

# Index

- Abbot framework, 312–317
- AbbotFixture class, 314–315
- Absolute performance testing, 86
- AbstractAnnotationAwareTransactionalTests class, 278–279
- AbstractDependencyInjectionSpringContextTests class, 244, 274–275
- AbstractSingleSpringContextTests class, 273
- AbstractSpringContextTests class, 272–273
- AbstractTransactionalDataSourceSpringContextTests class, 278
- AbstractTransactionalSpringContextTests class, 238, 244, 276–278
- AbstractWSTest class, 249
- activeAfter attribute, 367–371
- ActiveMQ message broker, 221
  - downloading and installing, 221–222
  - running, 222–223
  - state handling, 223–225
- @AfterClass annotation, 20, 170, 424
- AfterClassCalledAtEnd test, 356
- @AfterGroups annotation, 20, 424
- @AfterMethod annotation, 20, 424
- @AfterSuite annotation, 20, 170, 424
- @AfterTest annotation, 20, 424
- Algorithm complexity, 84–87
- AllTests pattern, 463–467
- alwaysRun attribute, 425, 427
- annotation option in JUnitConverter, 450
- Annotation transformers, 346
  - annotation history, 346–347
  - pros and cons, 348
  - usage, 353–355
  - working with, 348–353
- Annotations
  - custom, 366–367
    - implementation, 367–371
    - testing, 371–375
  - inheritance, 113–116
  - IntelliJ IDEA, 419
  - Javadocs, 423–428
  - JDK 1.4 and JDK 5, 421–423
  - scopes, 113–119
  - TestNG framework, 17–20
- annotations attribute
  - <suite>, 438
  - <test>, 445
- AnnotationServiceFactory class, 250
- AnnotationTransformerClassSampleTest class, 328
- ant task
  - JUnitConverter from, 452–453
  - for running groups, 126
- Anthill, 321
- Assert class, 457
- assert functions, 457
- Assertions, 265, 457–458
- AssertJUnit class, 457
- Asynchronous testing, 67–71
- Atomikos TransactionEssentials transaction manager, 218
- Attributes
  - ant, 126, 452–453
  - encapsulated, 36
  - factories, 35–36
  - Javadocs, 421–422
  - <package>, 441
  - <parameter>, 441–442
  - <selector-class>, 443
  - <suite>, 437–440
  - <suite-file>, 442
  - <test>, 444–445
- Bamboo, 321
- Base classes
  - dependencies, 116–118
  - JUnit tests, 4
- Bean Scripting Framework (BSF), 337

- BeanShell language, 335
  - interactive execution, 339–341
  - overview, 335–337
  - with TestNG, 337–339
- @BeforeClass annotation, 20, 69, 111–112, 424
- @BeforeGroups annotation, 20, 112, 424
- @BeforeMethod annotation, 20, 111–112, 424
- @BeforeSuite annotation, 20, 112, 424
- @BeforeTest annotation, 20, 112, 424
- Big O notation, 84–87
- Binary code coverage implementation, 146
- Block coverage, 135–136
- Bootstrapping in JNDI, 207–209
- Branch coverage, 136
- Broken tests, 127–129
- BSF (Bean Scripting Framework), 337
- Bugs, debuggers for, 392–394
- Build tool integration in code coverage, 137
- build.xml template, 12, 137–138
  - for Clover, 139
  - for converting JUnit, 452–453
  - for EMMA, 142
  - groups in, 126
  - parsing, 322
- Builds, CI server, 321
  
- c3p0 library, 212–213
- Caching Spring containers, 271, 273, 279
- Callback methods in asynchronous testing, 68–69
- Cascade failures, 103
- Categories, test group, 129–131
- Check-in tests, 130
- Checked exceptions
  - avoiding, 381
  - design patterns, 25–27
- CI (continuous integration), 320
  - benefits, 320
  - servers, 320–321
  - TestNG, 321–322
- Class coverage, 135
- <class> tag, 331, 446–447
- Class-wide initialization, 463
- Classes
  - dependencies, 116–118
  - Eclipse for, 405
  - IntelliJ IDEA tests, 415–416
  - JUnit tests, 3–4
  - mocks and stubs, 99–100
- <classes> tag, 446
  
