

# INDEX

## Symbols

- #Region directive, 116–117
- & (ampersand character)
  - for append, 188
  - for mnemonic, 159

## A

- About Box template, 109
  - abstraction, 18–19
  - Access (Microsoft)
    - databases, 91. *See also* databases
    - connecting to, 444–445, 480
    - QueryDefs versus stored procedures, 457
    - versus SQL Server, 444
  - ActiveMdiChild
    - property, 208–209, 211
  - ActiveX, 1
  - ActiveX Data Objects (ADO), 1
  - actors. *See* users
  - Add Connection dialog, 445–447
  - Add Database Reference dialog, 474
  - Add New Item dialog, 110, 112, 119
  - Add New Test dialog, 334
  - Add Reference dialog, 139, 364
  - AddHandler statement, 175–180
  - AddressOf operator, 176
  - ADO.NET, 82, 440, 483
  - ampersand character (&)
    - for append, 188
    - for mnemonic, 159
  - Anchor property, 157
  - AndAlso operators, 177
  - app.config file, 481–482
  - application
    - building, 119–121
    - definition, 101
    - running, 121–124
  - application events, 134, 135–137
  - application frameworks
    - building, 69–70
    - classes
      - business logic layer, 71
      - data access layer, 71
      - presentation layer, 70
    - designing, 68–69
  - application properties, 124–125
  - Application Type, 129
  - assembly information, 129–131
  - Assembly Name, 126
  - Authentication Mode, 132
  - Enable Application Framework, 131
  - Enable XP Visual Styles, 131
  - Icon, 129
  - Make Single Instance Application, 131
  - Root Namespace, 126–128
  - Save My.Settings on Shutdown, 131, 216
  - Shutdown Mode, 132–133
  - splash screens, 133, 134
  - Startup Form/Startup Object, 129
- application settings, 213–223
  - Application tab, Project Designer, 125–137
  - Application Type application property, 129
  - ApplicationException, 239, 243, 269–271
  - architecture
    - SOA (Service-Oriented Architecture), 67–68
    - three-tiered, 64–67

- ArgumentException, 239
- ArgumentNullException, 239
- ArgumentOutOfRangeException, Exception, 239
- arrays, reference types, 7
- assemblies
  - assembly information, 129–131
  - versus applications, 101
- Assembly Information dialog, 129–130, 133
- Assembly Name application property, 126
- Assert class, unit testing, 338–339
- attributes, 295. *See also* properties
- Authentication Mode application property, 132
- auto-completion, 6
- B**
- BackColor property, 170–171
- backing variable, 271
- base classes, 15–16, 180. *See also* base form classes
  - building, 296–299
  - inheriting, 300–302
  - overriding members, 302–303
  - versus interfaces, 17
- base form classes, 70, 181. *See also* base classes
  - building, 181–184
  - calling code, 187–189
  - inheriting, 182–184
    - in existing forms, 185
    - in new forms, 185–187
  - loading automatically, 183
- behaviors. *See* methods
- binding. *See* object binding
- BindingList class, 348–349, 398, 437
- BindingNavigator component, 377
- BindingSource component, 361, 370–372, 377–378, 386, 396–397, 404–405, 407–408, 412–415, 422
- DataSource property, 371–373
- Blank Solution template, 106
- BlinkStyle property, 422
- BODT (business object data transfer) classes, 66
- Boolean data type, 7, 279
- boxed value types, 7
- boxing, 7
- BrowsableAttribute, 369, 433–434
- Brush objects, 223–224
- bug-free code, 88
- Build option, projects, 121
- business logic layer, 64–66, 70–71, 73–74
  - handling data, 83–85
- business object classes. *See* classes; objects
- business objects. *See* classes; objects
- Button class, 10
- Button controls, 171
- ByRef keywords, 286–287
- ByVal keywords, 286–287
- C**
- Cancel property, 201
- Case Else, 90
- casting, 169
- catching exceptions, 241–244. *See also* exception handling
- .cd extension, 309
- child classes, 15–16
- child forms. *See* MDI child forms
- Choose Data Source dialog, 445
- CInt function, 169
- Class-Responsibility-Collaboration (CRC) cards, 53
- Class Designer class diagrams
  - adding items, 310–312
  - classes, 311–312
  - creating, 308–309
  - methods, 285, 313–315
  - properties, 273, 313–315
  - relationships, 312, 352
- extension, 309
- Inheritance tool, 312–313
- OTB (Object Test Bench), 316–319
- Class Details window, 313–315

