index

Symbols

- method 25
:paste command 20
!! method 215, 217
!? method 217
?: syntax 22
@BeanProperty annotation 253
@Column annotation 252
@cpsParam annotation 252
@ManyToOne annotation 252
@reflect.BeanInfo annotation 253
@reflect.BeanProperty 254
@serializable annotation 252
@specialized annotation 238
@switch annotation 63, 252
@tailrec annotation 63–64, 66, 252
* method 116–117
## method 28, 39
#:: method 195, 197
+ method 116–118
+: method 180, 189
++ method 141–142, 144, 180, 208
<:< 156
== method 28, 39, 131, 190

A

abstract class 243
abstract interfaces 82–86
  between software modules 84–85
  reasons for 85–86
AbstractAddress class 243–244
AbstractMethodError 84
AccessControlContext 103
accessor 152, 163
act method 214–215, 219–220, 223
Actor class 217, 219, 230, 232
Actor.actorOf method 231
ActorDispatcher class 87
ActorRef type 230–231
actors
  dynamic topology of 228–233
  failure zones for 221–225
  references for 216–221
  scheduling zones for 225–228
  supervisor 221
  when to use 212
AdaptiveSearchNode 231–232
add method 237
addHandle method 148
Address class 243–244
ALIVE class 249–250
annotation.target.getter 254
AnnotationHelpers 255
annotations 252–256
  and static fields 255–256
  for variance 141–144
  targets 254–255
anonfun class 50, 52
anonymous classes, serialization of 250–252
anonymous functions 7, 10
AnyRef 5, 28, 135, 138, 143, 245
AnyVal types 256
App trait 69–70
Application class 269
applicative functors 266
applicative style 268, 271
Applicative.build method 271
ApplicativeBuilder class 270–271
apply method 6, 187, 190, 215, 260, 263, 271
areEqual method 245–246
ArrayBuffer 186, 198–200, 208
ArrayList 199, 245–246
ArraySortTrait 211
avg method 51

B

B[A] parameter 151–152
Bar class 92, 100, 102
bar method 102
bar.Foo.type 124
Bar.type 100
bar(x) method 102
BeanInfo class 253
benefits, of type classes 166–167
binary compatibility 83
BinaryFormat type 99
BinaryTree 188–189
bindings 91, 122
BitSet 190
blocks, of code 45–47
boilerplate 26
BoxesRunTime class 238
boxing 235–236
Branch class 188
BufferedReader 274

BufferedWriter 275
build method 270
bundleResult method 219–220
by-name parameters 260
Byte type 11

C

C parameter 157
C.super 123
C++ variables 8
Callback type 136–137
CanBuildFrom class 207–208
canEqual method 41
case statements 23, 62, 65
Cat type 137, 139
Category Theory
  and functional
    programming 258–261
  functors 259, 262
  monads 259, 264–272
  morphism 258
cc target 21
chaining implicits 246, 248
changePassword method
    56–57
Child class 55
children method 159–160, 163
class arguments 26
ClassCastExceptions 239
ClassManifest 154–155, 211
closeResource method 126
code blocks 45, 47
coding conventions
  and other languages 44–47
  code blocks 45–47
Coll type 207–208
CollectionConverter type 247
collections 180–211
  and methods 205–211
  ArrayBuffer 198
  CanBuildFrom 207
  hierarchy of 180–181
  immutable collections
    192–198
    list 194–195
    stream 195–198
    vector 192–194
  IndexedSeq 189–190
  Iterable 185–186
  iterator 185
  LinearSeq 187–189
  List 194
  Map 191–192
  mutable collections 198–200
    ArrayBuffer 198–199
    observable 199–200
    synchronization of 200
  Observable collections 199
  parallel collections 203, 205
  Seq 187
  Set 190–191
  Splitable iterator 203
  Stream 195
  Traversable 182–185
  TraversableOnce 180
  vector 192
  views 201–203
colRank 107
combineResults 219
companion class 19
companion object 19
complexmath 117, 119
ComplexNumber class 115–
    119
composition
  inheritance 76–82
  member-composition 78–80
  using constructor
    arguments 80–82
CompressedIntSet 159
computeValue 108
concurrency, and
    immutability 31–34
conditional execution, using
    type system 167–178
  heterogeneous typed
    list 169–171
  IndexedView type 172–178
Config class 259, 261–263,
    269–271, 276
config file 202–203
ConnectionDataStore 270
constraints
  for type parameters 134–135
  for types 131–134
constructor arguments, compo-
    sition using 80–82
content method 160
context bounds 151, 153
convertToScala 240
count method 133
covariance 137
CreateArray 245
createConnection method 38
createDispatcher method 87
createErrorMessage
    method 23
createHandle 129
createPreparedStatement
    method 5
createSearchTree 223
CREATOR class 243–244
curCount parameter 219
currentIndex method 32
curried method 266
currying 266, 271