- Classloaders in code coverage, 146
- ClassScanner class, 201–202
- CLEAN\_INSERT operation, 298–299
- Clients
  - embedded containers, 258–259
  - JAX-WS, 246
- clover-setup task, 140
- Clover tool, 138–141
- Cobertura tool, 144–146
- Code conversion. *See* JUnit migration
- Code coverage, 132–133
  - Clover tool, 138–141
  - Cobertura tool, 144–146
  - design for, 149
  - difficulties, 148
  - EMMA tool, 142–144
  - example, 133–134
  - guidelines, 147–148
  - history, 150
  - implementation, 146–147
  - importance, 149
  - limitations, 147, 149–150
  - metrics, 134–136
  - percentages, 148–149
  - reports, 137, 148
  - tools, 136–138
- CollectingAlertHandler class, 309
- ComboPooledDataSource class, 213
- Command line
  - JUnitConverter, 449–452
  - running groups from, 125–126
- Commons DBCP library, 213
- Competing Consumers pattern, 182–186
- Compiling with Clover, 140
- Complex classes, mocks and stubs for, 99–100
- Complexity
  - algorithm, 84–87
  - mocks, 102
- Component Injection, 210
- ComponentDataProvider class, 180–181
- ComponentFunctionalTests class, 179–180
- ComponentTester interface, 316
- ComponentUnitTests class, 180
- concurrent package, 67–68
- Concurrent running, 79–82
- Concurrent testing, 72–78
- Configuration
  - ActiveMQ, 223
  - DbUnit, 295–297

- dependencies in, 110–112
- Eclipse, 404–410
- Guice, 286–290, 293–295
- HtmlUnit, 304–305
- JUnit tests, 6
  - logging, 396
  - Swing UI, 313–314
- Connection interface, 168–170
- Connections
  - JDBC, 211
  - transaction rollbacks, 168–170
- Containers
  - EJB3, 243–246
  - Java EE testing. *See* In-container testing
  - JPA simulation, 230–236
  - servlets, 257–260
- Continuous integration (CI), 320
  - benefits, 320
  - servers, 320–321
  - TestNG, 321–322
- Continuum product, 321
- Contract capture, mocks and stubs for, 100
- Conversions, code. *See* JUnit migration
- CreditCardTest class, 113–114
- .csv files, 63–65
- Custom annotations, 366–367
  - implementation, 367–371
  - testing, 371–375
- CvsReader class, 66
- d option for JUnitConverter, 450
- Dashboard
  - CI servers, 321
  - Clover, 141
- Data Access Object (DAO), 282
- Data Access Object (DAO) pattern, 90
- Data-Driven Testing
  - Data Providers. *See* Data Providers
  - design patterns, 39–42
  - enterprise, 164, 173
  - parameters and test methods, 42–50
- Data Providers, 47–50
  - Competing Consumers pattern, 185
  - data for, 60–62
  - enterprise testing, 164–165
  - vs. factories, 62–63
  - ITestContext parameters for, 52–54
  - lazy, 54–59
  - Method parameters for, 50–52
  - summary, 63–67
  - vs. testng.xml, 59–60
- Data size in algorithm complexity, 84–85
- Databases
  - enterprise testing, 166–167
  - JPA, 227
- @DataProvider annotation
  - default reports, 356
  - enterprise testing, 165
  - overview, 425–426
  - passing parameters with, 47–60
- dataProvider attribute
  - annotation transformers, 354
  - @Test, 427
- dataProviderClass attribute
  - annotation transformers, 354
  - @Test, 427
- DataSource class
  - c3p0 library for, 212–213
  - transaction rollbacks, 168
- DBTestCase example, 299–301
- DBTests class, 302–303
- DbUnit, 211
  - configuration, 295–297
  - features, 295
  - usage, 297–299
  - verifying results, 299–303
- DBUnitTests class, 299–300
- Debuggers, 392–394
- Decision coverage, 136
- Decorator pattern, 170
- Decoupling in enterprise testing, 162
- Default reports, 355–358
- <define> tag, 446
- Delegate pattern, 178
- Delegation, 8
- Dependencies
  - CI servers, 321
  - enterprise testing, 177
  - Guice, 282–284, 291–293
  - JUnit tests, 6–7
  - logging, 396
- Dependency Injection pattern, 13
  - EJB3, 236
  - and JNDI, 190, 210
  - Spring framework, 271–272
- Dependency stubs, 246