- class files
  - multiple classes, 269–271
  - partial classes, 160–162, 268–269
- class libraries, 101
- Class Library projects, 121
- Class Library template, 107–108
- Class template, 254, 296
- Class View window, 316
- classes, 5, 9–14, 252. *See also* base classes; base form classes; collection classes; derived classes; derived form classes; master/detail item classes; objects; partial classes
  - adding to projects, 254–255
  - application framework classes
    - business logic layer, 70–71
    - data access layer, 70–71
    - presentation layer, 70
  - code structure, 257–259
  - constructors, 260–263
  - creating, 252–253
  - destructors, 264–267
  - documenting, 256–257
  - files, 268–271
  - identifying, 52–55, 253
  - inheritance, 15–16, 20–21, 182–187, 296–303
  - Initialize events, 261
  - instantiating objects, 10, 14, 252, 259–260
  - interfaces, 17–18, 209–213
  - methods, 53
    - creating, 285–286
    - defining, 283
    - documenting, 287–289
    - obsolescing, 294–296
    - overloading, 289–291
    - parameters, 283–287
    - return type, 284
    - shared, 291–294
  - namespaces, 18, 126–128, 139–141
  - OOA, 22–24
  - OOD, 24–25, 39–98
  - OOP, 25–26, 39–502
  - organizing code structure, 257–259
  - polymorphism, 21–22
  - properties, 52
    - accessing, 276–278
    - binding, 358–416
    - defining, 271–274
    - documenting, 275–276
    - nulls, 278–282
    - Property statements
      - versus private variables, 274–275
  - stateful versus stateless, 282
  - static class data/
    - methods, 14
  - Terminate events, 265
  - versus objects, 253
- Clean option, projects, 121
- Click event, Button
  - controls, 166, 171
- Close method, 200, 208
- CLR (Common Language Runtime)
  - methods, 459
- code coverage, 343
- Code Editor
  - automatic name change
    - from Solution Explorer, 155, 255
  - event handlers, 137, 167
  - List Members box, 163–164, 257, 276, 288, 300
  - Task List, 118–119
  - regions, 117–118
  - syntax errors, 120
  - viewing code
    - classes, 254
    - designer-generated
      - partial classes, 160–161
    - forms, 160
  - XML comments
    - classes, 256–257
    - event handlers, 178
    - forms, 162–164
    - methods, 178, 287–289
    - properties, 275–276
- Code Snippet Picker, 320–322, 327
- code snippets, 320
  - creating, 327–330
  - inserting, 320–322
  - locating, 325
  - managing, 324–325

