

Index

A

- Abs class, 304
- Abstract classes, 347, 349
 - defined, 431
 - and interfaces, 9–10
- ABSTRACT FACTORY pattern, 160, 179–190, 191
 - abstract factories and factory method, 185–189
 - defined, 179, 190
 - GUI kits, 179–184
 - namespaces and abstract factories, 189
 - abstract keyword, and constructor declarations, 381
- Abstract methods, 69–71, 223, 426
 - defined, 431
- Abstract syntax tree, 341
 - defined, 431
- Abstraction, 17, 65–72, 214, 359
 - defined, 71, 431
- Accept () method, 342, 414–415
 - MachineComponent class, 330, 333
 - ProcessComponent hierarchy, 340
- Access modifiers, 151, 153
 - in C#, 78
 - function of, 80
- Active server pages (ASP), 132, 418
 - defined, 431
- ADAPTER pattern, 5, 17, 19, 19–33, 70, 226, 351, 410
 - usefulness of, 33
- Add or Remove Programs, Control Panel, 133
- AddNode () method, 332
- Add/Remove Windows Components, 133
- Administrative Tools, Control Panel, 133
- ADO.NET, 35–36
- Advisor interface, 250
- Aerial shells, 38, 58, 61, 176, 324, 358
 - defined, 431
 - process flow for making, 59
- Alexander, Christopher, 1
- Algorithms, 218–219, 220–221, 223–224
 - defined, 218, 431
- Anchoring a chain, 144–146
- Apogee, 130, 132, 225–226, 374
 - defined, 431
 - app subdirectory, oozinoz directory, 420
- Architectural patterns, intent of, 1–2
- Argument-free delegate, 102
- Arithmetic class, code for, 304–5
- Array class, 224
- ArrayList class, 224
- ArrayList collection, 14
- ASP.NET, 132, 137, 344
 - central benefit of, 137
 - defined, 431
- Assay class, 53
- Assay, defined, 53, 431
- AssignButton () method, 111
- AssignTab () method, 89
- Aster star press, 227–229

AsterStarPress class, 227–228, 261–262

B

:base(), 155–156, 381
 Batch class, 53
 Beck, Kent, 287, 289
 Bin class, 80, 363
 bin subdirectory, oozinoz directory, 420
 BorrowReader delegate type, 10–13, 260
 BorrowReader() method, 347
 BRIDGE pattern, 17, 65–73, 372, 410
 abstraction, 65–68
 defined, 65
 intent of, 65
 most common example of, 71
 Bridges:
 creating, 66
 drivers as, 70–71
 Build() method, Unforgiving-Builder class, 384–385
 Builder behavior, differences in, 167
 BUILDER pattern, 160, 161, 163–169, 191, 372
 building under constraints, 166–168
 code that uses a builder, 167–168
 defined, 163, 169
 forgiving builder, 168–169
 ordinary builder, 163–165
 BuilderException, 167
 buildInner step, 60
 BurnRate class, 96–97
 F() method, 96
 and Tpeak object, 103
 Business object, 28, 103–104
 defined, 431
 BusinessCore namespace, 312
 Button object, 192

C

C#, 6, 21–22

 delegate type in, 257
 delegates in, 264
 for and foreach statements in, 271
 C# and the .NET Platform (Troelsen), 132, 157, 345
 C# interfaces, 9
 C# Language Specification, The (Wiltamuth), 13, 16, 78–79, 214, 349, 362, 398
 C# method header, 214
 C++, 3, 9
 interface construct, 9
 Callback scheme, 258
 Carousel, 232, 236–238, 242, 255
 defined, 431
 CarryCommand class, 270–271
 Cartesian product, 115
 Centralization of responsibility, 109
 CHAIN OF RESPONSIBILITY pattern, 81, 139–146, 390
 anchoring a chain, 144–146
 intent of, 139, 143
 occurrences of, 146
 ordinary chain of responsibility, 139–141
 refactoring, 141–144
 Responsible property, 140
 without composite, 146
 Challenges, 1, 3, 4–6, 10, 13, 14, 15, 16, 23, 27, 32, 37, 44, 47, 48, 50, 51, 53, 55, 56, 58, 60, 66, 67, 70, 71, 76, 77, 78, 79, 80, 84, 86, 87, 93, 991, 103, 106, 112, 113, 117, 120, 121, 126, 127, 131, 136, 141, 143, 144, 145, 146, 148, 149, 151, 153, 157, 158, 159, 165, 168, 169, 173, 175, 177, 183, 184, 187, 189, 193, 194, 196, 203, 206, 208, 209, 214, 215, 217, 218, 219, 226, 229, 230, 232, 236, 239, 244, 245, 250, 252, 255, 256, 258, 261, 263, 264, 274, 278, 279, 285, 286, 287, 288, 298, 300, 306, 308, 314, 316, 323, 327, 331, 336, 341, 342
 Change delegate, 92–93
 as an event, 94

