Obsah

1. Úvod 1
   1.1. Proč tato knížka vznikla 1
   1.2. Co můžete od knížky očekávat 1
   1.3. Poděkování 2
   1.4. Konvence 3
2. Základní pojmy 4
   2.1. Úvod 4
   2.2. Základní důvody, proč testovat 4
   2.3. Nejdůležitější termíny 7
      2.3.1. Termíny chyba, porucha a selhání 7
      2.3.2. Testování nedokáže říci, že program je bez chyb 9
      2.3.3. Verifikace, validace a testování 9
      2.3.4. Zajištění kvality softwaru 10
      2.3.5. Vyhodnocení správnosti chování testovaného programu 10
      2.3.6. Kdy se vyplatí s testováním skončit 11
      2.3.7. Pozitivní a negativní testy 12
      2.3.8. Test dává opačné výsledky – je chybný 14
      2.3.9. Priorita versus severita 15
      2.3.10. Testovatelnost 16
      2.3.11. Metriky 16
      2.3.12. Popis jednotlivých testů a jejich skupin 17
   2.4. Typy testů 18
      2.4.1. Členění testů 18
      2.4.2. Testovací mix 22
      2.4.3. Jaké typy testů jsou v praxi v ČR používány 22
      2.4.4. Přehled základních typů testů 24
3. Jednotkové testování 28
   3.1. Úvodní informace 28
      3.1.1. Umístění JUnit 29
   3.2. Princip použití 29
      3.2.1. Ukázka jednoduché entitní třídy pro testování 29
      3.2.2. První testovací případ 33
      3.2.3. Spuštění testů z příkazové řádky 35
      3.2.4. Použití testů z Eclipse 36
      3.2.5. JUnit test odhalil chybu 40
   3.3. Základní informace o použití jednotkových testů 45
   3.4. Běžně využívané možnosti při psaní testovacích případů 47
      3.4.1. Parametry anotace @Test 47
      3.4.2. Testovací metody z org.junit.Assert 53
      3.4.3. Akce před a po spuštění testovacích případů 60
   3.5. Praktické náležitosti 64

XML to PDF by RenderX XEP XSL-FO Formatter, visit us at http://www.rend

3.5.1. Ukázka testů entitní třídy 64
3.5.2. Kam se soubory s třídami testů fyzicky umísťují 67
3.5.3. Jak se spouští najednou testy z více testovacích tříd 68
3.5.4. Způsob pojmenovávání testovacích metod 70
3.6. Další způsoby spouštění 72
3.6.1. Spouštění z Ant 73
3.6.2. Spouštění z Java programu 75
4. Antivzory v používání jednotkových testů 79
4.1. Obecně nedoporučované praktiky 79
4.1.1. Nevhodné použití aserce 79
4.1.2. Příliš složité testy 86
4.2. Konkrétní špatné techniky 89
4.2.1. Lhář (*The Liar*) 89
4.2.2. Nadměrná příprava (*Excessive Setup*) 89
4.2.3. Inspektor (*The Inspector*) 90
4.2.4. Velkorysé zbytky (*Generous Leftovers*) 92
4.2.5. Vykukující kocour (*The Peeping Tom*) též Nezvaní hosté (*The Uninvited Guests*) 92
4.2.6. Místní hrdina (*The Local Hero*) 93
4.2.7. Kazatel operačního systému (*The Operating System Evangelist*) 93
4.2.8. Hnidopich (*The Nitpicker*) 93
4.2.9. Tajemný lovec (*The Secret Catcher*) 94
4.2.10. Ulejvák (*The Dodger*) 95
4.2.11. Křikloun (*The Loudmouth*) 97
4.2.12. Nenasytný lovec (*The Greedy Catcher*) 97
4.2.13. Výčet (*The Enumerator*) 98
4.2.14. Cizinec (*The Stranger*) 99
4.2.15. Šťastná cesta (*Happy Path*) 99
4.2.16. Podřadní občané (*Second Class Citizens*) 100
4.2.17. Pomalé dloubnutí (*The Slow Poke*) 101
4.2.18. Jízda zadarmo (*The Free Ride*) 101
4.2.19. Bezejmenný test (*The Test With No Name*) 101
4.2.20. Spáč (*The Sleeper*) 101
4.2.21. Obr (*The Giant*) 101
4.2.22. Imitátorna (*The Mockery*) 101
4.2.23. Nebezpečnost jednotlivých antivzorů 102
5. Pokročilé možnosti frameworku JUnit 103
5.1. Matchers a metoda assertThat() 104
5.1.1. Nejdůležitější matchery z org.hamcrest.CoreMatchers a dalších 106
5.2. Pořadí vykonávání testů 112
5.3. Ignorování testů 114
5.4. Anotace @RunWith() 115
5.4.1. Spojování tříd testů do skupin – @RunWith(Suite.class) 115