D

dangling operators 48–49
Data class 254–255
DataAccess class 77–80
DatabaseConnection 261
DatabaseResultSetWalker 207
dataId 254
DataStore class 269
DEAD class 249–250
deadActor 224
default concepts 31
default parameters 106
DefaultHandles 129–130
defaults
  implicit parameters
    with 106–112
  returning 35–36
delayed construction, of
    objects 69–70
delayedInit method 69–70
DelayedInit trait 69–70
Dependencies object 148
describeContents method 243
Dog class 58–59
domain-specific languages.
    See DSLs
doPrivileged method 103
Double type 115–118
doubleToReal 118–119
doWork 252
DriverManager.getConnection
    method 258, 261, 268, 270
DSLs (domain-specific
    languages) 18
dynamic deoptimization 15
dynamic topology, of
    actors 228–233

E

eager parsing, in REPL 19–20
early member definitions 71
EJB (Enterprise JavaBeans) 4
else statements 47

empty implementations 72, 76
EmptyList class 142
endofunctor 264
Enterprise JavaBeans. *See* EJB
Entity beans 4
environment function 260, 270
equals method 28, 42
escape analysis 15
Event object 40
executeLocalQuery function 230
ExecutorScheduler 226
existential types 144–149
Expand type 175–176
experiment-driven development, and REPL 18–19
Explicit object 95–96
explicit return statement 26
explicit return types 86, 88
expression-oriented programming 21–26
  mutability of 24–26
  no return statements 22–24
expression-oriented syntax 24
expressions, with parentheses 48–49
externalbindings.scala 94

F

F[_] type 262, 264–265, 267–268, 270
fa parameter 267
factory method 86, 190
failure zones, for actors 221–225
File parameters 272, 275
FileInputStream 272
FileLike, as type classes 163–165
FileLike.children method 162
FileLineTraversable class 183–185, 202
FileObject 160
FileWrapper class 104
filter method 6, 36, 187, 191, 198, 205
find method 6
findAnInt method 90, 97
first-class functions 7
flatMap method 258–259, 261, 264–266
flatten method 259, 264–265, 273
foldLeft method 51, 204–205, 215
Foo class 91, 93, 101
foo method 48, 55, 101, 136–137, 151–152, 156
Foo object 19, 127, 141, 250, 255
Foo type 97
Foo.baz 128
foo.Foo object 101
Foo.type 129
foo.type#Bar 124
Foo#Bar 124
FooHolder 46
fooToBar 103
for expression 35
force method 201, 203, 215
foreach method 36, 182–185, 197, 258, 274
ForkJoinPool 203–205
ForkJoinScheduler 226–228
forSome method 146–147
forward method 216–217
FrontEnd 222
Function interface 6
Function object 139–141, 156
function traits 13
Function1 class 7
functional programming
  and category theory 258–261
  applicative style 268–271
  concepts of
    in existing frameworks 4–6
    in Google Collections 6–8
  currying 266–271
  functors 262–266
  monads 264–266, 272–276
  vs. object-oriented programming 2–8
functions, in Java 13–14
functionToPrivilegedAction 103
Functor interface 263–268, 273
Functor.apply method 273
Functor.map method 274
functorOps 263
functors 262, 266
Future object 215, 217