- Dependent testing, 103–104
  - dependent code, 104–105
  - groups vs. methods, 106–110
  - with TestNG, 105–106
  - and threads, 110–112
- dependsOnGroups attribute, 106, 425
  - annotation transformers, 355
  - vs. dependsOnMethods, 106–110
  - @Test, 109–110, 427
- dependsOnMethods attribute, 112, 425
  - Competing Consumers pattern, 185
  - vs. dependsOnGroups, 106–110
  - @Test, 427
- Deprecated annotations, 419
- Derby database, 227
- description attribute, 425
  - annotation transformers, 354
  - @Test, 427
- Design for code coverage, 149
- Design for testability, 8
  - design patterns, 9–10
  - extreme encapsulation, 14–16
  - object-oriented programming and
    - encapsulation, 8–9
  - problems, 10–11
  - recommendations, 16
  - static methods and fields, 11–14
- Design patterns, 9–10
  - asynchronous testing, 67–71
  - code coverage. *See* Code coverage
  - Data-Driven Testing. *See* Data-Driven Testing
  - dependent testing. *See* Dependent testing
  - factories, 34–39
  - inheritance and annotation scopes, 113–119
  - mocks and stubs. *See* Mock objects; Stub objects
  - multithreaded code, 71–72
    - concurrent running, 79–82
    - concurrent testing, 72–78
    - parallel bit, 82–83
  - performance testing, 83–89
  - test groups. *See* Test groups
  - testing for failures. *See* Failure testing
- Destinations in Competing Consumers pattern, 182
- @DirtiesContext annotation, 279
- DOM API, 253
- dom4j API, 263–264
- Don't Repeat Yourself (DRY) principle, 30, 113, 164
- Driver class, 168
- DriverManager class, 211–212, 214
- DRY (Don't Repeat Yourself) principle, 30, 113, 164
- EasyMock library, 96–99, 102
- Eclipse plug-in, 401
  - installing, 401–404
  - JUnit migration, 453
  - JUnit test conversions, 410–411
  - launch configuration, 404–409
  - preferences, 410
- EE testing. *See* Java EE testing
- EJB entity beans (JPA), 210
- EJB3. *See* Enterprise Java Beans 3.0 (EJB3)
- EJBGen tool, 347
- Email notifications, 321
- EmailableReporter reporter, 204–205
- Embedded containers for servlets, 257–260
- Embedded databases in enterprise testing, 166–167
- EMMA tool, 142–144
- enabled attribute, 425
  - annotation transformers, 353
  - @Test, 427
- Encapsulated attributes, 36
- Encapsulation, 8–9
  - extreme, 14–16
  - vs. testing, 391–392
- Enterprise Java Beans 3.0 (EJB3), 236–237
  - full containers, 244–246
  - Message-Driven Beans, 237–239
  - Session beans, 240–243
  - Spring containers, 243–244
- Enterprise testing, 153–154
  - Competing Consumers pattern, 182–186
  - current approach, 156–157
  - error handling, 172–175
  - example, 157–160
  - implementation, 160–161, 178–181
  - in-container components, 177–178, 193
  - refactoring, 186–193
  - scenario, 154–157
  - setup issues, 166–171
  - for success, 161–163
  - test data for, 163–165
  - unit tests, 175–177

- EntityManager class, 230
- EntityManagerFactory class, 230
- Error handling, 172–175
- Error reporting, 24–25
- Exception class, 25
- Exceptions,
  - design patterns, 25–27
  - JUnit migration, 467–468
  - JUnit tests, 4–5
  - overview, 378–382
- exclude attribute, 441
- Exclusion in code coverage, 137
- ExpectedAttribute class, 36
- @ExpectedException annotation, 278
- expectedExceptions attribute, 28–32, 427
- External APIs for mocks, 101
- Extreme encapsulation, 14–16
- Extreme Programming, 385, 397
  
- Factories
  - vs. Data Providers, 62–63
  - design patterns, 34–39
- @Factory annotation, 35–38, 426
- FadedIcon class, 317
- FailedReporter reporter, 204
- Failure testing, 23–24
  - error handling, 27–31
  - error reporting, 24–25
  - exceptions, 25–27
  - expectedExceptions, 31–32
  - testng-failed.xml, 32–34
- File filters, Java EE testing, 203
- Filters
  - ActiveMQ, 224
  - Java EE testing, 203
  - Servlet, 193
- final scope, 392
- Flickr application, 120
- Functional tests, 154
  
- Gamma, Erich, 2, 9
- Gang of Four, 9
- Global logging, 396
- Gmail application, 120
- Goals
  - enterprise testing, 159–160
  - mocks and stubs, 100
  
- Graphics code, 316–320
- Green bars in code coverage, 147
- Groups
  - in dependent testing, 106–110
  - Eclipse for, 406
  - Guice test dependencies, 291–293
  - IntelliJ IDEA tests, 415
  - test. *See* Test groups
  - TestNG framework, 20–21
- groups attribute, 425
  - annotation transformers, 355
  - @Test, 120–121, 427
- groups option in JUnitConverter, 450
- <groups> tag, 446
- groups variable, 339
- Guice, 280
  - benefits, 280–282
  - configuration, 286–290
  - dependencies, 291–293
  - dependency scenario, 282–284
  - injecting configuration, 293–295
  - object factory, 284–286
  - tests, 290–291
- GuiceModuleObjectFactory class, 291–292
- GuiceObjectFactory class, 289
- Helm, Richard, 9
- Hibernate tool, 210, 225, 234
- Hierarchies, mock object, 102
- Historical coverage tracking, 137
- History
  - annotation, 346–347
  - code coverage, 150
- Hooks, 15
- HtmlUnit, 303–304
  - complex example, 308–310
  - configuration, 304–305
  - usage, 305–308
- HTTP requests, 261
- HttpTester class, 262
- Hudson product, 321–322
- IAnnotationTransformer interface, 349
- Idempotent tests, 166
- IDEs (Integrated Development Environments)
  - code coverage, 136–137
  - Eclipse. *See* Eclipse plug-in
  - IntelliJ IDEA. *See* IntelliJ IDEA plug-in
- IMethodSelector interface, 341, 346

