

Index

• A •

- abstract class and Composite pattern, 193–194
- Abstract Factory pattern, 260–261
- abstraction
 - decoupling from implementation, 262–263
 - description of, 19–20
- accessing elements inside object.
See Iterator pattern
- Acme class
 - methods, 121
 - objects, creating, 123–124
- AcmeToAcmeAdapter class, 125–126
- Acme class
 - methods, 122
 - objects, creating, 124–125
- AcmeInterface interface, 125–126
- Adapter classes (Java), 144
- Adapter pattern
 - Acme object, creating, 123–124
 - Ace-to-Acme object adapter, creating, 125–127
 - Acme object, creating, 124–125
 - connection problems and, 121–123
 - description of, 11–12, 119
 - inheriting class adapters, 128–134
 - scenario for, 119–121
 - testing adapter, 127–128
- add method
 - Corporate class, 193–194
 - Division class, 182
 - Vector class, 71
- addAssemble method, 162–163, 165
- addGetParts method, 165
- adding new operation to structure of objects, 266–268
- addObserver method, 84
- addStart method, 162–163, 165
- addStop method, 162–163, 165
- addTest method, 162–163, 165
- Alexander, Christopher (*A Pattern Language: Towns, Buildings, Construction*), 282
- algorithm
 - adding hook to, 158–160
 - client code and, 161–164
 - creating, 29–30
 - description of, 29
 - redefining steps in, 150
 - selecting at runtime, 33–35
 - storing, 30
 - using, 30–33
- ApartmentRentedState class, 221–222
- Applicability section of pattern
 - catalog, 285
- Application object, 89
- Archiver class, 74, 82
- array, collections and, 181
- ArrayList object
 - Builder pattern and, 166–167, 169
 - Composite pattern and, 198–199
 - iterators and, 188, 190
- AsiaServer class, 237
- assemble method, 146, 152
- automat proxy
 - creating, 228–230
 - testing, 230–231
- automat server, creating, 225–228

AutomatInterface interface, 214–218
 AutomatProxy class, 229
 AutomatServer class, 226–227
 AutomotivеRobot class, 153–154
 AutomotivеRobotBuilder object, 163

• B •

Boss class, 75, 83
 Bridge pattern, 262–263
 BufferedInputStream object, 49
 BufferedReader object, 227
 Builder pattern
 buildable robot, creating, 168–172
 client rules and, 161–164
 description of, 146, 161
 letting client build robot, 165–168
 Template Method pattern compared to,
 161–162, 164
 testing, 172–175
 button, creating, 69

• C •

CD wrapper, 47
 Chain of Responsibility pattern
 chainable objects, creating, 87–89
 description of, 66, 86–87
 help interface, creating, 87
 Help system, testing, 89–90
 loose coupling and, 77
 chainable objects, creating, 87–89
 change, handling
 composition and, 23–27
 databases and Observer pattern, 66–68
 design patterns and, 41
 “has-a” and, 27–28
 reusing parts of code and, 27–28, 52
 checkApplication method, 211
 Checkbox class, 130

CheckboxAdapter object, 132–133
 Checkboxes object, 131
 Circular Buffer pattern, 269–274
 class loader, multiple, using, 106
 classes
 abstract, and Composite pattern,
 193–194
 abstract Connection, creating, 54–55
 Ace, 121, 123–124
 AceToAcmeAdapter, 125–126
 Acme, 122, 124–125
 Adapter (Java), 144
 ApartmentRentedState, 221–222
 Archiver, 74, 82
 AsiaServer, 237
 AutomatProxy, 229
 AutomatServer, 226–227
 AutomotivеRobot, 153–154
 Boss, 75, 83
 Checkbox, 130
 Client, 74–75, 83
 Component, 275
 ComponentDecorator, 45–46
 Computer, 41, 45
 concrete Connection, creating, 55–56
 ConnectionFactory, 59, 60
 CookieHookRobot, 159–160
 CookieRobot, 154–155
 CookieRobotBuildable, 169–172
 CookieRobotBuilder, 162–163,
 165–168
 Corporate, 193–194
 Database, 72, 76, 94, 95–96
 DatabaseThreaded.java, 103–104
 Decorator pattern and, 40, 43–44
 DifficultProduct, 137–140, 141
 Division, 181–183, 196–198
 DivisionIterator, 197–198
 EuroServer, 238
 final, 58