XML to PDF by RenderX XEP XSL-FO Formatter, visit us at http://www.rend

5.4.2. Kategorie testů – @RunWith(Categories.class) 117
5.5. Parametrizované testy – @RunWith(Parameterized.class) 121
5.5.1. Základní použití 122
5.5.2. Zajištění přehlednějšího chybového výpisu 123
5.5.3. N-tice parametrů 124
5.5.4. Programové generování parametrů testu 126
5.5.5. Načítání parametrů ze souboru 128
5.5.6. Předávání parametrů přes konstruktor 130
5.6. Rules 131
5.6.1. TemporaryFolder 132
5.6.2. ErrorCollector 133
5.6.3. ExternalResource 135
5.6.4. Verifier 136
5.6.5. TestWatcher 138
5.6.6. TestName 140
5.6.7. Timeout 141
5.6.8. ExpectedException 142
5.6.9. Stopwatch 144
5.7. Doplňkové akce s Rules 146
5.7.1. Několik *Rules* najednou 146
5.7.2. ClassRule 147
5.7.3. Použití ClassRule pro skupinu tříd 148
5.8. Assumptions – předpoklady testu 150
5.8.1. Assume pro @Before nebo @BeforeClass 153
6. Mockování 155
6.1. Ukázka doménové třídy s těsnými vazbami 156
6.1.1. Rozvolnění závislostí mezi třídami 157
6.2. Ukázka upravené doménové třídy 159
6.3. Principy řešení problému „rozplizlého“ testování 160
6.3.1. Implementace *stub* objektů 161
6.3.2. Implementace *mock* objektů 163
6.4. Framework EasyMock – základní dovednosti 163
6.4.1. První testovací případ 164
6.4.2. Využití @Before a @After 166
6.4.3. Využití více záznamů v *mock* objektu 168
6.4.4. Více testovacích případů 170
6.4.5. Spouštění z Ant 173
6.5. *Mock* objekt bez použití *bussines interface* 174
6.5.1. Předchozí případ pomocí „*class*“ *mock* objektu 174
6.5.2. Dva nezávislé *mock* objekty 178
6.5.3. Využití parametrizovaného testu 183
6.6. Speciality 185
6.6.1. Různé typy *mock* objektů 185