- Immutable state, 382–383
- In-container testing
  - enterprise, 177–178, 193
  - Java EE, 200
    - identifying tests, 201–203
    - registering results listeners, 204–206
    - registering tests, 203–204
    - test environment, 200–201
  - servlets, 255
- In-memory invocation, 260–262
- include attribute, 441
- Incremental JUnit migration, 455–456
- Information hiding, exceptions from, 380
- Inheritance
  - annotation, 113–116
    - pitfalls, 116–119
- inheritGroups attribute, 425
- InitialContext class, 207–209, 212
- InitialContextFactory class, 208
- Initialization
  - in enterprise test setup, 167–168
  - with extreme encapsulation, 14–16
  - JUnit, 463
  - lazy, 55–56
- InitializationStrategy class, 16
- @Inject annotation, 289
- Injection
  - dependency. *See* Dependency Injection pattern
  - Guice, 293–295
  - Message-Driven Beans, 239
  - resource, 230–231
- Injector class, 287–288
- Inline setup example, 301–303
- Inspecting variables, 393
- Instrumenting classes in EMMA, 142–143
- Integrated Development Environments (IDEs)
  - code coverage, 136–137
  - Eclipse. *See* Eclipse plug-in
  - IntelliJ IDEA. *See* IntelliJ IDEA plug-in
- Integration, 154, 269–270
  - continuous, 320–322
  - DbUnit
    - configuration, 295–297
    - features, 295
    - usage, 297–299
    - verifying results, 299–303
  - enterprise testing, 157
  - Guice. *See* Guice
  - HtmlUnit, 303–304
    - complex example, 308–310
    - configuration, 304–305
    - usage, 305–308
  - painting code, 316–320
  - Selenium, 310–311
  - Spring framework. *See* Spring framework
  - Swing UI testing, 312–316
- IntelliJ IDEA plug-in, 411
  - installing, 411–412
  - JUnit migration, 454–455
  - refactorings, 419
  - running tests, 412–419
- Interactive execution in BeanShell, 339–341
- Interfaces for mocks and stubs, 99
- Invalid data in enterprise testing, 173–175
- Inversion of Control (IoC) technique, 13
  - benefits, 270
  - container, 96, 210
- invocationCount attribute, 278
  - annotation transformers, 354
  - Competing Consumers pattern, 185–186
  - multithreaded code, 73, 75–76
  - @Test, 427
- InvocationTransformer class, 351
- IObjectFactory interface, 284
- IoC (Inversion of Control) technique, 13
  - benefits, 270
  - container, 96, 210
- IReporter interface, 364–366
- Isolation levels for rollbacks, 168
- Iterator interface, 55–56, 64–65
- ITest interface, 38–39, 349–350, 360
- ITestContext parameter, 52–54
- ITestListener interface, 326, 361–366
- ITestNGMethod interface, 327
- ITestResult interface, 327
- Java API for XML Web Services (JAX-WS), 246–247
  - recording requests, 248
  - remote services tests, 254–255
  - service test, 251–253
  - test environment, 248–251
  - XPath testing, 253–254
- Java Database Connectivity (JDBC), 210–212
  - c3p0 library, 212–213
  - Commons DBCP library, 213
  - Spring framework, 213–214