- FirstFactory, 53–54, 58
- FormulaOne, 24, 31
- FrontEnd, 88
- FullyRentedState, 222
- getApplicationState, 220–221
- Graphics and Graphics2D, 95
- Helicopter, 25, 27, 32
- improving behavior of, 134
- interface between objects and, fixing, 121–123, 128–134
- IntermediateLayer, 89
- Invoker, 246–247
- Java file system, 49
- Java window, 156
- Jet, 25–26, 32
- JFrame, 156–157
- Mediator, 254
- MySQLConnection, 56
- Observable, 79–81, 84–85
- OracleConnection, 50, 55–56
- PipedInputStream and PipedOutputStream, 274
- RealJet.java, 34–35
- RebootCommand, 241, 245
- RentalMethods, 209–213
- Robot, 146–148
- RobotHookTemplate, 158–159
- RobotTemplate, 152–153
- RunDiagnosticsCommand, 240, 246
- ScheduledThreadPoolExecutor, 278
- secure Connection, creating, 61–62
- SecureFactory, 60
- Shape, 20, 22
- ShutDownCommand, 239–240, 244
- SimpleProductFacade, 140–142
- single responsibility and, 179
- spreading out handling of changeable task over generations of, 26
- SqlServerConnection, 56
- StartTheRace.java, 32
- StreetRacer, 24, 31
- Student, 109–110
- StudentThreaded, 113–114
- testing, 48–50
- ThreadPoolExecutor, 278
- USServer, 238–239
- Vector, 71
- Vehicle, 23, 30–31
- VP, 180–181, 194–195
- VPIterator, 194–195
- WaitingState, 218–220
- XMLReaderFactory, 58
- Client class, 74–75, 83
- cloning object, 261–262
- code
 - closing for modification, 41–42
 - extracting for specific task, 35–38
 - opening for extension, 41–42
 - reusing parts of to handle change, 27–28, 52
 - segmenting into states, 209
 - wrapper, 43
- Collaborations section of pattern catalog, 287
- Collection interface, 189–190
- collections. *See also* Composite pattern;
Iterator pattern
 - array and, 181
 - iterators and, 178
- Command interface, 239, 244
- Command pattern
 - commands, creating, 239–241
 - description of, 234–236
 - invoker, creating, 241–242
 - receiver objects, creating, 236–239
 - testing, 242–244
 - undo method and, 244–249
- compartmentalizing code, 209
- Component class, 275
- ComponentDecorator class, 45

- Composite pattern
 - abstract classes and, 193–194
 - branches, creating, 196–198
 - complex organization, building, 198–200
 - description of, 178, 191–193
 - leaves, creating, 194–196
 - Node interface and, 203–206
 - testing, 200–203
 - composition
 - inheritance compared to, 23–27
 - object adapters and, 125
 - Computer class, 41, 45
 - connecting objects in chain of notification. *See* Chain of Responsibility pattern
 - connecting to remote object. *See* Proxy pattern
 - Connection class, creating
 - abstract, 54–55
 - concrete, 55–56
 - secure, 61–62
 - connection problems, fixing. *See* Adapter pattern
 - ConnectionFactory class, 59, 60
 - Consequences section of pattern catalog, 287
 - constructor
 - private, 95
 - ServerSocket, 225, 228
 - controller, 278
 - CookieHookRobot class, 159–160
 - CookieRobot class, 154–155
 - CookieRobotBuildable class, 169–172
 - CookieRobotBuilder class, 162–163, 165–168
 - copying object, 261–262
 - Corporate class, 193–194
 - createConnection method, 52, 53, 59, 60
 - createImage method, 275
 - Crisis Center example
 - commands, creating, 239–241
 - description of, 233–235
 - encapsulation and, 235–236
 - invoker, creating, 241–242
 - receiver objects, creating, 236–239
 - testing commands, 242–244
 - toolkit, 235
- D ●
- Database class
 - Observer pattern and, 72, 76
 - Singleton pattern and, 94, 95–96
 - testing Observable code and, 85
 - database connection object, building, 50–52
 - Database object
 - creating, 75–76
 - Observable object and, 78
 - update method and, 82
 - DatabaseThreaded.java class, 103–104
 - Decorator pattern
 - additional wrappers, adding, 47–48
 - concrete wrapper, adding, 46–47
 - core component, creating, 45
 - decorator, creating, 45
 - description of, 40, 42–44
 - testing, 48–50
 - decoupling abstraction from implementation, 262–263
 - description method, 41–42, 43–45, 46–47, 50
 - design pattern. *See also specific patterns*
 - creating, 282
 - description of, 1, 8
 - finding, 8–9

