abstract factory design pattern. See also factory design pattern
class diagram, 68
implementing in plain code, 69
when to use, 80
AbstractActionFactory class, 188
AbstractDrinksMachineFactory interface, 69
@AccessTimeout annotation, 45
AccountService EJB, 29–31
ActionFactory class, 188
addUser() method, 44
advices, 102. See also AOP
context, 1040105
default-level interceptors, 106
interceptor execution order, 107–109
AFTER_COMPLETION phase, 148, 149
AFTER_FAILURE phase, 148, 150
AFTER_SUCCESS phase, 148, 149
AKF scale cube, 210
@Alternative annotation, 59, 72
Alur, Deepak, 8
AnnotationLiteral class, 77–78
annotations
  @AccessTimeout, 45
  @Alternative, 59, 72
  @Any, 76–79
  @ApplicationPath, 175
  @AroundInvoke, 102, 104
  @ClearanceSale, 93–94
  @ConcurrencyManagement, 46
  @ConversationScope, 21
  @ConversationScoped, 57
  @Decorator, 61, 89–94
@Delegate, 61, 89–94
@DELETE, 175
@DependsOn, 40–42
@EJB, 53, 54
@Entity, 156, 158
@GeneratedValue, 156, 158
@GET, 175–178
@Id, 156, 158
@Inject, 53–54, 56, 57, 69–76
@InterceptorBinding, 109–110, 112
@Interceptors, 21
@interface, 73, 74
life cycle annotations, 105
@Lock, 43–45
@LockType, 43–45
@LongMessage, 73–74, 77–78
method-level annotations, 103–104
@Model, 60, 191
@MyEvent, 147
@Named, 54–55, 58, 60, 75–76, 190–191
@Observes, 144–150
@Path, 175–178
@PathParam, 177–178
@POST, 175, 178
@PostActivate, 105–106
@PostConstruct, 14, 39, 42, 46, 105
@PreDestroy, 105–106
@PrePassivate, 105–106
@Produces, 69–76, 80
@PUT, 175
@Qualifier, 73
@QueryParam, 178
@Remove, 21
@RequestScope, 21, 190–191
@RequestScoped, 57, 60
annotations (continued)

@Resource, 53, 132
@Schedule, 128–131
@Schedules, 128–131
@Secure, 109–110
@SessionScope, 21
@SessionScopes, 57
@ShortMessage, 73–74, 77–78
@Singleton, 14, 39, 46
@Startup, 14, 40
@Stateful, 31
@Target, 59, 74
@Timeout, 128, 132, 137
@Transient, 158
@WebServiceRef, 53
anti-patterns, 10, 204
lasagna architecture, 204–205
uber class, 204
@Any annotation, 76–79
AOP (aspect-oriented programming), 97–112
implementing in plain code, 100–101
interceptors
CDI interceptors, 109–111
default-level, 106
implementing, 102–105
life cycle annotations, 105–106
order of execution, 107–109
when to use, 111
applets, 7
@ApplicationPath annotation, 175
Application tier, 17
archives, CDI beans, 56–57
@AroundInvoke annotation, 102, 104
aspect-oriented programming. See AOP aspects, 61, 97, 98, 102. See also AOP
async() method, 120
asynchronous beans, 118–120
asynchronous programming, 113–126
asynchronous beans, 118–120
asynchronous servlets, 120–124
flow diagram, 116
implementing
in Java EE, 118–124
in plain code, 116–118
overview, 113–115
when to use, 124–125
asynchronous servlets, 120–124
automatic timers, 130–131
calendar-based attributes, 134–136
implementation, 131

B

Baklava Code, 205
BaseDAO interface, 162, 163
BasicDrinksMachineFactory class, 69
Batch, 16
bean archives, 56–57
bean-managed concurrency, 43
beans.xml file, 39, 61, 91, 110–111
BEFORE_COMPLETION phase, 148–149, 151
behavioral design patterns, 6
Bien, Andy, 205, 206
BlackFridayDiscountDecorator class, 91–94
bleeding-edge technologies, 205
Bloch, Joshua, 37
BlockingQueues, 118
Box, Don, 8
Brooks, Frederick, 209
Business layer (Middle tier), 14, 15, 16–17