- Change events, 366–367
- Change notification, listeners registering for, 105–106
- ChangeHandler delegate type, 90–92, 367
- CheapRockets class, 361
- Check() method, 384–385
- CheckFactoryUS class, 189
- Chemical class, 149, 151, 379–380
- Chemical object, 53–54
- ChemicalFactory class, 150–151
- ChemicalFactory2 class, 152, 380
- ChemicalImpl class, 152–153, 381
- _childEnumerator variable, 319–320
- Circle, 409
- Class adapters, 24, 33
- Class diagrams, UML, 422–423
- Class relationships, UML, 424–425
- Class types, 361–362
- Classes, 348
 - abstract, 347
 - attributes, 214, 362
 - defined, 361
 - implementation, 9
 - interface, 9
 - name of, 362
- UML, 422–423
- _classLock object, 85
- Click delegate, 258
- Click instance variable, 399
- Client code, 19
- Client, defined, 19, 431
- Clone() method, ArrayList class, 315–316
- Clone() signature, 398
- Clones, prototyping with, 193–196
- Cloning a collection, 315
- Code, refactoring, 37
- Code smells, removing, 286–287
- CommaListFilter class, 297–298
- Command hierarchy, 273, 279
- COMMAND pattern, 213, 257–265, 410
 - hooks, 261–263
 - menu commands, 257–259
 - in relation to other patterns, 263–264
 - using to supply a service, 259–261
- CommandSequence object, 269
- Common Object Request Broker Architecture (CORBA), 132
 - defined, 432
- CompareTo() methods, 225
- Complete() method, 243, 245
- ComponentEnumerator class, 318–319
 - code for, 319
 - MoveNext() method, 318–320
- Composable output streams, creating a toolkit of, 294–295
- Composite class, 49–50, 356
- COMPOSITE pattern, 17, 49–63, 214
 - DECORATOR pattern compared to, 309
 - ordinary composite, 49–50
- Composites:
 - and cycles, 53–58
 - with cycles, 58–62
 - defined, 49, 431
 - modeling, 63
 - recursive behavior in, 50–53
 - and trees, 53–58
- Concrete class, 70, 187, 391–392
 - defined, 432
- Concurrent Programming in Java (Lea), 85, 346, 414
- config subdirectory, oozinoz directory, 420
- ConsoleWrite class, 411
- ConsoleWriter.cs, 300
- Consolidation languages, 3
 - defined, 21, 432
- Construction patterns, 155–209
 - ABSTRACT FACTORY pattern, 160, 179–190
 - BUILDER pattern, 160, 163–169
 - construction challenges, 155–159
 - FACTORY METHOD pattern, 160, 171–178
 - MEMENTO pattern, 160, 197–209
 - PROTOTYPE pattern, 160, 191–196
- Constructors, 40, 157–158, 160–161, 163, 191, 193, 253

- defined, 432
- return type, 381
- special rules regarding, 381
- Context-free language, 434
 - defined, 432
- Control, 40, 46, 179–180, 184, 190, 192, 205, 353, 394
 - defined, 432
- Control class, 372–373
- ConveyIn() method, 66
- ConveyOut() method, 66
- Copy() method, adding to a non-sealed class, 194
- Cos class constructor, 304
- Create-() methods, 192
- CreateAdapter() method, 28
- CreateConnection() method, 11
- CreateCreditCheck() method, 175, 185, 387–388
- CreateGrid() method, 46
- CreateGroupBox() method, 46
- CreateHandheldUI() method, 192–193
- createMemento() method, 396–397
- CreatePlanner() method, 390, 401
- CreatePlotPanel() methods, 43, 46
- CreateTable() method, 260
- Credit namespace, 186–187
- Credit package, 185–186
- Credit.Canada package, 391–392
- CreditCheckFactory class, 175, 185, 189, 388, 393
 - IsAgencyUp() method, 188
- CreditCheckOffline class, 174
- CreditCheckOffline object, 186–188, 393
- CreditCheckOnline class, 174
- CreditLimit() method, 174
- Curious class, 90–91
 - React() method, 91, 93, 365
- Customer class, 251, 254
- Cycles, 53–58, 337–338
 - composites with, 58–62
 - consequences of, 62
 - defined, 54, 432

VISITOR pattern, 337–341

D

- Data access, 35
- Data members, 361
- Data proxy, 127–132
- Data reader proxy, uses for, 373
- Database drivers, 71
- DataGrid class, 29, 31–32, 33, 353
- DataLayer class library, Oozinoz, 127
- DataLayer namespace, 11, 259, 347–348
- DataLayer.dll, 134
- DataReaderProxy class, 129–130
- DataServices class, 35, 259–260
 - CreateTable() method, 36
- DataSet class, 29
- DataTable instance, 31
- DataGridView instance, 31
- DataGridViewManager instance, 31
- DataWebService proxy class, 135–137
- DataWebService.cs file, 135
- DateTime struct, 156–157
- db subdirectory, oozinoz directory, 420
- DECORATOR pattern, 289, 291–309
 - function wrappers, 300–308
 - in GUIs, 308–309
 - mechanics of, 309
 - in relation to other patterns, 309
 - streams, 291–300
- Deep copy, defined, 395, 432
- Default constructors/default constructor behavior, 155
- Delegate class, 92
- Delegate declaration, 92, 399
 - defined, 216
- Delegate instance, 399
 - delegate keyword, 10, 348
- Delegate type, 10–14, 90–92, 95, 201, 257–258, 427
 - defined, 432
- Delegates, 10, 16, 19, 89–90, 94, 101, 139, 213, 215, 216–217, 348, 427–428