- Java EE testing, 197–198
  - EJB3. *See* Enterprise Java Beans 3.0 (EJB3)
  - in-container. *See* In-container testing
  - JAX-WS. *See* Java API for XML Web Services (JAX-WS)
  - JDBC, 210–214
  - JMS, 219
    - ActiveMQ for, 221–225
    - sender/receiver tests, 219–221
  - JNDI. *See* Java Naming and Directory Interface (JNDI)
  - JPA. *See* Java Persistence API (JPA)
  - JTA, 215–218
  - servlets. *See* Servlets
  - XML, 262–266
- Java Messaging Service (JMS), 219
  - ActiveMQ for, 221–225
  - sender/receiver tests, 219–221
- Java Naming and Directory Interface (JNDI), 207
  - avoiding, 210
  - bootstrapping, 207–209
  - JMS in, 220–221
  - in JTA, 216
  - Spring setup, 209–210
- Java Open Transaction Manager (JOTM), 217
- Java Persistence API (JPA), 210, 225–227
  - container simulation, 230–236
  - database, 227
  - providers, 227–229
  - writing tests, 229–230
- Java Transaction API (JTA), 215–218, 228
- JavaDB, 227
- javadoc option for JUnitConverter, 450–451
- Javadocs, 421
  - annotations, 106, 423–428
  - JDK 1.4 and JDK 5, 421–423
  - TestNG class, 428–432
  - XML API, 432–433
- JAX-WS. *See* Java API for XML Web Services (JAX-WS)
- JDBC (Java Database Connectivity), 210–212
  - c3p0 library, 212–213
  - Commons DBCP library, 213
  - Spring framework, 213–214
- JdbcBasedDBTestCase class, 299
- JDK 1.4 and JDK 5 support, 421–423
- JDOM API, 253
- Jetty container, 257–258, 260
- jMock library, 96, 98–99, 102
- JMS (Java Messaging Service), 219
  - ActiveMQ for, 221–225
  - sender/receiver tests, 219–221
- JMSMessage class, 161
- JNDI (Java Naming and Directory Interface), 207
  - avoiding, 210
  - bootstrapping, 207–209
  - JMS in, 220–221
  - in JTA, 216
  - Spring setup, 209–210
- JOTM (Java Open Transaction Manager), 217
- JPA (Java Persistence API), 210, 225–227
  - container simulation, 230–236
  - database, 227
  - providers, 227–229
  - writing tests, 229–230
- JpaTransactionManager class, 234
- JTA (Java Transaction API), 215–218, 228
- junit attribute, 445
- JUnit migration, 449
  - code conversion, 456–457
    - AllTests pattern, 463–467
    - assertions, 457–458
    - class-wide initialization, 463
    - parameterized test case pattern, 469
    - single tests, 458–461
    - state maintenance, 461–462
    - suite-wide initialization, 463
    - testing exceptions, 467–468
  - Eclipse, 453
  - incremental, 455–456
  - IntelliJ IDEA, 419, 454–455
  - JUnitConverter, 449–453
- JUnit mode, 455–456
- JUnit tests, 1–2
  - base classes, 4
  - configuration methods, 6
  - converting, 410–411
  - dependencies, 6–7
  - exceptions, 4–5
  - JUnit 3 tests, 42–43
  - JUnit 4 tests, 7
  - parameters, 4
  - real-world testing, 6
  - running, 5
  - stateful classes, 3–4
  - summary, 7

- JUnitConverter tool, 449
  - from ant, 452–453
  - command line, 449–452
- JUnitReport plug-in, 361
- JUnitTest class, 450–452, 461–462
  
- language attribute, 444
- Launch configuration in Eclipse, 404–409
- Lazy Data Providers, 54–59
- Lazy initialization, 55–56
- Libraries
  - c3p0, 212–213
  - mock, 93–94, 96–99
  - Xerces, 380–381
- Lifecycle methods in Session beans, 242–243
- Listeners
  - registering, 204–206
  - reports, 361–366
- LoadTransformer class, 350
- Log and rethrow exception handling, 380
- Logging best practices, 394–397
- LoggingObjectFactory class, 285
- login servlet testing, 187–193
- LoginController class, 390
- LoginManager class, 189–192
- Luntbuild product, 321
  
- Mailer interface, 90–92, 282
- MailerStub class, 92
- Maintenance overhead for mocks, 102
- Manual rollbacks, 168
- Manual setup in EJB3, 246
- MapSession class, 249
- Merging instrumentation data in EMMA, 142
- Message-Driven Beans (MDBs), 177–178, 237–239
- MessageContext class, 251
- MessageHandler class, 237–239
- MessageHandlerTests class, 238
- MessageProcessor class, 183
- Messages
  - ActiveMQ, 224
  - Competing Consumers pattern, 182–183
- Method parameters for Data Providers, 50–52
- <method-selector> tag, 331, 443
- <method-selectors> tag, 443
- method variable, 339
- Methodology for enterprise testing, 155–156
  
