5. Upravená třída Zlomek s porovnáním a součtem

Ukázka použití JUnit (2)

Testovací třída JUnitDemo má „přípravnou“ metodu setUp, tearDown a testovací metody.

Obrázek 6. Testovací třída JUnitDemo

Ukázka použití JUnit (3)

Spuštění testovače v prostředí GUI Swing nad testovací třídou JUnitDemo.

Obrázek 7. Spouštění testovače nad testovací třídou JUnitDemo

Pokud testovací třída prověří, že testovaná třída/y je/jsou OK, vypadá to přibližně takto:

Obrázek 8. Testovač spouštějící třídu JUnitDemo

Ukázka použití JUnit (4)

Má-li testovaná třída/y chyby a zjistí-li to testovač, vypadá to třeba takto:

Obrázek 9. Testovací třída JUnitDemo našla chybu

Návrh dle kontraktu

Postup při práci s jass

 WIKIPEDIA
The Free Encyclopedia

[<http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD%Search?search=jass>]

jass je preprocesor javového zdrojového textu. Umožňuje ve zdrojovém textu programu vyznačit podmínky, jejichž splnění je za běhu kontrolováno.

Podmínkami se rozumí:

- pre- a postconditions u metod (vstupní a výstupní podmínky metod)
- invarianty objektů - podmínky, které zůstávají pro objekt v platnosti mezi jednotlivými operacemi nad objektem

Postup práce s jass:

- stažení a instalace balíku z <http://csd.informatik.uni-oldenburg.de/~jass/>
- vytvoření zdrojového textu s příponou .jass  WIKIPEDIA
The Free Encyclopedia [http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD%Search?search=.jass], javovou syntaxí s použitím speciálních komentárových značek
- takový zdrojový text je přeložitelný i normálním překladačem javac  WIKIPEDIA
The Free Encyclopedia [http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD%Search?search=javac], ale v takovém případě ztrácíme možnosti jass
- proto nejprve .jass  WIKIPEDIA
The Free Encyclopedia [http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD%Search?search=.jass] souboru převedeme preprocesorem jass na javový (.java  WIKIPEDIA
The Free Encyclopedia [http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD%Search?search=.java]) soubor
-



[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD_Search?search=javac] a

spustíme



[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD_Search?search=java], tedy jako každý jiný zdrojový soubor v Javě

- z



[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD_Search?search=.jass] zdrojů je možné vytvořit také dokumentaci API obsahující jass značky, tj. informace, co kde musí platit za podmínky atd. - vynikající možnost!

Odkazy

- JUnit homepage [<http://junit.org>]
- Java 1.4 logging API guide [<http://java.sun.com/j2se/1.4.2/docs/guide/util/logging/>]
- Log4j homepage [<http://jakarta.apache.org/log4j/docs/index.html>]
- jass homepage [<http://csd.informatik.uni-oldenburg.de/~jass/>]
- úvodní materiálek [<http://www.inf.fu-berlin.de/lehre/SS01/VIS/Dokumente/Vortraege/junit.pdf>] k použití junit (v němčině, jako PDF)