G

GathererNode 219–220, 231
Generic types 238
GenericSortTrait 209–210
GenIterator 181
GenSeq 181
GenTraversableOnce 181
get method 144–145, 172, 253–254, 259–260
getConnection function 261
getFoo 253
getLines method 274–275
getNextChild method 231
getOrElse method 34
getstatic operation 66
getTemporaryDirectory method 36
getValue 253
Google Collections, concepts of functional programming in 6–8

H

Handle type 129, 131, 147–148
handleMessage method 72–73
hashCode method 28, 30
HashMaps 191, 214
HashSet 190–191
HasLogger 79
hasNext method 185–186
HCons class 169–172, 175
head method 187, 189, 195
HeadNode 215, 217, 220–222, 224, 228
heterogeneous typed list 169, 171
hierarchy, of collections 180–181
higher-kinded types 135–136
HList class 170, 174–178
HListViewN class 173–174, 176
HNil class 169–172, 175
holder object 20
HotSpot runtime optimizer 14–15
HttpSession 36

I

i object 115–117, 119
identifiers 91–92
if block 22
if clause 22
if statements 25, 47, 61
If type 168
if_icmpne 63

imaginary method 115, 118
immutability 23–34
  and concurrency 31–34
  and object equality 27–31
immutable collections 192–198
  list 194–195
  stream 195–198
  vector 192–194
immutable references 26
ImmutableHashMap 32
ImmutableService 33
Imperative coding 24
implicit constraints 151–153, 207
implicit conversions 244–248
  and object identity 245–246
  chaining implicits 246–248
implicit method 90, 96–97, 101
implicit parameters, with defaults 106–112
implicit views 101, 106
implicitly function 99–100
implicits
  capturing types with 153–159
    Manifests 153–155
    specialized methods 158–159
    type constraints 156–158
  context bounds 151
  conversions 10
  implicit scope 98–102
  scope of 112–119
    and bindings 92–96
    creating implicits for import 113–115
    via nesting 99–101
    via type parameters 98–99
    without requiring import 115–119
  view bounds 151
import statement 92, 113
index service 31
IndexedSeq class 187, 189–190
IndexedView type 172–178
indexN function 177
inexpressible language features, in REPL 20–21
inheritance, composition including 76, 82
initCoreSize 228
inlining 15
inner type 124
InputChannel 216
InputStream 160, 274–275
InputStreamReader 274
insert method 31
InstantaneousTime class 38
int type 237–238
interfaces, abstract 82–86
  between software modules 84–85
  reasons for 85–86
IntHolder class 130
IntStore class 130–131
intToString 101
InvalidClassException 249
IScheduler interface 225
isDirectory 159–160
isEmpty method 187
isLoggedIn method 56
ItemType 141–142
Iterable interface 181–182, 185–186, 206, 210
IterableLike 210
Iterables object 7
Iterator method 181, 185

J

Java and Scala 12–15
  annotations 252–256
    and static fields 255–256
    annotation targets 254–255
  benefits of JVM 14–15
  implicit conversions 244–248
    and object identity 245–246
    chaining implicits 246–248
  language differences 235–244
    in primitive boxing 236–240
    in visibility 240–241
    unique features 241–244
  Scala objects in 13
  serialization 248–252
Java class 12
Java interfaces 12
java.awt.Component 74
java.io.File 104–105, 163–164
java.lang.Class 154
java.lang.IndexOutOfBoundsException 186
java.lang.Integer type 239–240, 246–247
java.lang.Object 135, 144–145, 235
java.lang.String type 113
java.net.URL 163
java.security class 103
java.sql.DriverManager.getConnection(...) method 266
java.util.ArrayList 199, 239
java.util.Collections class 152
java.util.concurrent.Executor 226
java.util.concurrent.Executors 110
java.util.Date class 38
java.util.Executor 226
java.util.List 92, 145
JavaClass 245
JavaConversions 244–245
javap 82, 87
javax.swing.JComponent 74
jdbc_password 261
JdbcTemplate class 4
JdbcTemplate method 5
jdbc_url 261
jdbc_user 261
JRebel 21
JVM bytecode 14
JVM, benefits of 14–15