cacheable constraint (REST), 168
calendar-based timer attributes, 134–136
callback methods, 130
CDI (Context and Dependency Injection). See also
dependency injection design pattern
vs. EJB, 56
injectable object types, 56
interceptors, 109–111
multiple interface implementations and, 76–79
CDI beans, 20–21, 56–57
archives, 56–57
backing beans for JSF pages, 58
for backing JSF, 58
naming, 38
requirements, 56
scopes, 21, 57–58
stereotypes, 60
Children class, 149
class diagrams, 4–5
  abstract factory pattern, 68
data access object pattern, 154–155
  decorator pattern, 85
da façade pattern, 27
factory pattern, 64–65
observer pattern, 141–142
singleton pattern, 34
classes
AbstractActionFactory, 188
ActionFactory, 188
AnnotationFactory, 77–78
BasicDrinksMachineFactory, 69
BlackFridayDiscountDecorator, 91–94
Children, 149
CoffeeMachine, 66
doubleExtra, 86, 88
DrinksMachine, 66–68
Endpoint, 196–197
EventService, 70–76, 145
FrontController, 187
FutureTask, 117
GourmetDrinksMachineFactory, 69
HelloEndpoint, 196–197
helper classes, 206
ListUserAction, 189–190, 191
MessageFactory, 79
Movie, 156, 157
MySingleton, 35–38
naming, 19
NoCostExtra, 86, 88
Observable, 142–144
Pizza, 86
Plate, 93
PriceDiscountDecorator, 90–94
qualifier classes, 59
RegularExtra, 86, 87
Runnable, 116
ScheduleExpression, 133–137
SoftDrink, 67
SoftDrinksMachine, 67
Table, 89–90
TraceObserver, 145
UserService, 50–52, 53–55, 187
UserServiceFactory, 51
utility classes, 206
@ClearanceSale annotation, 93–94
Client tier, 14–15
client-server constraint (REST), 167
CMP (Container-Managed Persistence), 156
code on demand constraint (REST), 168
coding by convention, 19
CoffeeMachine class, 66
CoffeeType enum, 67–68
cross-cutting concerns, 98, 104, 111
dependency injection, 19
Concurrency Utilities, 16, 17, 18
creational design patterns, 43, 54, 56
Container-Managed Persistence (CMP), 156
creation over configuration, 19
@ConversationScope annotation, 21
@ConversationScoped annotation, 57
Core J2EE Patterns: Best Practices and Design Strategies, 8
createCalendarTimer() method, 133
createIntervalTimer() method, 133
createSingletonActionTimer() method, 133
createTimer() method, 133
creationality design patterns, 6
cross-cutting concerns, 98, 104, 111
Crupi, John, 8
customerService EJB, 29–31

data access object (DAO) design pattern, 65–66, 153–163
class diagram, 154–155
DTOs (data transfer objects), 155–156
implementing in Java EE, 157–162
JPA (Java Persistence API), 156–158
ORM (Object Relational Mapping), 156
overview, 153–154
type-safe implementation, 162
when to use, 163
Data Access tier, 17
Data tier, 18
data transfer objects (DTOs), 155–156
dayOfMonth timer attribute, 134
dayOfWeek timer attribute, 134
debugging, and asynchronous programming, 124
decomposing monolithic applications, 201–202
@Decorator annotation, 61, 89–94
decorator design pattern, 83–96
class diagram, 85
implementing
  in Java EE, 89–94
  in plain code, 86–89
overview, 83–85
when to use, 94–95
@Delegate annotation, 61, 89–94
@DELETE annotation, 175
DELETE method, 171, 172, 173, 174, 178
dependency injection design pattern. See also CDI
  benefits, 50
implementing
  in Java EE, 53–61
  in plain code, 50–53
@Named attribute, 54–55
overview, 20–21, 49–50
@DependsOn annotation, 40–42
design patterns. See also specific patterns
class diagrams as, 4–5
  abstract factory, 68
  data access object, 154–155
  decorator, 85
  façade, 27
  factory, 64–65
  observer, 141–142
  singleton, 34
vs. enterprise patterns, 8–9
group classifications, 6
history of, 5
implementing. See implementing design patterns
  need for, 5
overview, 4–6
for success, 209–211
unnecessary use of, 5–6, 211–212
when to use
  asynchronous programming, 124–125
  data access object, 163
  decorator, 94–95

façade, 31
factory, 80
interceptors, 111
model view controller, 193
observer, 150–151
REST, 180–181
singleton, 45

Design Patterns: Elements of Reusable
  Object-Oriented Software, 5, 6
DI. See dependency injection design pattern
Dijkstra, Edsger, 210
Dispatcher servlet, 186
doFilter() method, 100
doGet() method, 187
double-checked locking, 37
DoubleExtra class, 86, 88
Drink interface, 67
DrinksMachine class, 66–68
DTOs (data transfer objects), 155–156