- replacements, 322–323, 329–330
- Code Snippets Manager, 330, 324–325
- collection classes, 348–350.
  - See also* classes
  - combo boxes, 397–398
  - object binding to grids, 385–387
- columns, 447–449
  - defining, 78
  - Identity columns, 451
  - primary keys, 78–79, 450–451
  - saving, 448–449
  - system columns, 78, 453–454
- combo boxes
  - collection classes, 397–398
  - DataTable, 397–400, 409
  - object binding, 394
    - data entry, 395–396, 402–406
    - programmatic use, 394, 407–416
- ComboBox controls, 394
  - Anchor property, 157
  - DisplayMember property, 396, 404, 407–408, 412
  - SelectedValue property, 396, 404–405, 407, 412
  - SelectedValueChanged event, 171
  - sharing event handlers, 177
  - ValueMember property, 396, 404–405, 407–408, 412
- comments
  - List Members box, 163–164, 257, 276, 288, 300
  - standards, 90
  - Task List, 118, 119
  - XML comments
    - classes, 256–257
    - event handlers, 178
    - forms, 162–164
    - methods, 178, 287–289
    - properties, 275–276
- Common Language Runtime (CLR)
  - Compile tab, Project Designer, 144
- composite controls, 195
- composite formatting, 236
- concurrency requirements, 71
- connect-the-dots binding, 370–374
- Connection objects, 484–486
- connection strings, 479–482
- Console Application
  - projects, 121
- Console.WriteLine method, 236
- constants
  - enumeration, 297
  - standards, 90
  - for stored procedure names, 488–490
- construction-phase, design.
  - See* Strategies for Construction
- constructors, 260–263. *See also* destructors
- ContainsKey method, 426
- control arrays, 173
- Control data type, 169
- controller form UI
  - approach, 190–191
- controls
  - adding to forms, 153–157, 375–376
  - binding, 370–420
  - composite controls, 195
  - event handlers
    - coding, 179–180
    - for controls at runtime, 175–179
    - for multiple controls, 172–175
    - for one control, 166–172
  - naming conventions, 154
  - Properties window, Events view, 168
  - smart tags, 156
  - user controls, 194–196
  - wrapped, 194–195
- conversion functions, 169
- CRC (Class-Responsibility-Collaboration) cards, 53
- Create New SQL Server Database dialog, 442–443
- Create Unit Tests dialog, 332–333

- CRUD operations, 82
- CStr function, 169
- CType function, 169
- D**
- DAC (data access class), 71, 476
  - calling
    - from business objects, 488–492
    - object collections, 493–495
    - stored procedures, 482–487
  - creating, 477–478
  - saving data, 496–498
- data access class. *See* DAC
- data access layer, 64–67, 70–71, 73–74, 83–85
- data binding. *See* object binding
- data design, 42, 75–76.
  - See also* GUIDS Methodology
  - building models, 80–81
  - data access strategies, 82
  - data storage mechanisms, 76
  - database terminology, 76–77
  - performance tuning, 81–82
  - RDBMS, 77–80
  - three-tiered architecture, 83–85
- Data Source Configuration Wizard, 363–365
- Data Sources window
  - Add New Data Source link, 362–363
  - object data sources
    - adding properties, 366–367
    - binding to combo boxes, 394–416
    - binding to existing controls, 370–374
    - binding to grids, 384–393
    - binding to new controls, 374–376
    - binding to radio buttons, 381–383
    - building forms from, 376–379
    - creating, 362–365
    - deleting properties, 367
    - hiding properties, 368–369
- Database project, 473–476
- databases. *See also* RDBMS; stored procedures
  - connecting to, 442–447
  - connection strings, 479–482
  - creating sample data, 456
  - defining, 76–82, 442–443
  - diagrams, 80–82, 454–455
  - scripts, 473–475
  - tables
    - creating, 447–449
    - primary keys, 78–79, 450–452
    - saving, 448–449
    - system columns, 453–454
    - terminology, 76–77
- DataBindings collection, 370, 372, 405
- DataReader, 84, 483
- DataRow objects, 494–495
- DataSet objects, 83–85, 483
- DataSource property, 371–373
- DataStateChanged method, 298–299
- DataTable, 83–85
  - binding to combo boxes, 397–400, 409–413
  - creating in code, 411
  - objects, 405, 483, 485, 489–494
- date data type, 279–281
- DbProviderFactory, 476, 478, 486
- DCOM (Distributed Component Object Model), 1
- Debug class, 14, 141, 287
  - Print method, 236
  - WriteLine method, 274, 279
- debugging applications, 123–124
  - bug-free code, 88
- decimal data type, 7
- default form instances, 198
- defensive development, 233–235
- Define a Property code snippet, 322, 325
- delegates, 176
- deployment requirements, 72