- Methods
  - BeanShell, 336
  - callback, 68–69
  - coverage, 135
  - dependent testing, 106–110
  - Eclipse for, 407
  - IntelliJ IDEA tests, 416
  - private, 388–391
  - scope, 391–392
  - selectors, 341–346
  - <methods> tag, 447
  - MethodSelector class, 343
  - Metrics for code coverage, 134–136
  - Microdesign, 385–386
  - Migration, JUnit. *See* JUnit migration
  - Mock objects, 90
    - designing for, 95–96
    - enterprise testing, 161–162
    - examples, 91–94
    - external APIs, 101
    - libraries, 93–94, 96–99
    - names, 94–95
    - pitfalls, 100–102
    - servlets, 255–256
    - strategies, 99–100
    - vs. stubs, 90–95
  - Moore’s Law, 71
  - Multithreaded code, 71–72
    - concurrent running, 79–82
    - concurrent testing, 72–78
    - parallel bit, 82–83
  - Mutable state, 383–385
  - Mutable static fields, 11
  - MyReporter class, 365–366
  - MyTimeOutTransformer class, 328–329
- name attribute
  - <classes>, 447
  - @DataProvider, 426
  - <parameter>, 442
  - <selector-class>, 443
  - <suite>, 438
  - <test>, 445
- Names
  - mocks and stubs, 94–95
  - test groups, 129–132
- Nest and throw exception handling, 380–381
- nogui parameter, 11–12



- @NotTransactional annotation, 279
- NullPointerException class, 25, 160, 172–173
- O complexity, 86
- Object factory, 284–286
- Object-oriented programming (OOP), 8–9
- Object-relational mapping (ORM) tools, 210
- Offline mode in EMMA, 142
- Online mode in EMMA, 142
- OOP (object-oriented programming), 8–9
- OpenJPA, 225, 234
- ORM (object-relational mapping) tools, 210
- Orthogonal test groups, 131
- Out-of-container Java EE testing, 198–199
- Output formats in Clover, 139
- Overhead for mocks, 102
- overwrite option for JUnitConverter, 450
- <package> tag, 331, 441–442
- Packages for test groups, 119–120
- <packages> tag, 441–442
- Painting code, 316–320
- Pair programming, 398
- parallel attribute
  - <suite>, 438–440
  - <test>, 445
- Parallel bit in multithreaded code, 82–83
- Parallel processing, 71–72
  - concurrent running, 79–82
  - concurrent testing, 72–78
  - parallel bit, 82–83
- <parameter> tag, 441–442
- Parameterized Test Case pattern, 42–43, 469
- Parameters
  - annotations, 18
  - data-driven testing, 42–50
  - @DataProvider annotation, 47–59
  - Eclipse, 408
  - JUnit tests, 4
  - testng.xml, 44–46
- @Parameters annotation, 356, 426
- ParameterTest class, 19
- Participants in enterprise testing, 155
- Patterns, design. *See* Design patterns
- Percentages in code coverage, 148–149
- Performance testing, 83–87
- Persistence class, 232
- persistence.xml file, 227–229, 231–232, 234
- PersistenceAnnotationBeanPostProcessor bean, 236
- PersistenceUnitInfo class, 232
- Person class, 225–226
- PersonManager class, 234–236, 240
- PersonManagerTests class, 235
- PictureTest class, 35, 37, 39
- Pitchfork project, 243–246
- PlatformTransactionManager class, 216, 276
- Plug-ins
  - Clover, 138–141
  - Eclipse. *See* Eclipse plug-in
  - IntelliJ IDEA. *See* IntelliJ IDEA plug-in
  - JUnitReport, 361
  - Report, 360
- POJO-based objects, 225–226
- Polymorphism, 8
- Poolable objects, 212
- @PostConstruct annotation, 241
- @PostCreate annotation, 244
- @PreDestroy annotation, 244
- priority attribute, 443
- Private methods, 388–391
- PropertiesBasedJdbcDatabaseTester class, 296
- Providers
  - Data. *See* Data Providers
  - JPA, 227–229
- QA cycles in enterprise testing, 156
- Queues
  - ActiveMQ, 223–225
  - Competing Consumers pattern, 182
- Quick Fix command, 410–411, 453
- QuickSelectedTest class, 370
- quiet option for JUnitConverter, 450
- Raw HTTP requests, 261
- Real-world testing, 6
- Red bars in code coverage, 147
- Refactorings
  - enterprise testing, 186–193
  - IntelliJ IDEA, 419
  - servlets, 257
- Reflection in Message-Driven Beans, 239
- Registering
  - Java EE tests, 203–204
  - results listeners, 204–206
  - service classes, 252
- Regression testing, 157
- Regular expressions, 123
- Relative performance testing, 86–88
- Remote services tests, 254–255

- @Repeat annotation, 278
- Reporter class, 360–361
- Reporter Output window, 359
- Reports
  - Clover, 141
  - code coverage, 137, 148
  - default, 355–358
  - EMMA, 142–144
  - error, 24–25
  - formats, 137
  - navigation, 137
  - Report API, 360–366
  - Reporter API, 360
  - Test Classes view, 358–359
  - TestNG, 355
- Requests
  - JAX-WS, 248
  - servlets, 261–262
- Resource injection, 230–231
- Resource Locator pattern, 210
- Response time in algorithm complexity, 84–85
- Results listeners, registering, 204–206
- Return codes for reporting errors, 24
- Robertson, Jacob, 63
- Robot class, 312, 315
- Rollbacks in enterprise test setup, 168–170
- <run> tag, 446
- Runnable class, 468
- Running
  - concurrent, 79–82
  - JUnit tests, 5
  - test groups, 125–126
- Runtime
  - design pattern exceptions, 25–27
  - Eclipse parameters, 408
  - test groups, 122–125
- RuntimeException class, 25
  