- in C#, understanding, 14
 - defined, 432
 - and interfaces, 10–14
 - strength of, 13
 - syntax for using, 216
 - Demos, 47–48
 - facades compared to, 355
 - Depth() abstract, CompositeEnumera-
tor class, 325–326
 - Design Patterns*, 2–6, 9, 83, 209, 404
 - Design patterns:
 - building fluency with, 342
 - built-in C# features, 9
 - classics, understanding, 343
 - defined, 2, 432
 - extending behavior with, 410
 - intent, categories of, 5–6
 - learning, primary purpose of, 344–345
 - purpose of, 2
 - Directed graphs, 54
 - defined, 432
 - Directions, 343–346
 - disassemble step, 60
 - Discharge() method, 66
 - DischargePaste() method, 230
 - DisplayUpMachines() method, 313,
315, 413
 - Dispose() method, 30
 - Domain-specific composites, 327–328
 - Dominant state variable, 246
 - Door class:
 - ChangeHandler variable, 237
 - SetState() method, 238–239
 - Touch() method, 238–239, 241, 244
 - DoorClosed class, Touch() method, 241
 - DoorClosed.cs, 240
 - DoorOpen class, Touch() method, 244
 - DoorState class, 242, 244–245
 - Double dispatch, 334–335, 342
 - defined, 334, 432
 - Driver-based design, 70
 - Drivers, 71–72
 - as bridges, 70–71
 - database, 71
 - defined, 70, 432
 - Dublin() method, ExampleMachine
class, 331
 - Dud, 38–39, 46
 - defined, 38, 432
- ## E
- Ellipse, 409
 - Enabled property, 370
 - Encapsulation, 81, 209, 398
 - defined, 432
 - Engine class, Tpeak property, 103
 - Enumerator, instantiating, 328
 - Enumerators, 171–173
 - Equals class, 275
 - Eval() method, 275
 - event keyword, 348
 - EventArgs argument, 102
 - EventHandler delegate type, 258, 264,
366
 - Events, 13, 16, 89, 94, 112, 235, 348–349
 - defined, 433
 - ExampleMachine class, Dublin()
method, 331
 - Execute() method, 280
 - CommandSequence object, 269
 - ForCommand class, 408–409
 - ExistingClass class, 24
 - ExitApp() method, 405
 - Extensible Markup Language (XML), 135,
163, 384
 - defined, 433
 - Extension patterns, 281
 - code smells, removing, 286–287
 - DECORATOR pattern, 289, 291–309
 - ITERATOR pattern, 289, 311–328
 - Law of Demeter (LoD), 285–286, 289
 - Liskov Substitution Principle (LSP),
284–285, 289
 - object-oriented design, principles of, 283
 - VISITOR pattern, 289, 329–342

F

- F () method, 304, 308
 - Facade, defined, 48
 - FACADE pattern, 17, 35–37, 35–48, 372
 - and .NET FCL, 37
 - Facades:
 - advantages of, 353
 - demos compared to, 355
 - and utilities, 46–47
 - FACTORY METHOD pattern, 160, 171–178, 190, 191, 372
 - and client code, 178
 - defined, 170
 - enumerators, 171–173
 - GetEnumerator () methods as examples of, 171
 - and instantiation, 173
 - in parallel hierarchies, 176–177
 - recognizing, 173–174
 - taking control of which class to instantiate, 173–174
 - FileFinder class, 11, 419
 - FileStream object, 292
 - Fill () method, 28, 30
 - Filter classes, developing a collection of, 292
 - Firework class, 383
 - Firework object, 362
 - Fireworks, 6, 19, 50, 58, 60, 65
 - defined, 433
 - Fireworks library, 351
 - FireWorks.dll, 134
 - FLYWEIGHT pattern, 81, 147–153, 372, 404
 - defined, 147, 153
 - extracting the immutable part of a flyweight, 148–150
 - immutability, 147–148
 - sharing flyweights, 150–153
 - ForCommand class, 271, 274–275
 - Execute () method, 274
 - foreach statement, 311
 - Forgiving builder, 168–169
 - ForgivingBuilder class, 169
 - Fowler, Martin, 287, 289
 - Framework Class Libraries (FCL), 6, 28, 94, 193–194, 257, 309, 327, 344, 355
 - defined, 433
 - and mediators, 112
 - Frapper class, 301–305
 - Function delegate, 43–45
 - Function members, 361
 - Fuse () constructor, 382
 - Fuse superclass, 156
 - FuserController class, 66, 67
 - FuserDriver class, 70
 - FuserManager class, SwitchSpool ()
 - method, 67
 - Futurize () method, 165
- G**
- “Gang of Four” (GoF) book, 3
 - GetChemical () method, 381
 - GetEnumerator () method, 311, 313, 321
 - FACTORY METHOD pattern, 178
 - GetEnumerator (:Set) method, 320–321, 323
 - GetFactory () method, 84
 - code, 85
 - GetFileName () method, 418–419
 - GetImage () method, UI class, 126
 - GetLocation () method, 360
 - GetMachineCount () method, 51, 55, 356–358
 - GetMass () method, 27
 - GetMaterial () method, 52–53, 357
 - GetNames () method, 10–11, 13
 - and LimitingReader proxy, 131
 - GetOwners () method, 52–53, 357
 - GetPlanner () method, 231, 401
 - GetPretty () method, 339
 - GetRecommended () method, 249–253, 362
 - getResponsible () method, 376
 - GetSqlMoney () method, 71
 - GetStepCount () method, 61