K

KittyDoggy class 58

L

lambdas 10
language differences, Java and Scala 235–244
  in primitive boxing 236–240
  in visibility 240–241
  unique features 241–244
LazyTraversable type 274–275
Leaf type 188–189
LeafNode 229, 232
limiting scope, of implicits 112–119
  creating implicits for import 113–115
  without requiring import 115–119
LinearSeq 187–189, 194–195, 204
lineLengthCount method 275
link method 221, 223
List class 134–135, 145–146, 195–196, 237–238

list collections 194–195
List(start) method 64
ListView 201
loan method 273–276
loaner pattern 5
LoggedDataAccess class 78
Logger class 77–79
login method 56
logout method 56
longWrapper 114
lookUp method 31
loop method 65

M

makeLineTraversable method 274
makeList method 145
ManagedResource 273–276
ManagedResource.writeFile method 275
Manifest class 154
MatchError 63
MatrixService.multiply 111–112
maxDocs 219
maxResponses 219
maxResults 214
maxSize 228
member-composition 78, 80
menu button click 25
MessageDispatcher 86–88
method inlining 15
method parameters 26
methods
  and collections 205–211
    overridden, marking 55–60
MixableParent 75
mkdirs method 160
MODULE$ 13, 255
Monad type 264–266, 273
monadOps 265
Monads 264–266, 272–276
move method 27
multiple inheritance, of objects 70–72
mutability, of expression-oriented programming 24–26
mutable collections 198–200
  ArrayBuffer 198–199
  observable 199–200
  synchronization of 200
Mutable objects 25
MutableService 32
mutation statements 26

N

NaiveQuickSort object 206
naiveWrap method 247
named and default parameters 49
named parameters 53, 55
naming, variables 49–55
Nat type 175–177
nesting, scope of implicits via 99–101
NetworkEntity 73, 75
newMethod method 83
next method 185–186, 189, 197, 238
NextIdxView 174
NilTree 188–189
Node class 64–66
NoManifest class 154
None 34–38
  creating new object or returning default 35–36
  executing block of code if variable is initialized 36–37
  using potential variables to construct another 37–38
NonZero type 176
null object 160
NullDispatcher 87
Numeric type 158–159
NumericRange 114

O

object equality, and immutability 27–31
object identity, and implicit conversions 245–246
ObjectInputStream 249
object-oriented programming
  composition 76
  traits 69
  vs. functional programming 2–8
ObjectOutputStream 249
objects 69–72
  delayed construction 69–70
  in Java 13
  multiple inheritance 70–72
observable collections 199–200
ObservableBuffer 199–200
ObservableMap 199
ObservableSet 199
observe method 129
On Stack Replacement 15
operator notation 10
operator overloading 25
operators, dangling 48–49
optimization
  tableswitch optimization 61–64
  tail recursion optimization 64–66
Option class 34, 258–261, 263, 266, 271
OptManifest 154
or method 7
Ordering type 190, 206, 211
OriginalType 101
OtherItemType 142
Outer class 124
OutputChannel 216–217, 219–220
OutputStream 275
overridden methods 55
override keyword 55–57, 59–60