- derived classes, 15–16, 180.
  - See also* child classes
- derived form classes, 181–183
- design. *See* data design;
  - goal-centered design;
  - GUIDS Methodology;
  - implementation-centered design;
  - OOD; strategies for construction; user interface design
- design patterns 25, 68, 73, 196, 261
- .designer.vb extension, 161, 185
- designer-generated
  - partial classes
  - inheritance, 185
  - viewing, 160–162
- destructors. *See also* constructors
  - defining, 264
  - Finalize method, 264
  - garbage collection, 264
  - IDisposable interface, 265–267
  - unmanaged resources, 264–267
- detail item classes. *See* master/detail item classes
- Dialog template, 109
- DialogResult property, 200
- Dictionary generic class, 425
  - ContainsKey method, 426
  - Count property, 427
- Dim keyword, 6–7, 199
- DirectCast method, 169, 170, 211
- DisplayMember
  - property, 396, 404, 407–408, 412
- Dispose method, 264
  - closing forms, 200
  - IDisposable interface, 265–266
- Distributed Component Object Model (DCOM), 1
- Dock property, 157
  - Panel controls, 155
- documentation
  - classes, 256–257
  - event handlers, 178
  - forms, 162–164
  - methods, 287–289
  - properties, 275–276
- domain experts, 19, 23
- domain models, 22–24
  - goal-centered design, 54–55
  - scenarios, 23
- Dotfuscator, 91
- double binding, 404
- DrawImage method, 227–228
- DrawToBitmap
  - method, 227–228
- E**
- Enable Application Framework
  - application property, 131
- Enable XP Visual Styles
  - application property, 131
- encapsulation, 5, 13, 19–20
- Enter events, 167, 179
- Enum keyword, 297
- enumeration, 297
- Error List window, 120
- ErrorProvider, 420–422, 424
- event handlers, 166
  - AddHandler statement, 168, 175–176, 180
  - arguments, 169–170
  - coding, 179–180
    - for controls at runtime, 175–179
    - for multiple controls, 172–175
    - for one control, 166–172
  - RemoveHandler statement, 175
- Exception class, 239, 243, 271
- exception handling, 135
  - ApplicationException class, 239, 243, 269–271
  - catching exceptions, 241–244
    - expected and unexpected, 135
  - defensive development, 233–235
  - Exception class, 239, 243, 271
  - Finally blocks, 242
  - global, 135–137

- resource files, message text, 235–237
  - standards, 90–91
  - throwing exceptions, 238–240
  - Try/Catch block, 135, 214–243
  - Exception snippet, 269–271
  - Export Template Wizard, 113–114, 119
  - Extreme Programming (XP), 24, 41
- F**
- Façade design pattern, 68
  - Factory design pattern, 261–262, 284
    - factor classes, 262
    - factory method, 262, 292
  - feature sets, 47–49
  - fields, 77–78
    - foreign keys, 78
    - joins, 81
    - primary keys, 78–79, 450–451
  - Finalize method, 264–267
  - Finally blocks, 242
  - Find and Replace dialog, 117, 145
  - For statement, 208
  - foreign keys, 78
  - Form class, 160–162
    - Close method, 200, 208
    - DialogResult property, 200
    - DrawToBitmap method, 227–228
    - FormBorderStyle property, 157
    - FormClosing event, 201, 214, 221
    - Location property, 216
    - ShowDialog method, 198–200
  - Format method, 236
  - FormBorderStyle property, 157
  - FormClosing events, 201, 214, 221
    - Cancel property, 201
  - FormLocation settings, 216, 218
  - forms
    - classes, viewing, 160
    - closing, 200–201
    - code structure, 164–166
    - creating, 152–153
    - displaying
      - with custom instances, 199–200
      - with default instances, 198–199
    - documenting, 162–164, 178
    - drawing graphics, 223–225
    - fonts, 157
    - instances
      - custom, 199–200
      - default, 198–199
    - layouts, 60–61, 153–160
    - naming conventions, 153
    - partial classes, 160–162, 268–269
    - printing, 225–229
      - Print Preview, 229–231
      - Print Setup, 231–232
    - resizing, 157–158
    - Tab Order, 158–160
    - UI approaches, 57–61, 190–193
      - implementation, 193–197
      - visual elements, 153–160
  - Forms Designer
    - event handlers, 167
    - smart tags, 156
    - Tab Order feature, 158–159
  - FromKnownColor method, 171
  - FxCop, 91
- G**
- garbage collector, 264
  - generics, 279–281
    - collection classes, 348–349, 398, 425, 437
  - getter, 271
  - global exception handling, 135–137, 242
  - globally unique identifier (GUID), 130, 451
  - goal-centered design, 42–55. *See also* GUIDS Methodology
    - business object identification, 52–55
    - domain model, 54
    - feature sets, 47–49
    - goals, 44–45
    - scenarios, 50–51
    - use cases, 45–47
  - GotFocus event, 170
  - graphics, 223–225