ProcessComponent class, 61
 GetThrust () method, 27
 GetTubs () method, 285
 Grammar, 280, 341, 432, 434
 defined, 433
 Graph, defined, 53
 Graph theory, 53, 54
 defined, 433
 Graphical user interface (GUI), 29, 432
 defined, 433
 Graphical user interface (GUI) objects, 29
 Graphs, 55, 62
 defined, 53, 433
 directed, 54, 432
 Grid object, 192
 GroupAdvisor class, 251–253, 404
 GroupBox class, 40
 GroupBox object, 192, 194–195
 GUI layer, 103
 GUI mediators, 109–114

H

HandleUndo () method, 397
 Handshaking, 68–69
 Hashtable class, 117
 HasItem () method, 253
 HasMaterial class, 276
 Holland, Ian, 285
 Hooks:
 COMMAND pattern, 261–263
 defined, 229, 433
 TEMPLATE METHOD pattern, 229–230
 Hopper, 227
 defined, 433
 Hopper class, 363
 Hsk subclasses, 68

I

IBorrower interface, 347
 IChemical interface, 152
 IComparable interface, 225

IComparer interface, 224–226
 ICompositeEnumerable interface, 321–322
 ICreditCheck interface, 175, 185, 387–388
 IDataReader interface, 12, 128–129
 IDataRecord, 129
 IEnumerator interface, 318
 ICommand class, 275, 277
 IFrapper interface, 301
 IList interface, 31
 IListSource interface, 31
Illness as Metaphor (Sontag), 217
 IMachineDriver interface, 70
 images subdirectory, oozinoz directory, 420
 Immutability, 153
 of strings, 378
 Immutable, defined, 147, 433
 Implementation, defined, 9, 433
 Index () method, 66
 Indexers, and interfaces, 14, 16
 inspect step, 60, 63
 Integrated Development Environment (IDE), 35
 advantages of, 353
 defined, 433
 Interesting class, 90–92
 Change delegate, 93–94
 Wiggle () method, 92, 93, 365
 Interface patterns, 17
 ADAPTER pattern, 5, 17, 19–33
 BRIDGE pattern, 17, 65–72
 COMPOSITE pattern, 17, 49–63
 FACADE pattern, 17, 35–48
 Interfaces, 9–17, 349
 and abstract classes, 9–10
 adapting to, 19–24
 defined, 9, 433
 and delegates, 10–14
 details, 15–16
 power of, 16
 and properties, 14–15
 UML, 424–425

internal access, 78
 Internet Information Server (IIS), 133–134
 Interpreter, 267–268, 275–276
 defined, 433
 INTERPRETER pattern, 213, 267–280, 410
 COMMAND pattern compared to, 263
 DECORATOR pattern compared to, 309
 example, 267–279
 interpreters, 279–280
 languages, 279–280
 parsers, 279–280
Introduction to Algorithms (Corman), 218
 “Intuitive meanings” of access modifiers, 78–79
 IRocketSim interface, 21–23, 350–351
 IsAgencyUp() method, 388–389
 IsCompletelyUp() method, 52–53, 357
 IsComposite() method, 416
 ISimpleWriter interface, 293
 IsLeaf() method, 416
 IsLiquid() method, 361
 IsTree() algorithm, 219–220
 IsTree() method, 357–358
 MachineComponent class, 56–58
 IsTree(h:Hashtable) method, 57
 ITable interface, 353
 ItemAdvisor class, 251–253, 404
 ITERATOR pattern, 171, 289, 311–328
 intent of, 327
 iterating over a composite, 316–327
 composite enumerator, adding depth to, 325–326
 enumerating leaves, 326–327
 ordinary iteration, 311
 thread-safe iteration, 311–316

J

Java, 3
 Java 2 Platform Enterprise Edition (J2EE)
 specification, 3
 JUnit, 419

K

Kit, defined, 434

L

Language, 3, 257, 267, 279–280, 341
 defined, 434
 Law of Demeter (LoD), 285–286, 289
 defined, 285, 434
 Layer, defined, 103, 434
 Layering, 103–107
 Lazy-initialize, 84–85, 87, 363
 defined, 434
 Leaf class, 49–50
 Leaves, 49
 defined, 434
 LendReader() method, 11–12, 259–260
 lib subdirectory, oozinoz directory, 420
 Lieberherr, Karl, 285
 LikeMyStuff engine, 248–249
 LimitingReader object, 131, 374
 LiquidRocket class, 360
 Liskov Substitution Principle (LSP), 284–
 285, 289
 defined, 434
 violating, 286
 ListBox object, SelectedItem property, 370
 ListTable class, 32, 353
 Load() method, 126
 LoadImage() method, 216, 399
 Lock, 314–315, 435
 defined, 85, 434
 Lookup() method, 423
 Loose coupling, 109, 114, 121
 defined, 434
 LowerCaseFilter class, 297