Design Patterns: Elements of Reusable Object-Oriented Software (GoF book, Gamma, Helm, Johnson, and Vlissides), 1, 9

DifficultProduct class,
137–140, 141

Disk wrapper, 46–47

dispenseKeys method, 212, 221–222

display method, 204–206

Division class, 181–183, 196–198

DivisionIterator class, 197–198

document node, 203

documents, different types of,
creating, 164

Double Buffer pattern, 274–277

draw method, 20

drawGraphics method, 275–276

drawImage method, 276

● E ●

editRecord method

Database class, 72, 76, 85, 94

Observable class, 80

encapsulation

of code components for Veto pattern,
283–284

Command pattern and, 235–236

description of, 20

effective, 136

global objects and, 107

of interaction between objects,
250–251

loose coupling and, 77

using objects to encapsulate state,
213–218

EuroServer class, 238

event listener

Observer pattern compared to, 68

registering observer and, 14–15

execute method, 236, 240, 241, 245

Exit page, 254

extending functionality of class and

Decorator pattern, 40

extension, opening code for, 41–42

extracting code for specific task, and

Strategy pattern, 35–38

extracting generic template class from

existing object. *See* Flyweight

pattern

● F ●

Facade pattern

description of, 119, 135–136

difficult object, dealing with, 137–140

Mediator pattern compared to, 251

scenario for, 134–135

simplifying facade, creating, 140–142

testing facade, 143–144

Factory Method pattern. *See also*

Factory pattern

deciding when to use, 58–59

using, 59–63

factory object

abstract Connection class, creating,
54–55

abstract, creating, 59

building, 52–54

concrete Connection class, creating,
55–56

concrete, creating, 60–61

secure Connection class, creating,
61–62

testing, 56–59, 62–63

factory of factories, creating, 260–261

Factory pattern. *See also* Factory Method pattern; factory object
 Builder pattern compared to, 164
 description of, 40
 formal GoF, 58–59
 new object and, 50–52
 XML parser class and, 9

FilteredInputStream object, 49

final class, 58

finding design pattern, 8–9

Finkelstein, Ellen (*Jakarta Struts For Dummies*), 279

FirstFactory class
 creating, 53–54
 rewriting, 58

Flyweight pattern
 description of, 106–109
 drawbacks to, 115
 Singleton pattern compared to, 93
 student, creating, 109–110
 testing, 110–112
 threading and, 112–115

forces, 282

for/in statement, 190–191

FormulaOne class, 24, 31

FrontEnd class, 88

FullyRentedState class, 222

functionality of class, extending, and
 Decorator pattern, 40, 43–44

• G •

Gamma, Erich (*Design Patterns: Elements of Reusable Object-Oriented Software*), 1, 9

Gang of Four (GoF) book, (*Design Patterns: Elements of Reusable Object-Oriented Software*) (Gamma, Helm, Johnson, and Vlissides), 1, 9

getApplication method, 210–211

getCount method, 215

getFirstName method, 122, 124–125, 126–127

getHelp method, 87, 89, 90

getInstance method
 Flyweight pattern and, 113
 Singleton pattern and, 96, 97, 98
 stripping all object-creation code from, 103–106
 synchronized keyword and, 99–100
 synchronizing, 102

getLastName method, 122, 124–125, 126–127

getName method
 Ace class, 121, 123, 126
 AutomotiveRobot class, 154
 Database class, 94
 DifficultProduct class, 139, 142
 VP class, 180

getParts method, 152

getRobot method, 163, 165, 167

getStanding method, 109–110

getState method, 130–131

getter methods, 81

global object, as singleton, 107

go here method, 147

go method
 algorithms and, 29, 31
 CookieRobotBuildable class, 169
 customizing inherited, 152–153
 Helicopter class, 27
 overriding, 26
 RobotBuildable interface, 168–169
 Vehicle class, 23