- grids
    - object binding, 384–387
    - displaying related
      - properties, 390–393
    - modifying columns, 389–390
    - read-only grids, 389
    - smart tags
      - disabling edits, 389
      - modifying grid columns, 389–390
  - GUID (globally unique identifier), 130, 451
  - GUIDS Methodology, 42.
    - See also* data design;
    - goal-centered design;
    - implementation-centered design;
    - OOD; strategies for construction;
    - user interface design
- H**
- HACK tokens, 118
  - Handles clause, 167–168
    - controls sharing event handlers, 172–175
  - handling exceptions, 135
    - ApplicationException class, 239, 243, 269–271
    - catching exceptions, 135, 241–244
    - defensive development, 233–235
    - Exception class, 239, 243, 271
    - Finally blocks, 242
    - global, 135–137
    - resource files for message
      - text, 235–237
    - standards, 90–91
    - throwing exceptions, 238–240
  - HasValue property, 281
- I**
- Icon application
    - property, 129
  - IDataErrorInfo interface, 424, 432–434
  - Identity columns, 451
  - IDisposable interface, 224
    - Dispose method, 265–266
    - Implements statement, 265
    - SuppressFinalize method, 266–267
    - using statement, 267
  - image library, 204
  - IMDICHild interface, 70
  - implementation-centered design, 42, 63–74.
    - See also* GUIDS Methodology
  - application frameworks, 68–71
  - conversion to
    - implementation models, 72–74
  - implementation
    - requirements, 71–72
  - SOA (Service-Oriented Architecture), 67–68
  - three-tiered architecture, 64
    - business logic layer, 64–66, 70–71, 73–74
    - data access layer, 64–67, 71
    - presentation layer, 64–66, 70
  - Implements statement, 17, 210–211, 265
  - Import and Export Settings Wizard, 145–147
  - importing namespaces, 140–141
  - Imports statement, 141
  - inheritance, 15–16, 20–21
    - base business object classes, 296–303
    - base form classes, 182–184
      - in existing forms, 185
      - in new forms, 185–187
    - Form class, 162
    - versus templates, 181
    - visual inheritance, 181
  - Inheritance Picker dialog, 186
  - Inheritance tool, Class Designer, 312–313
  - Inherited Form template, 186
  - Inherits keyword, 16
  - Inherits statements, 187
  - Initialize events, 261
  - InstallShield, 91
  - instances. *See* classes,
    - instantiating objects
  - Integer data type, 6–7, 169
  - Intellisense, 6, 120, 163–164, 198, 211, 257, 276, 288, 291, 300

interface, 17–18, 209–213  
 versus base class 17  
 Interface template, 209  
 interim deliverables, 31, 48  
 Interval property, 6  
 InvalidOperationException, 239, 242  
 Invoke Method dialog, 317  
 IsDirty property, 298  
 IsMdiContainer  
 property, 203  
 IsNullOrEmpty  
 method, 189