P

package.scala 100
par method 201
parallel collections 203, 205
parameters 53, 55
parameterspaths, and types 122
ParArray 204
Parent class 55
parentheses, expressions with 48–49
ParentNode 231
parsedConfigFile method 202
parsing, in REPL 19–20
ParVector 204
paths, and types 122, 124
pattern matching 23, 25, 61
Player class 249
PlayerStatus 249
plus method 159
Point2 class 28
Point2D class 27
polymorphic equality 38–42
  implementing 40–42
  timeline library example 38, 40

postfix operator 10
Predef class 113–114
Predicate interface 6–7
Predicates class 6
PreparedStatementCreator interface 5
Prev type 177
primitive boxing, in Java and Scala 236–240
primitive widenings 11
primitives 236
println method 131, 148, 186, 189
private keyword 20
private variables 19
PrivilegedAction 103
PrivilegedExceptionAction 103
Property trait 71
PureAbstract 85

Q

qsort 10
QueryResponse 218
QuickSort 209, 211
Quicksort method 9
QuickSortBetterTypes object 206

R

randomElement method 134
Range object 114
react method 215, 224
Read Eval Print Loop. *See* REPL
readFile method 272–275
readLine method 274
real method 115, 118
realToComplex 117–118
receive method 215, 232
receiver method 217
Receiver type 153
receiveWithin method 220
Ref type 147–148
references, for actors 216–221
regular object 19
reification 156
remove method 148
removeDependencies method 148
REPL (Read Eval Print Loop) 16–21
  and experiment-driven development 18–19
  eager parsing in 19–20
  inexpressible language features 20–21
reply method 215
replyTo 217
repr member 40
ResizableThreadPoolScheduler 226
Resource type 126
result variable 23
ResultSet 5
return method 65
return statements, lack of 22–24
return types, explicit 86–88
Router class 75
RowMapper interface 5
rowRank 107
run method 224
runtime type 152

S

SameThreadStrategy 109, 111
Scala file 21
Scala functions 13
Scala objects 12–13
Scala type 8
Scala variables 8
Scala, in Java 13–14
scala.actors.TIMEOUT 220
scala.collection.immutable.List class 180
scala.collection.immutable.List type 238
scala.collection.immutable .Vector 180
scala.collection.Java-Conversions 103
scala.collection.JavaConverters 244
scala.collection.parallel 205
scala.collection.script 200
scala.collections.mutable 198
scala.immutable.List 169
scala.Int 236–237, 239, 246–247
scala.Iterable 245, 247
scala.List 92, 154, 183
scala.Option 34, 122
scala.Predef 10, 112–113, 157, 191, 246
scala.Predef.longWrapper 114
scala.runtime.BoxesRunTime class 238
scala.String 124
scala.type#String 124
scala.util.control.Control-Throwable 184
scala> prompt 17
ScalaClass 245
ScalaMain 84
ScalaObject 84
ScalaSecurityImplicits 103
scatter-gather example 217, 221, 225
schedule method 251
scheduling zones 225–228
SchedulingService 251–252
scope, of implicits
  and bindings 92–96
  limiting 112–119
  via nesting 99–101
  via type parameters 98–99
sealed trait 216
search method 64
search, using actors to 213–216
SearchableDocument 229, 231
SearchNode 214–215, 218, 221–223, 226, 228
SearchNodeMessage type 217–218
SearchNodes 215, 217–221
SearchNodeSupervisor 223–224
SearchQuery class 214–215, 217–218, 220, 229, 231
self-type 73
send method 153
sendMsgToEach 153
seq method 201
SeqLike class 207
Serializable class 153, 158, 165–166
serialization 248–252
service classes 2
Session beans 4
Set class 132, 159, 178
set method 253–254
setFoo 253
setValue 253
Simple Build Tool 21
SimulationEntity 72, 74
sizeHint 208
SLEEPING class 249–250
sort method 206, 208–209, 211
Sortable type 209–211
sortBy method 215
Sorter class 167–168, 209
Sorter.sort method 210