M

Machine class, 55, 57, 120
 CreatePlanner() method, 230–231

- Parent and Responsible properties, 140
 - UnloadBuffer subclass, 284
 - Machine object, 376
 - Machine parallel hierarchy, 176–177, 389
 - MachineComponent class, 330, 336, 400
 - IsTree() method, 56–58
 - methods, 52–53
 - MachineComponent object, 377
 - MachineComposite class, 54, 55, 57, 330, 356, 414–415
 - Parent and Responsible properties, 140
 - MachineComposite object, 376
 - MachineManager abstraction, 69
 - MachineManager class, 67, 69, 70
 - Shutdown() method, 68
 - MachineManager parallel hierarchy, 69–70
 - MachineManager2 class, 362
 - MachinePlanner parallel hierarchy, 176–177, 231, 389
 - MachineSimulator class, 395
 - Main() method, 313
 - and Limiting Reader proxy, 131
 - Make() method, 324
 - make process, 60
 - MarkMoldIncomplete() method, 228, 261–262, 400–401
 - Math class, 364
 - MEDIATOR pattern, 81, 109–122
 - combining COMMAND pattern with, 263
 - defined, 109, 121
 - GUI mediators, 109–114
 - and responsibility, 109
 - MemberwiseClone() method, 194–196, 394–395
 - MEMENTO pattern, 160, 197–209
 - defined, 197, 209
 - intent of, 407
 - memento:
 - defined, 206
 - durability, 206
 - using for undo, 197–206
 - and object state, 206
 - persisting mementos across sessions, 206–209
 - Mementos, storing as objects, 397
 - MenuItem constructors, 207
 - Message, 107, 113, 123, 127, 132, 168, 205, 242, 285
 - defined, 434
 - MessageBox class, 47–48, 355
 - Method headers, leaving exception declarations out of, 399
 - Method lookup, defined, 220, 434
 - Methods, 219, 220
 - attributes, 214
 - body, 214
 - defined, 213–214, 434
 - header, 214
 - invoking, 220
 - modifiers, 214
 - Minus sign (-), and instance variables/methods, 423
 - Mixture class, 149
 - Model/View/Controller (MVC), 108, 199, 203, 216, 263, 368, 372
 - defined, 101, 106–108, 434
 - Moles, defined, 149, 434
 - Mortar, 58
 - defined, 434
 - MoveATub class, 110–113, 116
 - AssignButton() method, 111
 - SelectChanged() method, 112
 - MoveNext() method, CompositeEnumerator class, 318–321, 327
 - MulticastDelegate class, 92
 - Multi-threaded environment, lazy-initializing a singleton in, 84–85
 - Mutex, defined, 313, 314, 435
- N**
- Namespace, defined, 435
 - NAnt tool, 418, 420

- defined, 435
- Nested classes, 151, 153
- Nested types, 361
- .NET, adapting data in, 28–33
- .NET FCL, *See* Framework Class Libraries (FCL):
- .NET Framework, 3, 6, 135, 257, 433
 - defined, 435
- .NET Framework Class Libraries (FCL), 6
- .NET Framework Class Libraries (FCL), 6, 28
- .NET menus, architecture of, 258
- .NET System.Collections library, 57
- .NET System.Data namespace, 11
- .NET System.Windows.Forms library, 94
- new keyword, and constructor declarations, 381
- NewClass class, 20, 24
- NewMachineComesUp() method, 315, 413
- N-tier, 10, 28, 107
 - defined, 435
 - design, 107
- NullCommand class, 277
- nunit subdirectory, oozinoz directory, 420
- NUnit tool, 419–420
 - test code, from TubTest.cs, 120–121
 - testing code with, 419
 - testing framework, 62

O

- Object adapters, 24–28, 33
- Object model relation, 115
- Object-oriented (OO) design, principles of, 283
- Object-oriented (OO) language, 2–3, 114
- Object-oriented (OO) programming, advantage of, 35
- Objects, UML, 428–429
- OBSERVER pattern, 81, 89–108, 410
 - C# support for, 89–90
 - delegate mechanics, 90–94
 - in GUIs, 94–101
 - intent of, 89
 - layering, 103–107
 - point of, 98
 - relational integrity mediators, 114–121
- OLE DB adapters/readers arguments in favor of using, 360
- OleDbDataAdapter class, 28–29
- OleDbDataReader class, 71, 127
- OnPaint() method, 126
- Oozinoz:
 - defined, 435
 - ShowProperties class library, 14
- Oozinoz DataLayer class library, 127
- Oozinoz DataServices class, 28, 157
- oozinoz environment variable, creating, 134
- Oozinoz Fireworks, 6
- Oozinoz Processes namespace, 62
- Oozinoz source, 417–420
 - acquiring/using, 417
 - finding files yourself, 419–420
 - helping the Oozinoz code find files, 418–419
 - Oozinoz code, building, 417–418
 - testing code with NUnit, 419
- Oozinoz Utilities class library, 418
- Oozinoz Utilities namespace, 11
- OozinozFilter class, 295–296, 301–302
- oozinoz.mdb database, 36
- OozinozRocket class, 22–23, 350, 351
- OozinozSkyrocket class, 25–27, 352
 - code for, 27
- Operation, 139, 141, 169, 173, 197, 213–214, 220–221
 - abstract, 65, 69–70
 - defined, 62, 215, 435
- Operation patterns, 211–221
 - algorithms, 218–219
- COMMAND pattern, 213, 257–265
 - delegates, 216–217
 - exceptions, 217–218