XML to PDF by RenderX XEP XSL-FO Formatter, visit us at http://www.rend

6.6.2. Opakování jednou nahraného volání ... 186
6.6.3. Testování void metod a výjimek ... 186
6.6.4. *Mock* objekty s nekonkrétními formálními parametry ... 188
7. Techniky strukturálního testování ... 196
7.1. Ukázka možností ... 198
7.2. Druhy pokrytí ... 199
7.2.1. K čemu je to dobré? ... 202
7.3. Nástroje pro zjištění pokrytí kódu v Javě ... 202
7.3.1. Seznam *open-source* nástrojů ... 203
7.4. Nástroj EclEmma ... 204
7.4.1. První spuštění EclEmma pro ověření správnosti instalace ... 204
7.4.2. Zobrazení naměřených výsledků ... 210
7.4.3. Konfigurace ... 216
7.4.4. Export výsledků ... 216
7.5. Nejužívanější druhy pokrytí kódu ... 217
7.5.1. Pokrytí příkazů ... 217
7.5.2. Pokrytí rozhodnutí/logických podmínek – DC ... 223
7.5.3. Pokrytí podmínek – CC ... 225
7.5.4. Rozdíly mezi DC, CC, C/DC, MC/CD a MCC ... 227
7.5.5. Testování základních cest ... 238
7.5.6. Cyklomatická složitost ... 239
8. Jak připravit testovací data ... 245
8.1. Rozdělení tříd ekvivalence ... 246
8.1.1. Typy tříd ekvivalence ... 247
8.1.2. Statické a dynamické hodnoty ... 249
8.1.3. Problémy dekompozice – počet tříd ekvivalence ... 249
8.1.4. Praktický případ – třída ekvivalence pro datum ... 250
8.2. Testování hraničních hodnot ... 255
8.2.1. Typy hraničních hodnot ... 256
8.2.2. Testování hraničních případů ... 257
8.2.3. Počty testů a postup ... 257
8.2.4. Triviální optimální příklad testování hraničních případů ... 258
8.2.5. Problémový příklad testování hraničních případů ... 259
8.3. Rozhodovací tabulky ... 265
8.3.1. Konkrétní případ ... 266
8.4. Speciální případy ... 272
8.4.1. Subhraniční podmínky ... 272
8.4.2. Unikátní hodnoty ... 273
8.4.3. Zvláštní podmínky ... 274
8.4.4. Prázdné hodnoty ... 274
8.4.5. Neplatné, chybné, nesprávné a nesmyslné údaje ... 275
9. Testování webových aplikací ... 276
9.1. Specifika webových aplikací ... 276

XML to PDF by RenderX XEP XSL-FO Formatter, visit us at http://www.rend

9.1.1. Značně různá složitost na technologické úrovni 276
9.1.2. Časté změny 277
9.1.3. Proměnlivá zátěž 277
9.1.4. Ztížená práce se soubory 278
9.1.5. Bezestavovost 278
9.1.6. Možná identifikace ovládacích prvků 278
9.1.7. Množství používaných web prohlížečů a jejich verzí 279
9.2. Testovací framework Selenium 279
9.3. Selenium WebDriver 280
9.3.1. Úvodní informace 280
9.3.2. Ukázka jednoduchého kompletního funkčního testu 281
9.3.3. Základní možnosti Selenium WebDriver API 283
9.3.4. Test statických prvků 293
9.3.5. Testování na konkrétním webovém prohlížeči 297
10. Logování 300
10.1. Základní pojmy 300
10.2. Apache Log4j 2 304
10.2.1. Logované informace 305
10.2.2. Úrovně závažnosti 306
10.2.3. Základní použití logování 307
10.2.4. Základy konfigurace 314
10.2.5. Dva typy XML konfiguračních souborů 332
10.2.6. Appendery 336
10.2.7. Layouty 345
10.2.8. Filtry 351
10.2.9. Realistický příklad 360
11. Statická analýza kódu 367
11.1. Základní pojmy, přístupy a dosah statické analýzy 367
11.1.1. Pojmy ohledně čitelnosti zdrojového kódu 369
11.1.2. Pojem technický dluh 370
11.1.3. Obecné vlastnosti statických analyzátorů 371
11.2. PMD 372
11.2.1. Spuštění z příkazové řádky 374
11.2.2. Pravidla a jejich seznamy 377
11.2.3. Potlačení chybového hlášení 381
11.2.4. Praktické zkušenosti 382
11.2.5. Integrace do Eclipse 385
12. Použitá literatura 392

XML to PDF by RenderX XEP XSL-FO Formatter, visit us at http://www.rend