- Scalability in big O notation, 85
- ScheduledDefaultValue class, 372
- ScheduledInTheFuture class, 372
- ScheduledInThePast class, 372
- ScheduledInvalidDate class, 372
- @ScheduledTest annotation, 368–369
- ScheduledTestSelector class, 369–370
- Scopes
  - annotation, 113–119
  - methods, 391–392
  - testng.xml, 437
- <script> tag, 443
- <selector-class> tag, 443
- Selenium RC framework, 310–311
- Sender/receiver tests in JMS, 219–221
- sequential attribute, 428
- Servers
  - CI, 320–321
  - embedded containers, 257–258
- Service Locator pattern
  - EJB3, 236
  - JNDI, 95, 190, 210
- Service Provider Interface (SPI), 208
- Service test, 251–253
- ServiceLocator pattern, 214
- Servlet Filters, 193
- Servlets, 187–193, 255
  - embedded containers, 257–260
  - in-container testing, 255
  - in-memory invocation, 260–262
  - mock/stub objects, 255–256
  - refactoring, 257
- Session beans, 240–243
- SessionMap class, 191–193
- Setup issues in enterprise testing, 166–171
- Shared states
  - immutable, 382–383
  - mutable, 384–385
- Shortcut syntax for JDK 5 annotations, 423
- SimpleNamingContextBuilder class, 209–210
- Simula language, 8
- SinglePageReporter class, 205
- SingleSpringContextTests class, 273
- Singleton class, 72–73
- Smoke tests, 130
- SOAP protocol, 246
- Source approach in code coverage, 146–147
- SPI (Service Provider Interface), 208
- Spring framework, 213–214, 270–271
  - AbstractAnnotationAwareTransactionalTests, 278–279
  - AbstractDependencyInjectionSpringContextTests, 274–275
  - AbstractSingleSpringContextTests, 273–274
  - AbstractSpringContextTests, 272–273
  - AbstractTransactionalDataSourceSpringContextTests, 278
  - AbstractTransactionalSpringContextTests, 276–278
  - EJB3 layer, 243–244

- features, 271–272
- JPA support, 231–236
- logging for, 395
- spring-mock library, 94
- spring.xml file, 235
- SpringInjectionTests class, 274–275
- SpringTransactionTests class, 276–278
- sreaddir option for JUnitConverter, 450
- Stateful classes, 3–4
- Stateful tests, 382
  - immutable state, 382–383
  - mutable state, 383–385
- Statement coverage, 135
- States
  - ActiveMQ, 223–225
  - maintaining, 461–462
  - shared, 382–385
- Static data in enterprise test setup, 167
- Static methods and fields in design, 11–14
- Static/source instrumentation in code coverage, 146–147
- Statistics for IntelliJ IDEA tests, 419
- StatusHandler class, 308
- Stub objects, 90
  - EJB3, 246
  - enterprise testing, 161–162
  - examples, 91–94
  - vs. mocks, 90–95
  - names, 94–95
  - servlets, 255–256
  - strategies, 99–100
- Success, enterprise testing for, 161–163
- successPercentage attribute
  - annotation transformers, 354
  - @Test, 428
- <suite-file> tag, 442–443
- <suite-files> tag, 442–443
- suite method, 43, 458, 463–466
- <suite> tag, 331, 437–440
- Suite-wide initialization in JUnit migration, 463
- SuiteHTMLReporter reporter, 204–205
- suiteName attribute, 428
- Suites
  - Eclipse for, 406
  - IntelliJ IDEA tests, 414–415
  - TestNG framework, 18–19
- Swallow and throw exception handling, 379
- Sweeney, Tim, 71
- Swing UI, 312
  - configuration, 313–314
  - testing approach, 312–313
  - usage, 314–316
- Switch annotation style, 419
- Syntax for test groups, 120–122
- Synthetic XML files, 126, 333–334
- Tags
  - Javadocs, 421–422
  - XML, 331, 437–447
- TCP Monitor (tcpmon tool), 248
- TDD (test-driven development)
  - application difficulties, 386–387
  - benefits, 388
  - microdesign from, 385–386
- TeamCity, 321–322
- @Test annotation, 19–20, 28–29, 427–428
- Test capture in enterprise testing, 156
- Test Classes view, 358–359
- Test data for enterprise testing, 163–165
- Test-driven development (TDD)
  - application difficulties, 386–387
  - benefits, 388
  - microdesign from, 385–386
- Test groups, 119–120
  - categories, 129–131
  - excluding broken tests, 127–129
  - names, 129–132
  - running, 125–126
  - runtime, 122–125
  - syntax, 120–122
- Test methods in data-driven testing, 42–50
- <test> tag, 331, 437, 444–445
- Testability in EJB3, 236
- TestCase class, 271
- TestHeavyClass class, 14–15
- Testing vs. encapsulation, 391–392
- TestListenerAdapter class, 326–327, 364
- testName attribute, 428
- @testng annotations, 421
- TestNG class, 200–201, 325–326, 428–432
- @testng.data-provider annotation, 425–426
- @testng.factory annotation, 426
- testng-failed.xml file, 32–34, 361, 436, 460–462
- TestNG framework
  - annotations. *See* Annotations
  - BeanShell, 335–341