- INTERPRETER pattern, 213, 267–280
 - polymorphism, 219–220
 - signatures, 215
 - STATE pattern, 213, 235–246
 - STRATEGY pattern, 213, 247–256
 - TEMPLATE METHOD pattern, 213, 223–233
 - Operations, 220
 - compared to signatures, 215
 - defined, 213–214
 - Ordered pair, 115
 - Organization/categorization of patterns, 4–5
 - OurBiggestRocket class, 364
 - override keyword, and constructor declarations, 381
- P**
- PaddedPanel object, 192
 - Parallel hierarchy, 176–177, 230, 372, 416
 - defined, 176, 435
 - Parametric equations, 41–43, 302
 - circle from, 41
 - defined, 435
 - Parent node, trees, 54
 - Parse() method, 165
 - Parsers, 163–166, 168, 279, 279–280, 280, 341, 383
 - defined, 279, 435
 - Path, 418
 - defined, 54, 435
 - Pattern Hatching: Design Patterns Applied* (Vlissides), 346
 - Pattern Language, A: Towns, Buildings, Construction, A* (Alexander), 1
 - Patterns:
 - categorizing by intent, 5
 - defined, 1, 435
 - weaving into code, 344–345
 - Persistent storage, 28, 33, 206–207, 397–398
 - defined, 206–207, 435
 - PhysicalRocket class, 21–22, 25–26
 - code for, 21
 - PhysicalRocket objects, 350
 - PictureBox class, 124, 372–373
 - PictureBoxProxy class, 123–125
 - constructor, and PictureBox object, 125–126
 - design, 126
 - PictureBoxProxy object, 373
 - Plant() method, ExampleMachine class, 56
 - plant object, 54
 - PlotPanel class, 43, 95
 - as part of the UserInterface namespace, 45–46
 - refactored, 45
 - PlotPanel2 class, 305
 - Plus sign (+), and instance variables/methods, 423
 - Point struct, 156–157
 - Polymorphism, 219–220, 362, 404
 - defined, 435
 - Postorder traversal, 316–317
 - defined, 316–317, 435
 - Preorder traversal, 316–317
 - defined, 316–317, 435
 - PrettyVisitor class, 340
 - PrinterManager class, 364
 - PrinterSpooler class, 364
 - PrintIndentedString() method, 339
 - private access, 78, 363
 - Process class, 80, 363
 - ProcessComponent class, 321–323, 337, 340, 414
 - GetStepCount() method, 61
 - ProcessComposite class, 61
 - Processes namespace, 324
 - ProcessStep class, implementation of GetStepCount() method, 62
 - Prolog, 3
 - PromotionAdvisor class, 252–253
 - Properties, 14
 - defined, 435
 - and interfaces, 14–15, 16
 - PropertyHolder class, 103–104

- constructor, 105
 - and OBSERVER pattern, 104
 - protected access, 78, 80
 - protected internal access, 78
 - PROTOTYPE pattern, 160, 191–196
 - advantages of, 394
 - defined, 191, 196, 197
 - disadvantages of, 394
 - intent of, 193
 - object creation, 193
 - prototypes as factories, 191–193
 - prototypical instance variables, 192
 - prototyping with clones, 193–196
 - Proxy objects, interface, 123
 - PROXY pattern, 81, 123–138, 214, 410
 - classic example of, 123
 - data proxy, 127–132
 - DECORATOR pattern compared to, 309
 - defined, 123
 - remote proxies, 132–137
 - simple proxy, 123–127
 - public access, 78
- Q**
- QuickFuse class, 382
- R**
- RakeVisitor class, 336
 - Random case, defined, 436
 - RandomAdvisor class, 252–253
 - React() method, Curious class, 365
 - Recommend() method, 250–252, 252
 - RecordWipMove() method, 86, 363
 - Refactor, defined, 436
 - Refactoring, 109
 - code, 37
 - and consolidation of mediator into a separate class, 113
 - to FACADE, 38
 - to STATE pattern, 239–244
 - to STRATEGY pattern, 250–255
 - to TEMPLATE METHOD pattern, 230–232
 - Refactoring (Fowler), 369
 - Refactoring: Improving the Design of Existing Code (Beck/Fowler), 287
 - Reflection, defined, 157, 436
 - Regex object, 165, 383
 - Rel8 engine, 248–249
 - Relation, defined, 436
 - Relation of tubs and machines, 115
 - Relational consistency, guaranteeing, 117
 - Relational integrity, 115
 - Relational integrity mediators, 114–121
 - Relationally consistent, 115–116
 - defined, 115
 - Remote Method Invocation (RMI), 132
 - Remote proxies, 132–137
 - RequiredClass class, 24
 - RequiredMethod() method, 19
 - Reservation class, 166, 167, 361
 - ReservationBuilder class, 164, 166, 384
 - ReservationParser class, constructor, 164–165
 - Responsibility, controlling with accessibility, 77–80
 - Responsibility patterns, 75–153
 - defined, 75
 - CHAIN OF RESPONSIBILITY pattern, 81, 139–146
 - FLYWEIGHT pattern, 81, 147–153
 - intent of, 81
 - MEDIATOR pattern, 81, 109–122
 - OBSERVER pattern, 81, 89–108
 - ordinary responsibility, 75–77
 - PROXY pattern, 81, 123–138
 - SINGLETON pattern, 81, 83–87
 - Responsible property, 140
 - Restore() method, VisMediator class, 207
 - Return type, constructors, 381
 - ReturnInterior property, ComponentEnumerator class, 326
 - reworkOrFinish subprocess, 60