**J – L**

joins, 81  
 Label controls, 154  
 Tab Order, 159  
 Text property, 159  
 UseMnemonic  
 property, 159  
 Leave events, 170–171  
 LineJoin property, 224–225  
 List generic class, 395, 408  
 List Members box, Code  
 Editor, 163–164, 257,  
 276, 288, 300  
 Load event, 166, 175, 177,  
 221, 373  
 base form class, 183  
 Location property, 216  
 logical tiers, architecture, 64  
 Login Form template, 109  
 loosely coupled  
 applications, 67

**M**

Make Single Instance  
 Application  
 application property,  
 131  
 master/detail item classes,  
 343–344  
 detail collections, 348–350  
 detail item, 343–346  
 relationships, 343–344,  
 351–352  
 MDI (Multiple Document  
 Interface), 201  
 child forms  
 accessing, 207–212  
 creating, 205–207  
 implementing, 210–213  
 parent forms, 70, 201–205  
 UI approach, 191  
 MDI Parent Form  
 template, 191,  
 201–205, 226  
 MdiParent property,  
 205–206  
 members. *See* properties;  
 methods  
 MenuStrip controls, 204  
 MessageBox class, 14  
 Method Call Result  
 dialog, 318–319  
 methods, 3–8, 53  
 adding, 314  
 class inheritance, 16  
 classes, 10–13  
 creating, 283–286  
 defining, 53, 252, 283  
 documenting, 287–289  
 interfaces, 17–18

naming conventions  
 283–284  
 Object Test Bench,  
 316–319  
 obsolescing, 294–296  
 overloading, 289–291  
 parameters, 283–284,  
 286–287  
 recursive, 176–177  
 regions, 258–259  
 return type, 284  
 scenarios, 9  
 shared, 14, 291–294,  
 316–317  
 testing, 338–341  
 viewing, 313–315  
 Microsoft  
 Composite UI application  
 block, 249  
 Patterns and Practices,  
 197, 248  
 UI application  
 blocks, 197, 249  
 Microsoft Access databases.  
*See* Access databases  
 Microsoft SQL Server. *See*  
 SQL Server  
 mnemonics, 159  
 modal forms, 198–200  
 Model-View-Controller  
 (MVC) design  
 pattern, 196–197  
 modeless forms, 198–200  
 Modify Connection, Server  
 Explorer, 444, 479  
 module, 14  
 multi-pane UI  
 approach, 192

- Multiple Document Interface. *See* MDI1
  - MVC (Model-View-Controller) design pattern, 196–197
  - My keyword, 198
  - My.Forms objects, 198–199
  - My.Settings object, 213, 216
  - My.Resources, 236
  - MySettings object, 218–223
  - MySQL databases. *See also* databases.
    - connecting to, 444–445
    - versus SQL Server, 444
- N**
- N-tiered architecture, 64
    - business logic layer, 64–66, 71
    - data access layer, 64–67, 71
    - presentation layer, 64–66, 70
    - transferring data, 83–85
  - Namespace keyword, 127–128
  - namespaces, 18, 126–128
    - importing, 139–141
  - naming conventions, 90
    - controls, 154
    - forms, 153
    - method parameters, 284
    - methods, 283
    - private variables, 272
    - project items, 110–111
    - projects, 102, 107
    - properties, 272
    - solutions, 102
  - navigation techniques, user interface design, 60
  - .NET Framework, 1, 10
  - New keyword, 6–8, 13–14, 199–200, 260–261
  - New method, 260
  - New Project dialog, 101–103, 106, 112, 114–115
  - New Test Project dialog, 333
  - normalization, 79–80
  - Nullable class, 279–281
    - HasValue property, 281
    - Value property, 281
  - NUnit tool, 91, 331
- O**
- object-orientation. *See* OO
  - object-oriented analysis. *See* OOA
  - object-oriented design. *See* OOD
  - object-oriented programming. *See* OOP
  - object binding, 84, 358–361
    - to combo boxes, 394, 420
    - creating a form, 376–379
    - error handling, 420–422
    - to existing controls, 370–374
    - to grids, 384–390
      - related properties, 390–393
    - to new controls, 375–376
    - to radio buttons, 381–383
    - validating properties, 422–424
    - validation class, 424–434
    - versus data binding, 360
  - object data source, 362. *See also* object binding
    - adding properties, 366–368
    - creating, 362–366
    - hiding properties, 368–369
  - Object data type, 7, 169–171
  - Object Test Bench (OTB), 316–319
  - objects, 3–8, 52, 252. *See also* classes.
    - abstraction, 18–19
    - binding 358–359
    - class instances, 14
    - classes, 10–14
    - constructors, 260, 262–263
    - destructors, 264–267
    - domain models, 54–55
    - encapsulation, 5, 13, 19–20
    - instantiating, 252, 259–260
    - methods, 4–8, 53, 252, 283–296
    - populating, 488–492
    - properties, 3–8, 52, 252, 271–282
    - remoting, 270
    - saving, 496–499
    - scenarios, 9
    - state, 296–299
    - validating, 421–435
    - variables, 6–8, 13
      - casting, 169–170