Eclipse, 211
Effective Java, 37, 38
Ehcache, 45
EIS (Enterprise Information Systems) tier, 14–15, 18
EJB (Enterprise JavaBeans)
  vs. CDI, 56
  component model, 8
  containers, 15, 16, 17
  context, 21
stateful beans, façade pattern, 31
stateless beans, façade pattern, 29–31
@EJB annotation, 53, 54
EL (Expression Language), 16, 17, 58, 184
Endpoint class, 196–197
Enterprise Information Systems (EIS) tier, 14–15, 18
enterprise patterns, 7–10
@Entity annotation, 156, 158
entity beans, 156
EntityManager, 159–160, 161, 163
EventService class, 70–76, 145
execute() method, 188, 192
ExecutorService, 117, 123–124
explicit bean archives, 56–57
Expression Language (EL), 16, 17, 58, 184

façade design pattern, 25–32
  benefits, 28–29
  class diagram, 27
  implementing
    in plain code, 27–29
    with stateful beans, 31
    with stateless beans, 29–31
  overview, 25–27
  when to use, 26, 31
faces-config.xml file, 190
FacesServlet, 190–192
factory design pattern, 63–81
  abstract factory, 68
    class diagram, 68
    implementing in plain code, 69
  factory method, 64
    class diagram, 64–65
    implementing in Java EE, 69–80
    implementing in plain code, 66–68
  overview, 63–64
  when to use, 80
Fielding, Roy, 166, 167
Fork/Join Framework, 118
Fowler, Martin, 49
FrontController class, 187
Future interface, 117
FutureTask class, 117

getTimer method, 133
getTimeRemaining method, 133
getUserList() method, 192
getUsers() method, 176
GlassFish Server Open Source Edition 4.0, 18
GoF (Gang of Four)
GourmetDrinksMachineFactory class, 69

HATEOAS (Hypermedia As The Engine Of Application State), 178–180
Head First Design Patterns, 6, 26, 34, 38, 64, 68, 85, 140
heavilySoiled method, 27–28
HelloEndpoint class, 196–197
helper classes, 206
helper package, 206
Hollywood principle, 140
home.jsp file, 191
hour timer attribute, 134
HTTP
  asynchronous servlets and, 121
  methods, 171–172
  REST and, 166–167
  Hypermedia As The Engine Of Application State (HATEOAS), 178–180

@Id annotation, 156, 158
  implementing design patterns
  in Java EE
    asynchronous programming, 118–124
    data access object, 157–162
    decorator, 89–94
    dependency injection, 53–61
    façade, 29–31
    factory, 69–80
    interceptors, 102–111
    model view controller, 190–192
    observer, 144–151
    singleton, 38–45
    timer service, 130–137
    WebSockets, 196–197
implementing design patterns (continued)
in plain code

abstract factory, 69
aspect-oriented programming, 100–101
asynchronous, 116–118
decorator, 86–89
dependency injection, 50–53
façade, 27–29
factory, 66–68
model view controller, 186–190
observer, 142–144
singleton, 35–38

implicit bean archives, 56–57

IN_PROGRESS phase, 148
info attribute, timer service, 130–131
@Inject annotation, 53–54, 56, 57, 69–76
Integration tier, 18
@InterceptorBinding annotation, 109–110, 112
interceptors, 21–22
CDI interceptors, 109–111
default-level, 106
implementing, 102–105
life cycle annotations, 105–106
order of execution, 107–109
when to use, 111
@Interceptors annotation, 21
@interface annotation, 73, 74
interfaces
AbstractDrinksMachineFactory, 69
BaseDAO, 162, 163
Drink, 67
Future, 117
MessageType, 77–79
MovieDAO, 159–163
multiple implementations of, 76–79
Observer, 141–142, 143–144
Order, 86–89
Product, 89–94
Publisher, 143
TimedObject, 128
UserDataRepository, 51–52, 54–55, 59
InvocationContext parameter, 102, 104
isCalendarTimer method, 133–134
isPersistent method, Timer object, 134