- Rocket table, 30
- RocketHome () method, 134–135
- Rocket.Thrust () method, 360
- Roman candle, 229
 - defined, 436
- Root, 55, 62, 144, 377
 - defined, 436
- Root node, trees, 54

- S**
- Save () method, VisMediator class, 207–208
- sealed keyword, and constructor declarations, 381
- SelectChanged () method, 112
- SelectedChange () method, 369
- Self, 3
- Sequence diagram, 370, 428, 434
 - defined, 436
- Sessions, 209, 363
 - defined, 206, 436
- SetDataBinding () method, 30, 31
- SetSimTime () method, 22
- SetTimeout () method, 68–69
- Shallow copy, 315, 387
 - defined, 395, 436
- Sharing flyweights, 150–153
- ShellAssembler class, 390
 - GetPlanner () method, 231
- ShellProcess class, 324–325
 - Make () method, 62
- Shells, *See* Aerial shells:
- Show () method, 355
- ShowBallistics class, 95, 97, 98–99, 101, 103, 367
- ShowBorrowing class, 12–13, 348
- ShowClient.cs, 136–137
- ShowConcurrentFor program, 313–314
- ShowConcurrentWhile program, 313
- ShowFilters class, 299
- ShowFlight class, 38–39, 109
 - calculation in OnPaint () code, 39
 - CreateGroupBox () method, 40
 - Main () method, 39–40, 46
 - OnResize () method, 39
 - refactoring code, 43
- ShowFlight2 class, code for, 46
- ShowInternal.cs, ShowAccessibility directory, 80
- ShowProperties class library, Oozinoz, 14
- ShowProxy class, 123
- ShowReflection class, 157, 383
- ShowRocketsFromTable class, 37
- ShowStructs class, 157, 382–383
- Shutdown () method, 66–67
- ShutDown () method, 228–230, 261–262
- ShutdownCommand class, 270
- Signatures, 220
 - compared to operations, 215
 - defined, 215, 436
- Simple Object Access Protocol (SOAP), 135–136, 208–209
 - defined, 436
- SimpleStreamWriter class, 293–294
- Sin class, 304
- SINGLETON pattern, 81, 83–87
 - categorizing, 83
 - code, 87
 - and distribution of responsibility, 109
 - intent of, 83
 - mechanics of, 83–84
 - recognizing, 86–87
- Singletons:
 - lazy-initializing, 84–85, 87, 363
 - and threads, 84–86
- Skyrocket class, 25–26
- Sliders, 94
- SlideScroll () method, 368
- Smalltalk, 3, 9
 - interface construct, 9
- Smells, 286–287
- SOAP, *See* Simple Object Access Protocol (SOAP):
- Solutions, 347–416
- Sontag, Susan, 217

- Sort () methods, 224–226
 - Sorting algorithms, 223–226
 - SQL Server, arguments in favor of writing code specific to, 360
 - SqlDataReader class, 71, 127
 - Star press, 68–69, 139, 226–229, 261–262, 278–279
 - defined, 436
 - StarPress class:
 - CreatePlanner () method, 390
 - GetPlanner () method, 232
 - planner attribute, 232
 - StarPressController class, 66, 67
 - Stars, 59, 68–69, 139, 176, 227, 229–230, 306, 308, 436
 - defined, 436
 - Start () method, StarPressControl-ler class, 66
 - StartMachine () method, FuserCon- troller class, 66
 - StartupCommand class, 272
 - State, 83, 98, 137, 147, 160, 197–198, 206, 209
 - defined, 235, 436
 - modeling, 235–239
 - STATE pattern, 213, 235–246, 372
 - DECORATOR pattern compared to, 309
 - making states constant, 244–246
 - modeling states, 235–239
 - refactoring to, 239–244
 - State(s):
 - defined, 235
 - handling transitions between, 246
 - States:
 - UML, 429–430
 - Static method, defined, 436
 - Status () method, 242–243
 - StopAll () method, 52–53, 357
 - StopMachine () method, 362
 - FuserController class, 66
 - Stop () method, StarPressControl-ler class, 66
 - Strategy:
 - defined, 247, 436
 - multiple strategies, and complexity of code, 247
 - STRATEGY pattern, 213, 247–256, 372, 404–405
 - comparing STRATEGY and STATE, 255
 - comparing STRATEGY and TEM- PLATE METHOD, 256
 - DECORATOR pattern compared to, 309
 - intent of, 247
 - modeling strategies, 247–248
 - refactoring to, 250–255
 - Streams, 291–300, 291–301, 309, 344
 - defined, 291, 436
 - StreamWriter class, 292, 293
 - String.Compare () method, 165
 - Structured query language (SQL), 28, 36, 71, 260
 - defined, 437
 - Subclassing, 26–28
 - _subEnumerator variable, Componen- tEnumerator class, 319–320
 - Substance class, 148–149, 379
 - get- properties, 149
 - Substance2 class, 379
 - SwitchSpool () method, 70–71
 - Synchronized () methods, 312, 316, 413–414
 - System.Collections library, 57, 117, 312
 - System.Console class, 364
 - System.Delegate class, 92
 - System.EventHandler delegate, 216, 399
 - System.Reflection namespace, 420
 - SystemStartup class, 84
 - System.Windows.Forms library, 331
 - System.Windows.Forms package, 47
- T**
- TEMPLATE METHOD pattern, 213, 223–233, 261, 263, 405, 410
 - algorithm, completing, 226–229