- versus classes, 253
- obsolescing methods, 294–296
- Obsolete attribute, 295
- ObsoleteAttribute
  - attribute, 295
- On Error statement, 241
- OO (object-orientation), 1
  - benefits, 26–33
  - overview, 2–3
  - process, 22–26
  - reference types, 7–8
  - scenarios, 19
  - value types, 7
- OOA (object-oriented analysis), 22–24
- OOD (object-oriented design), 24–25, 39–98.
  - See also* data design;
  - goal-centered design;
  - GUIDS Methodology;
  - implementation-centered design;
  - strategies for construction; user interface design
- OOP (object-oriented programming), 25–26, 39–502
- operations. *See* methods
- Option Compare, 144
- Option Explicit, 90, 142–143
- Option statement, 145
- Option Strict, 90, 143–144, 162, 269
- Options (Tools) dialog, 104–105, 142–145
- Oracle databases, 91. *See also* databases
  - connecting to, 444
  - stored procedures, 457
- OrElse operators, 177
- OTB (Object Test Bench), 316–319
- Output window, 121
- overloaded methods, 261, 289–291
- Overriding methods, 302–303, 428
- P**
- Paint events, 223
- Panel controls, 154, 377
  - Dock property, 155
  - event handlers, 176
- param tags, XML comments, 178, 288
- parameterized constructors, 261
- parameters
  - methods, 283–287
  - overloaded methods, 289–291
- parent classes, 15–16
- parent forms. *See* MDI parent forms
- partial classes, 160–162, 268–269. *See also* classes
  - viewing, 160–162
- Partial keyword, 162, 268
- partial types. *See* partial classes
- patterns,
  - Facade pattern, 68
  - Factory pattern, 261–262
- Patterns and Practices
  - group, Microsoft, 197
- Pen objects, 223–225
- performance tuning, 81–82
- performance requirements, 71
- physical tiers,
  - architecture, 64
- polymorphism, 21–22
- presentation layer,
  - architecture, 64–66
  - application framework classes, 70
  - transferring data, 83–85
- primary keys, 78–79, 450–452
  - Identity columns, 451
- primitive data types, 7
- Print method, 226
- PrintDialog controls, 231–232
- PrintDocument controls, 226, 229–230
- printing forms, 225–229
  - Print Preview, 229–231
  - Print Setup, 231–232
- PrintPage event, 226
- PrintPreviewDialog controls, 230
- Private keyword, 12–13, 20, 199, 262, 271–272, 277, 297
- private properties, 165, 277
- private variables, 271–273
  - naming conventions, 272
  - versus Property statements, 274–275
- procedural models, 27
- ProductName property, 393