- TestNG framework (*continued*)
  - dependent testing with, 105–106
  - groups, 20–21
  - integration, 321–322
  - method selectors, 341–346
  - philosophy, 378
  - reports, 355–366
  - synthetic XML files, 333–334
  - TestNG API, 325–331
  - XML API, 331–333
- @testng.parameters annotation, 426
- testng-single.xml file, 460
- @testng.test annotation, 427–428
- testng.xml file, 326, 435
  - vs. Data Providers, 59–60
  - overview, 436–437
  - packages in, 19
  - passing parameters with, 44–46
  - purpose, 21
  - reports, 359
  - sample, 331–332
  - scopes, 437
  - XML tags, 437–447
- testngMethod variable, 339
- TestNGTest class, 462
- TestSuite class, 458
- TextMessage class, 160–161
- thread-count attribute, 79
  - <suite>, 438–439
  - <test>, 445
- Thread1Test class, 439–440
- threadPoolSize attribute
  - annotation transformers, 354
  - parallelism, 185
  - @Test, 73, 75, 77, 428
- Threads
  - and dependent testing, 110–112
  - multithreaded code, 71–72
    - concurrent running, 79–82
    - concurrent testing, 72–78
    - parallel bit, 82–83
- time-out attribute, 445
- Time value, 397–399
- @Timed annotation, 279
- timeOut attribute
  - annotation transformers, 353
  - <suite>, 438, 440
  - @Test, 76–77, 428
- Topics in Competing Consumers pattern, 182
- TopLink, 210, 225, 234
- Total Percentage Coverage (TPC), 139
- TransactionManager class, 216–217
- Transactions
  - enterprise test setup, 168–170
  - Spring framework, 271, 276–278
- Transformers, annotation, 346
  - annotation history, 346–347
  - pros and cons, 348
  - usage, 353–355
  - working with, 348–353
- Unit tests, 129, 154, 157, 175–177
- UnsupportedOperationException class, 208
- Update site for Eclipse, 401–403
- URLConnection class, 259
- URLs for ActiveMQ configuration, 223
- UserAgentTest class, 43
- UserAppFactory class, 288
- UserAppModule class, 287
- UserAppObjectFactory class, 289
- UserDAO interface, 90–92, 282
- UserDAOSTub class, 92
- UserLoginPanel class, 313–315
- UserManagement unit, 228
- UserManager interface, 90–93, 282–283
- UserManagerImpl class, 288
- UserTransaction class, 216–217
- value attribute, 426, 442
- Value of time, 397–399
- Variables
  - BeanShell, 336, 338–339
  - inspecting, 393
  - static, 3
- verbose attribute
  - <suite>, 438
  - <test>, 445
- Web services. *See* Java API for XML Web Services (JAX-WS)
- @WebService annotation, 247
- WebTest class, 116–117
- White-box code coverage testing, 132
- Whitespace in XML, 265
- WrappedConnection class, 170–171
- Wrapping exceptions, 381
- Wsd1Test class, 466
- XDoclet, 347

- Xerces library, 380–381
- XFire services, 247–249
- XML, 262–263
  - API, 331–333, 432–433
  - for DbUbit, 296–297
  - document testing, 253
  - dom4j, 263–264
  - synthetic files, 333–334
  - tags, 331, 437–447
  - XMLUnit, 264–265
- XML over HTTP, 246
- XmlClass class, 331, 433
- XmlFileFilter class, 181
- XmlMethodSelector class, 342
- XmlMethodsSelector class, 331, 433
- XmlPackage class, 331, 433
- XmlSuite class, 331–334, 364, 433
- XmlTest class, 331, 433, 466
- XMLTestCase class, 264
- XMLUnit, 264–265
- XNIException class, 304–305, 381
- XPath
  - JAX-WS, 253–254
  - XML, 264