GoAlgorithm interface, 29

GoByDrivingAlgorithm interface, 29, 31, 34

GoByFlyingAlgorithm interface, 29, 32, 33

GoByFlyingFast interface, 30, 32

GoF book (*Design Patterns: Elements of Reusable Object-Oriented Software*) (Gamma, Helm, Johnson, and Vlissides), 1, 9
gotApplication method, 219
gotApplicationState class, 220–221
Graphics and Graphics2D classes, 95

• H •

handle method, 254
“has-a” relationship, 28
hasNext method, 179–180, 183, 185
head in circular buffer, 269
Helicopter class, 25, 27, 32
Helm, Richard (*Design Patterns: Elements of Reusable Object-Oriented Software*), 1, 9
help interface, creating, 87
Help system, testing, 89–90
hierarchy, part-whole. *See* Composite pattern
hook
 adding to algorithm, 158–160
 testing, 160–161

• I •

Image object, 276
implementation, decoupling abstraction from, 262–263
Implementation/Sample Code section of pattern catalog, 288–292
improving behavior of class or object, 134
inheritance
 class adapters and, 128–134
 composition compared to, 23–27
 description of, 22–23
 as “is-a” relationship, 27–28

Strategy pattern and, 36
 template base class and, 150–151
InputStream object, 49, 227
instantiating one object. *See* Singleton pattern
Intent section of pattern catalog, 284
interface
 AcmeInterface, 125–126
 algorithm and, 29
 AutomatInterface, 214–218
 Collection, 189–190
 Command, 239, 244
 GoAlgorithm, 29
 GoByDrivingAlgorithm, 29, 31, 34
 GoByFlyingAlgorithm, 29, 32, 33
 GoByFlyingFast, 30, 32
 help, creating, 87
 ItemListener, 131
 java.util.Iterator, 179, 183–186
 Node, 203–206
 between objects and classes, fixing, 121–123, 128–134
 Observer, 70, 78, 82–83
 OOP, and Facade pattern, 135
 Receiver, 236–237
 RobotBuildable, 168–172
 RobotBuilder, 165
 simplifying with facade, 135–136
 Subject, 69–70
IntermediateLayer class, 89
Interpreter pattern, 264
invoker
 creating, 241–242
 description of, 236
Invoker class, 246–247
“is-a” relationship, 28
isSelected method, 130–131, 133
itemChanged method, 133
ItemListener interface, 131
iterator method, 193–194

Iterator pattern

- accessing objects and, 179–181
- Composite pattern and, 193
- description of, 178–179
- for/in statement and, 190–191
- gathering objects into collection, 181–183
- iterating over objects, 186–190
- iterator, creating, 183–186

• J •

Jakarta Struts, 278, 279

Jakarta Struts For Dummies (Robinson and Finkelstein), 279

Java

- Adapter classes, 144
- built-in template methods, 156–158
- circular buffer and, 274
- collections, 178
- double buffering, 274, 275
- facades and, 143
- file system classes, 49
- garbage collector, 106
- iterators, 188–190
- knowledge of, 2
- recycle bins, 278
- ServerSocket constructor, 225, 228
- single inheritance and, 129–130
- window, 156
- WindowAdapter, 144
- Java All-In-One Desk Reference For Dummies* (Doug Lowe), 40
- Java Virtual Machine (JVM), 103
- JavaServer Pages, 278
- java.util.Iterator interface, 179, 183–186
- Jet class, 25–26, 32
- JFrame class, 156–157
- JFrame object, 131

Johnson, Ralph (*Design Patterns: Elements of Reusable Object-Oriented Software*), 1, 9