Java
history of, 7–10
servlet filters, 98, 99–101
Java API for RESTful Web Services (JAX-RS), 17, 178
Java API for XML Registries (JAXR), 17, 19
Java API for XML-Based RPC (JAX-RPC), 17, 19
Java API for XML-Based Web Services (JAX-WS), 17, 19
Java Caching System, 45
Java Community Process (JCP), 18
Java EE
component model, 19
core principles, 19
multitier architecture, 14–18
servers, 18
standards, 18
web profile, 18–19
Java EE 7 Tutorial: Part VIII Persistence, 156
Java Message Service (JMS) API, 199
Java Naming and Directory Interface (JNDI), 53, 130
Java Persistence API (JPA), 156–158
Java Specification Requests. See JSRs
Java Transaction API (JTA), 16, 17, 161
JavaServer Faces (JSF), 17, 55, 184
JavaServer Pages (JSP), 16, 17, 184, 185
JAX-RPC (Java API for XML-Based RPC), 17, 19
JAX-RS (Java API for RESTful Web Services), 17, 178
JAX-WS (Java API for XML-Based Web Services), 17, 19
JAXR (Java API for XML Registries), 17, 19
JCP (Java Community Process), 18
JMS (Java Message Service) API, 199
JNDI (Java Naming and Directory Interface), 53, 130
join() method, 117
JPA (Java Persistence API), 156–158
JSF (Java Server Faces), 17, 55, 58, 184
JSP (JavaServer Pages), 16, 17, 184, 185
JSRs (Java Specification Requests), 17, 18
JSR133, 118
JSR166, 114
JSR220, 53
JSR229, 53
JSR250, 53
JSR299, 55
JSR315, 120
JSR330, 53
JTA (Java Transaction API), 16, 17, 161

Lasagna architecture, 204–205
Layered system constraint (REST), 168
Levels, Richardson Maturity Model, 168–169
Life cycle
AsyncListener methods, 122
CDI beans, 57
interceptors, 105–106
WebSockets methods, 196–197
lightSoiled method, 27–28
listuser.jsp file, 189–190
ListItemAction class, 189–190, 191
LoanService EJB, 29–31
@Lock annotation, 43–45
@LockType annotation, 119–120
logger interceptor, 21–22
Logic tier, 17
logSync() method, 120
@LongMessage annotation, 73–74, 77–78
loosely coupled applications, 19

Malks, Dan, 8
ManagedExecutorService, 123–124
ManagedThreadFactory, 122–123
Message-oriented middleware (MOM), 198–199
MessageA bean, 71–76
MessageB bean, 71–76
MessageEvent qualifier, 146
MessageFactory class, 79
MessageType interface, 77–79
method-level annotations, 103–104
microservice architecture, 199–204
benefits, 202–203
costs, 203–204
scalability, 200–201
Middle tier, 14–15, 16–17
minute timer attribute, 134
@Model annotation, 60, 191
model view controller design pattern, 183–193
implementing
in Java EE, 190–192
in plain code, 186–190
overview, 183–185
types, 185–186
when to use, 193
model view presenter (MVP), 186
MOM (message-oriented middleware), 198–199
monolithic architecture, 199–200
month timer attribute, 134
Movie class, 156, 157
MovieDAO interface, 159–163
multitier architecture, 14–18
MVC. See model view controller design pattern MVP (model view presenter), 186
@MyEvent annotation, 147
MySingleton class, 35–38
The Mythical Man-Month, 209–210

Named annotation, 54–55, 58, 60, 75–76, 190–191
naming classes, 19
NoCostExtra class, 86, 88
nonblocking method execution, 113
notifyObservers method, 141, 150
nouns, REST, 166, 167
HTTP verbs and, 169
naming, 170
plural, 171
posts, 173–174
topics, 173–174
uniform interface constraint, 167
users, 172–173
Object Relational Mapping (ORM), 156, 205
object-oriented programming. See OOP
Observable class, 142–144
observer design pattern, 139–152
class diagram, 141–142
implementing
   in Java EE, 144–150
   in plain code, 142–144
overview, 139–141
when to use, 150–151
Observer interface, 141–142, 143–144
@Observes annotation, 144–150
onClose method, 196
onError method, 122, 196
onOpen method, 196–197
onStartAsync method, 122
onTimeout method, 122
OOP (object-oriented programming)
   background, 5, 210
design patterns in, 4
lasagna architecture, 204
uber classes in, 204
Order interface, 86–89
ORM (Object Relational Mapping), 156, 205