specialization 238
specialized methods 158–159
split method 230
Splitable 203
Static class 12
static fields, and annotations 255–256
static methods 13
static typing 8–12
  dropping verbose syntax 9–10
  implicits 10–12
  type annotations for variables 8–9
  type inference 9
statics 241
Stream class 192, 195–196, 198
stream collections 195, 198
Stream.empty 195
strictEquals method 29
String class 121, 124, 135, 138, 140, 143, 146
String object 101–102
structural types 125, 131
style guide 47
Succ trait 175–176
sum method 159, 187
super.handleMessage 73
synchronization, of mutable collections 200
synchronize function 161, 163, 178
synchronized block 32
SynchronizedBuffer 200
synchronizedCollection (Collection) 152
synchronizeDirectory 161
SynchronizedMap 200
SynchronizedPriorityQueue 200
SynchronizedSet 200
SynchronizedStack 200
syncronizedList(List) 152

T

T#X type 127
T#Y type 127
tableswitch bytecode 61
tableswitch optimization 61, 64
tail method 187–189
tail recursion optimization 64, 66
take method 184–185, 187, 215
TBool 168
TDD (test-driven development) 18
Test object 69, 74, 94, 102, 114, 247
test.Foo class 91–92
test.txt file 183, 272
test-driven development. *See* TDD
testExplicitImport method 95
testInlineDefinition method 96
test_prop property 260
testSamePackage 94
testWildcardImport method 95
TFalse type 168
ThreadPoolStrategy 110–112
ThreadStrategy 109–112
Time object 113–114
timeline library example, polymorphic equality 38, 40
TimeRange class 113–114
tmp method 93
toList method 202
toSet method 191
toString method 17, 71, 94, 101, 107, 130, 183
trait linearization 71
Traversable class 180–185, 197, 202
TraversableOnce 180
TraversableOnce.scala 158
TraversableView 197–198, 202–203, 275
traverse method 188
traverseHelper method 188–189
TreeMaps 191
TreeSet 190–191
TTrue type 168–169
TupleN 169
two-dimensional geometric point class 27
two-dimensional plane 27
type annotations, for variables 8–9
type classes 159–167
  benefits of 166–167
  FileLike as 163–165
type constraints 131, 156, 158
type erasure 167, 235
type inference 9
type keyword 124–125
type lambda 137
type parameters
  constraints for 134–135
  scope of implicits via 98–99
type system, conditional execution using 167–178
  heterogeneous typed list 169–171
  IndexedView type 172–178
type traits 99
types
  abstract types 124
  and paths 122–124
  capturing with implicits 153–159
    Manifests 153–155
    specialized methods 158–159
    type constraints 156–158
  concrete types 124
  constraints 131–134
  existential types 136, 144–149
  higher-kinded types 135–136
  path-dependent type 123
  structural types 125–131
  type keyword 124–125
  type parameters, constraints 134–135
  type projection 123
  volatile type 122

U

useFile method 105
UserService class 57
UserServiceImpl class 57
UserSession object 56
using statement 92

V

value method 253
value_$eq method 253
var syntax 26
variables
  executing block of code if initialized 36–37
  naming of 49–55
  type annotations for 8–9
VariableStore 148–149
variance 137–144
  annotations for 141–144
  contravariance 139
  covariance 137
  invariance 137
vector collections 192, 194
view bounds 151, 153
view method 197, 201–202, 215

ViewAt method 175–177
views 201, 203
ViewType 101
visibility, in Java and Scala 240–241

W

Wildcard object 95–96
withFilter 258
workerPool method 269
wrap method 105, 247
WrappedArray 152
writeClient method 165
writeContent method 160
writeToParcel method 243

X

x method 91, 114, 127–128, 241
x object 94
x parameter 90
X type 127–128, 130, 168–169
x-axis 32

Y

y method 91, 111, 124, 128, 131, 146
y-axis 32

Z

zip method 186, 243