JVM (Java Virtual Machine), 103

• K •

Known Uses section of pattern catalog, 292–293

• L •

language, programming, putting together own, 264

Law of Demeter, 136

listening for response from automat proxy, 230

loadActions method, 167, 169

loose coupling

- Facade pattern and, 136

- Mediator pattern and, 250–251

- Observer pattern and, 75, 77

Lowe, Doug (*Java All-In-One Desk Reference For Dummies*), 40

• M •

main method, 275

Mediator class, 254

Mediator pattern

- connecting pages and, 254–255

- description of, 10–11, 234, 249–251

- Rutabagas-R-Us site, designing, 251–254

- Rutabagas-R-Us site, testing, 255–256

Memento pattern, 264–266

memory and Circular Buffer pattern, 269–274

methods. *See also specific methods*

- Ace class, 121

- Acme class, 122

- CookieRobotBuilder class, 166–167

- iterators and, 179
- Observable class, 79–80
- state objects and, 218–222
- using to hold state, 209–213

model, 278

Model/View/Controller pattern, 278–279

modification, closing code for, 41–42

Monitor wrapper, 47–48

Motivation section of pattern catalog, 285

multithreading, 93, 99–100

MySQLConnection class, 56

• N •

navigating Web site. *See* Mediator pattern

new object, 40, 50–52

new operator

- Flyweight pattern and, 112–113
- Singleton pattern and, 92, 94–95

next method, 179–180, 183–184

Node interface, 203–206

notification, passing, 13–15, 78

notifying objects, 86–87

notifying observer. *See* Observer pattern

notifyObservers method

- Observable class, 78, 79–80
- Observer pattern and, 69, 70, 72–73

• O •

object. *See also* factory object

- accessing elements inside, 178
- adding new operation to structure of, 266–268
- Application, 89
- ArrayList, 166–167, 169, 188, 190, 198–199
- AutomotiveRobotBuilder, 163
- BufferedInputStream, 49

- BufferedReader, 227
- chainable, creating, 87–89
- CheckboxAdapter, 132–133
- Checkboxes, 131
- cloning, 261–262
- composites of, 28
- copying and customizing, 261–262
- Database, 75–76, 78, 82
- database connection, building, 50–52
- description of, 18–19
- FilteredInputStream, 49
- global, as singleton, 107
- Image, 276
- improving behavior of, 134
- InputStream, 49, 227
- interface between classes and, fixing, 121–123, 128–134
- JFrame, 131
- mass producing, 145–148
- new, 40, 50–52
- notifying, 86–87
- PrintWriter, 225–227, 228–230
- receiver, 236–239
- registry, 93
- remote, connecting to, 12–13
- restoring state of, 264–266
- save-state, 264–266
- singleObject, 96–97
- state, and methods, 218–222
- traverser, 267–268
- using to encapsulate state, 213–218

object, packaging complex commands into single. *See* Command pattern

object-oriented programming (OOP)

- abstraction and, 19–20
- composition and, 23–27, 125
- encapsulation and, 20, 107
- extending, 18–19
- inheritance and, 22–23
- overview of, 9–10
- polymorphism and, 20–22

- Observable class
 - addObserver method, 84
 - extending, 80–81, 85
 - methods of, 79–80
 - testing code, 84–85
 - observer, creating, 73–75
 - Observer interface
 - building, 70
 - description of, 78
 - objects, creating, 82–83
 - Observer pattern
 - description of, 13–15, 66–68
 - in Java, 69
 - loose coupling and, 77
 - observer, creating, 73–75
 - observer interface, creating, 70
 - subject, creating, 70–73
 - subject interface, creating, 69–70
 - testing, 75–77
 - one-to-many dependency and Observer pattern, 66
 - online development, 121
 - OOP (object-oriented programming)
 - abstraction and, 19–20
 - composition and, 23–27, 125
 - encapsulation and, 20, 107
 - extending, 18–19
 - inheritance and, 22–23
 - overview of, 9–10
 - polymorphism and, 20–22
 - OOP interface and Facade pattern, 135
 - OracleConnection class, 50, 55–56
 - overriding
 - paint method, 156–158
 - steps in algorithm, 150
- *p* ●
- packaging complex commands into single object. *See* Command pattern
 - paint method
 - Image object and, 276
 - JFrame class and, 156–158
 - parameterizing clients with different requests, 245
 - Parser.java code, 203–206
 - parsing XML files, 203–206
 - Participants section of pattern catalog, 286
 - passing notification. *See* Observer pattern
 - pattern. *See also specific patterns*
 - creating, 282
 - description of, 1, 8
 - finding, 8–9
 - pattern catalog
 - Applicability section, 285
 - Collaborations section, 287
 - Consequences section, 287
 - Implementation/Sample Code section, 288–292
 - Intent section, 284
 - Known Uses section, 292–293
 - listing pattern in, 293
 - Motivation section, 285
 - Participants section, 286
 - Related Patterns section, 293
 - Structure section, 285–286
 - style, 283
 - A Pattern Language: Towns, Buildings, Construction* (Alexander), 282
 - Patterns Library, 260
 - PipedInputStream and PipedOutputStream classes, 274
 - planning for change, 27–28
 - polymorphism, 20–22
 - Portland Pattern Repository, 260, 293
 - Principle of Least Knowledge, 136
 - print method
 - Composite pattern and, 192
 - Corporate class, 193–194, 199–200
 - VP class, 180
 - PrintWriter object, 225–227, 228–230
 - private constructor, 95
 - Prototype pattern, 261–262
 - proxy, definition of, 13