- hooks, 229–230
 - refactoring to, 230–232
 - sorting algorithms, 223–226
 - Term hierarchy, 272–273, 279
 - Testing namespace, `TestProcess` class, 62
 - `Testing.dll` class library, 419
 - `TestProcess` class, 62
 - `TextBox` subclass, 100
 - `TextWriter` class, 364
 - Thick applications, defined, 109
 - `:this()`, 155–156
 - `:this()`, 381
 - Thread-safe iteration, 311–316
 - Thrust class, 96, 99
 - and `Tpeak` object, 103
 - Tier, 107
 - defined, 437
 - time instance variable, 22
 - `Timeout()` method, 243, 245
 - Title case, 299
 - defined, 437
 - `TitleCaseFilter` class, 297
 - Tokenize, 165
 - `Tool.Responsible`, 376
 - `TopSalesAssociate` class, 364
 - `ToString()` method, 213–214, 387
 - `Touch()` method, 238–239, 241, 243, 245
 - `Tpeak` class, 101, 103
 - `Tpeak` object, 102–103, 367
 - `Tpeak` property, `BurnRate` and `Thrust` classes, 96, 100
 - `tPeak` variable, 94–95, 99
 - `TpeakFunction` superclass, 96, 100
 - “Track bars”, 94
 - `TrackBar` class, 366
 - `TrackBar` object, `Scroll` event, 98, 100, 106
 - `TreeNode` class, 331
 - `TreeNodeVisitor` class, 331–332, 334, 336
 - Trees, 53–58
 - defined, 437
 - `try/catch` statement, 218
 - `Tub` class, 120
 - `GetHashCode()` and `Equals()` methods, 119
 - Location property, 118–119, 120
 - `tub.Location.IsUp()`, 409
 - `TubMediator` class, 371–372
 - Hashtable object, 119–120
 - `TubMediator` singleton object, 118
- ## U
- UI class, 46
 - making methods static, 46–47
 - Oozinoz, 29
 - UML, *See* Unified Modeling Language (UML):
 - UML Distilled* (Fowler), 4
 - UML notation, 4
 - UML state machine, 236
 - `UnforgivingBuilder` class, 168–169
 - Unified Modeling Language (UML), 421–430
 - class diagrams, 422–423
 - class relationships, 424–425
 - classes, 422–423
 - defined, 423, 437
 - delegates and events, 427–428
 - interfaces, 424–425
 - notation, 4
 - objects, 428–429
 - states, 429–430
 - Unified Modeling Language User Guide, The* (Booch), 4, 345
 - Uniform Resource Locator (URL), 133
 - defined, 437
 - Unload buffer, 284
 - `Unload()` method, 363
 - `UnloadBuffer` subclass, `Machine` class, 284
 - `UpperCaseFilter` class, 297
 - using statement, 259
 - Utilities, and facades, 46–47

V

ValueTextBox class:
 code, problem with, 100–101
 TextBox subclass, 100
 ValueTextBox object, Property-
 Changed() method, 368
 virtual keyword, and constructor decla-
 rations, 381
 Visit() method, 332–333, 342, 415
 Visit(:MachineComposite) method,
 333
 visited collection, 58
 VISITOR pattern, 289, 329–342
 alternatives to applying, 415–416
 controversy about, 341–342
 cycles, 337–341
 fragility of, 341–342
 mechanics of, 329–331
 ordinary visitor, 331–337
 Visual Studio, 355, 417–418
 defined, 437
 Visualization class, 263–264
 Visualization2 class, 207

W

Web Services Description Language
 (WSDL), 135
 WebMethod attribute, 134, 214
 WhileCommand class, 275, 278
 WhileCommand.cs, 408–409
 Wiggle() method, Interesting class,
 365
 _wipMoves variable, 85
 Wizard, 111, 353
 defined, 437
 Work in process (WIP), 85
 defined, 437
 WrapFilter class, 298–299
 WrapFilter.cs, 298
 Write() methods, 298
 Write(:char) method, 296
 WriteLine() method, and Limitin-

gReader proxy, 131
 wsdl.exe, 135, 136

X

XML, *See* Extensible Markup Language
 (XML)
 XmlTextReader class, 384