@Path annotation, 175–178
@PathParam annotation, 177–178
Paypal.com, 180
Persistence tier, 18
persistence units, 160–161
persistence.xml file, 159–161
Pizza class, 86
Plate class, 93
point-to-point MOM implementations, 198–199
pointcut, 102
@POST annotation, 175, 178
@PostActivate annotation, 105–106
@PostConstruct annotation, 14, 39, 42, 46, 105
POST method, 166, 171–172, 172–173, 178
posts noun, 173–174
@PreDestroy annotation, 105–106
@PrePassivate annotation, 105–106
Presentation tier. See Client tier
PriceDiscountDecorator class, 90–94
@Produces annotation, 69–76, 80
Product interface, 89–94
programmatic timers, 130, 131–134
calendar-based attributes, 134–136
implementation, 132
“Prototyping Patterns for the J2EE Platform”, 8
publish/subscribe MOM implementations, 198–199
Publisher interface, 143
@PUT annotation, 175
@PUT method, 171, 172, 173, 174, 178

@Qualifier annotation, 73
qualifier classes, 59
Quartz, 118, 125, 128
@QueryParam annotation, 178
queues, message-oriented middleware, 198

RadioChannel observer, 143
Rahman, Reza, 204
Reenskaug, Trygve, 184
RegularExtra class, 86, 87
@Remove annotation, 21
@RequestScope annotation, 21, 190–191
@RequestScoped annotation, 57, 60
@Resource annotation, 53, 132
REST, 165–181
costants, 167–168
designing RESTful API, 169–174
HATEOAS, 178–180
implementing RESTful API, 175–178
nouns. See nouns, REST
overview, 165–167
Richardson Maturity Model, 168–169
when to use, 180–181
Richardson Maturity Model, 168–169
rollbacks, observer pattern, 150–151
Runnable class, 116
scalability, 200–201
@Schedule annotation, 128–131
ScheduleExpression class, 133–137
@Schedules annotation, 128–131
scopes, CDI beans, 21, 57–58
Seam, 55
second timer attribute, 134
@Secure annotation, 109–110
Service Oriented Architecture (SOA), 8, 199, 212
servlet filters, 98, 99–101
servlets
asynchronous servlets, 120–124
FacesServlet, 190–192
servlet filters, 98, 99–101
@SessionScope annotation, 21
@SessionScoped annotation, 57
@ShortMessage annotation, 73–74, 77–78
@Singleton annotation, 14, 39, 46
singleton beans, 13–14, 39–42
singleton design pattern, 33–47
class diagram, 34
implementing
in Java EE, 38–45
in plain code, 35–38
overview, 33–34
when to use, 34, 45
sleep() method, 117, 119
Smalltalk, 205, 210
SOA (Service Oriented Architecture), 8, 199, 212
SoftDrink class, 67
SoftDrinksMachine class, 67
Spring, MVC implementation, 186
startAsync() method, 121–122
startService() method, 70–71, 104, 147
@Startup annotation, 14, 40
@Stateful annotation, 31
stateful EJB, façade pattern, 31
stateless constraint (REST), 168
stateless EJB, façade pattern, 29–31
stereotypes, 60
structural design patterns, 6
subjects, observer pattern, 140
Sugarsync, 174
sync() method, 120

Table class, 89–90
@Target annotation, 59, 74
Technology Compatibility Kit (TCK), 18
testLoggers() method, 120
TimedObject interface, 128
@Timeout annotation, 128, 132, 137
@Transient annotation, 158
timeout method, 132
timer service, 127–138
automatic timers, 130–131
calendar-based attributes, 134–136
development cycle, 128–129
implementation, 130
overview, 127–129
programmatic timers, 130, 131–134
transactions, 136–137
TimerService object, 128, 129, 131–134
timezone attribute, timer service, 130–131
TMAX JEUS 8, 18
topics noun, 173–174
topics, message-oriented middleware, 198
TraceObserver class, 145
Transaction Event Observer, 148–149
transactions
timer service, 136–137
transaction-sensitive observers, 151
type-safe DAO implementation, 162

uber classes, 204
uniform interface constraint (REST), 167
User class, 51
UserRepository interface, 51–52, 54–55, 59
users noun, 172–173
UserService class, 50–52, 53–55, 187
UserServiceFactory class, 51
util package, 206
utility classes, 206
web container, 15, 16, 17
Web layer (Middle tier), 14, 15, 16
web profile, 18–19
WebBeans, 55
@WebServiceRef annotation, 53
WebSockets, 195–197
Wildfly 8.0.0, 18

x-axis scaling, 201–202

y-axis scaling, 201–202
year timer attribute, 134