Proxy pattern

- automat proxy, creating, 228–230
- automat proxy, testing, 230–231
- automat server, creating, 225–228
- description of, 12–13, 208, 224–225

Purchase page, 253

• Q •

queue of commands, 245

• R •

reading item in circular buffer, 270, 272

RealJet.java class, 34–35

RebootCommand class, 241, 245

Receiver interface, 236–237

receiver object, 236–239

Rectangle class, 21, 23

Recycle Bin pattern, 277–278

redefining steps in algorithm, 150

registering

- as observer, 71

- to receive notification, 67

registerObserver method

- Database class, 76

- Observer pattern, 69, 70–71

registry object, 93

Related Patterns section of pattern

- catalog, 293

remote object, connecting to. *See* Proxy pattern

remove method, 179–180, 183, 185–186

removeObserver method, 70, 71

removing object from vector, 71

rental automat example

- methods for state objects and, 218–222

- methods, using to hold state, 209–213

- state objects and, 213–218

- states of, 208–209

- testing, 223

RentalMethods class, 209–213

rentApartment method, 212

repaint method, 276

restoring object state, 264–266

restricting resource use, 93

reusing parts of code to handle change, 27–28, 52

ring buffer, 269

Robinson, Mike (*Jakarta Struts For Dummies*), 279

Robot class, 146–148

RobotBuildable interface, 168–172

RobotBuilder interface, 165

RobotHookTemplate class, 158–159

RobotTemplate class, 152–153

Rule of Three, 282

run method, 241–242, 245

RunDiagnosticsCommand class, 240, 246

runtime, selecting algorithm at, 33–35

Rutabagas-R-Us Inc. Web site example designing, 251–254

- Mediator pattern and, 249–251

- testing, 255–256

• S •

save-state object, 264–266

ScheduledThreadPoolExecutor class, 278

secure Connection class, creating, 61–62

SecureFactory class, 60

selecting algorithm at runtime, 33–35

sending test back and forth between points on Internet. *See* Proxy pattern

separating parts of code that change, 27–28, 52

ServerSocket constructor, 225, 228

setChanged method, 79

setCommand method, 241–242

setCount method, 215

setFirstName method, 122, 124–125, 126–127

setGoAlgorithm method, 30, 35
 setName method
 Ace class, 121, 123
 DifficultProduct class, 142
 Shape class, 20, 22
 sharing data in object, 93
 Shopping page, 252–253
 ShutdownCommand class, 239–240, 244
 SimpleProductFacade class, 140–142
 single responsibility, 179
 singleObject object, creating, 96–97
 Singleton pattern
 alternatives to, 107
 database, creating, 94–97
 description of, 92–93
 Flyweight pattern compared to, 93
 multithreading and, 99–100
 pre-thread solution and, 104–106
 synchronized solution and, 100–102
 testing, 98–99
 threading and, 103–104
 SqlConnection class, 56
 start method, 146, 152
 StartTheRace.java class, 32
 State pattern
 description of, 208–209
 methods, using to hold state, 209–213
 objects, using to encapsulate state, 213–218
 state objects, creating, 218–222
 testing, 223
 stop method, 152
 storing
 algorithm, 30
 item in circular buffer, 269–270, 271
 Strategy pattern, 35–38
 StreetRacer class, 24, 31
 structure of objects, adding new operation to, 266–268
 Structure section of pattern catalog, 285–286
 Student class, 109–110

StudentThreaded class, 113–114
 subclasses
 Factory pattern and, 58
 singleton and, 107
 subject, creating, 70–73
 Subject interface, creating, 69–70
 synchronized keyword, 99–100

● T ●

tail in circular buffer, 269
 task, extracting code for specific, 35–38
 template class, generic, extracting from existing object. *See* Flyweight pattern
 Template Method pattern
 Builder pattern compared to, 161–162, 164
 built-in template methods in Java, 156–158
 description of, 146, 149–150
 hook, adding, 158–160
 robots, creating, 150–155
 testing hook method, 160–161
 testing robot creation, 155–156
 test method, 146, 152
 TestAdapter.java test harness, 127–128
 TestAutomat.java test harness, 223
 TestAutomatProxy.java test harness, 230–231
 TestCircularBuffer.java test harness, 272–274
 TestCommands.java test harness, 242–244, 247–249
 TestConnection.java test harness, 56–59
 TestCorporation.java test harness, 200–203
 TestDivision.java test harness, 186–190

- TestFacade.java test harness, 143–144
 - TestFactory.java test harness, 62–63
 - TestFlyweight.java test harness, 110–112
 - TestFlyweightThreaded.java test harness, 114–115
 - TestHookTemplate.java test harness, 160
 - testing
 - adapter, 127–128
 - automat proxy, 230–231
 - circular buffer, 272–274
 - class, 48–50
 - commands, 242–244
 - Composite pattern, 200–203
 - facade, 143–144
 - factory object, 56–59, 62–63
 - Flyweight pattern, 110–112
 - Help system, 89–90
 - hook method, 160–161
 - iterator, 186–190
 - Observable code, 84–85
 - Observer pattern, 75–77
 - rental automat, 223
 - RentalMethods class, 213
 - Robot class, 147
 - robot creation, 155–156
 - Rutabagas-R-U.s site, 255–256
 - Singleton pattern, 98–99
 - undo method, 247–249
 - Veto pattern, 290–292
 - Test.java test harness, 48–50
 - TestMediator.java test harness, 255–256
 - TestObservable.java test harness, 84–85
 - TestObserver.java test harness, 75–77
 - testOK hook method, 158–160
 - TestRentalMethods.java test harness, 213
 - TestRobotBuilder.java test harness, 172–175
 - TestSingleton.java test harness, 98
 - TestSingletonSynchronized.java test harness, 100–102
 - TestSingletonThreaded.java test harness, 104–105
 - TestTemplate.java test harness, 155–156
 - TestVeto.java test harness, 290–292
 - text node, 203
 - threading
 - Flyweight pattern and, 112–115
 - Singleton pattern and, 99–100
 - ThreadPoolExecutor class, 278
 - traverser object, 267–268
 - tree-like structures, creating, 178, 192
- U •
- undo method
 - Command pattern and, 236, 244–247
 - testing, 247–249
 - unregistering, 68
 - update method
 - Archiver class, 74
 - Database object, 82
 - Java window classes and, 156
 - Observable class, 79–80
 - Observer interface, 70, 78, 82
 - upgrading and Adapter pattern, 11–12
 - USServer class, 238–239
- V •
- Vector class, 71
 - vector, using, 70–71
 - Vehicle class, 23, 30–31

Veto pattern example

Applicability section, 285

Collaborations section, 287

Consequences section, 287

description of, 283–284

Implementation/Sample Code section,
288–292

intent, 284

Known Uses section, 292–293

motivation, 285

Participants section, 286

Related Patterns section, 293

Structure section, 285–286

testing, 290–292

view, 278

Visitor pattern, 266–268

Vlissides, John (*Design Patterns: Elements
of Reusable Object-Oriented Software*),
1, 9

VP class, 180–181, 194–195

VPIterator class, 194–195

• W •

WaitingState class, 218–220

Web site. *See also* Mediator pattern

Jakarta Struts, 279

Patterns Library, 260

Portland Pattern Repository, 293

Welcome page, 251–252, 256

WindowAdapter (Java), 144

wrapper code, 43, 46–48. *See also* Facade
pattern

• X •

XML element, 203

XML nodes, 203

XML parser class, 9

XMLReaderFactory class, 58