- Project Designer
    - Application tab, 125–137
    - Compile tab, 144
    - References tab, 138–140, 477
    - Resources tab, 235
    - Settings tab, 131, 213–218, 480–481
  - project group. *See* solutions
  - projects, 10
    - adding to solutions, 106–109
    - naming conventions, 102, 107
    - project items, 101
      - adding to projects, 109–111
      - naming conventions, 110–111
      - Solution Explorer, 104–105, 111
    - referencing within solutions, 137–142
    - startup projects, 121–123
    - templates, 110–116
      - Class Library, 107–108, 112
      - creating, 112–119
      - Windows Application, 102–103, 112, 121
    - types, 102–103
  - properties, 3–8, 52
    - accessability, 276–278
    - creating, 271–274
    - defining, 52, 252
    - documenting, 275–276
    - naming conventions, 272
    - nulls, 278–282
    - scenarios, 9
    - testing, 337–338
  - Properties window, 370–371
    - Events view, 168
  - Property statements, 13, 271–274, 367, 382, 402, 424
    - accessability, 276–278
    - versus public variables, 274–275
  - Protected keyword, 188–189, 297
  - prototype, 28–29, 56, 62
  - Public Property
    - statement, 275
- Q – R**
- Query Builder, dialog, 463–467
    - insert operations, 471–472
  - QueryDefs versus stored procedures, 457
  - Radio Button controls, 381–383
  - RDBMS (relational database management system), 76–77. *See also* data design.
  - ReadOnly keyword, 277
  - Rebuild option, projects, 121
  - record, 77
  - recursive methods, 176–177
  - reference types, 7
    - boxing, 7
    - converting to value types, 7
    - nullable, 278
    - object variables, 7, 8
    - strings, 8
  - References tab, Project Designer, 138–139, 477
  - regions, 90, 116–117
    - #Region directive, 117
    - classes, 164–165, 258–259
    - collapsing and expanding, 117
    - forms, 164
    - searching, 117
    - templates, 116–117, 164–165
  - regression testing, 331
  - relational database
    - management system. *See* RDBMS
  - relationships, master/detail
    - classes, 343–344, 351–352
  - remarks tags, XML
    - comments, 163, 256–257
  - remoting, 270
  - RemoveHandler
    - statement, 175
  - replacements in code snippets, 322–323, 329
  - requirements, 44
  - resource files, message text, 235–237
  - resources. *See* properties
  - Resources tab, Project Designer, 235
  - responsibilities. *See* methods
  - Return statements, 273, 284

- returns tags, XML
  - comments, 275–276
- Root Namespace
  - application property, 126–128
- Run Stored Procedure dialog, 461
- S**
- Save My.Settings on Shutdown
  - application property, 132, 216
- save stored procedure, 468–470
- scalability, 72
- scenarios, 9
  - abstraction, 19
  - goal-centered design, 50–51
  - OOA, 23
  - UML, 51
  - versus use cases, 45
- schedules. *See* strategies for construction
- scope, variable, 90
- SDI (Single Document Interface), 190
- security requirements, 71
- SelectedValue property, 396, 404–405, 407, 412
- SelectedValueChanged
  - event, 171, 189, 408
- Serializable attribute, 270
- serialization, 270
- Server Explorer
  - connecting to a database, 442–447
  - creating sample data, 456
  - creating a SQL Server database, 442–443
  - database diagrams, SQL Server 2005, 454–456
  - stored procedures, SQL Server, 457–462
  - tables, SQL Server
    - creating, 447–449
    - primary keys, 78–79, 450–452
    - saving, 448–449
    - system columns, 453–454
- Service-Oriented Architecture (SOA), 67–68
- services. *See* methods; SOA
- Set statement, 259
- setter, 271
- Settings tab, Project Designer, 131
- SettingsKey property, 219–222
- shared data, 14
- Shared keyword, 14, 292–293
- shared methods, 14, 291–294, 316–317
- shell UI approach. *See* multi-pane UI approach
- short-circuiting, 177
- ShowDialog method, 198–200
- Shutdown Mode application property, 132–133
- Single Document Interface (SDI), 190
- single instance applications, 131
- Singleton design pattern, 25
- smart tags, 156–157
  - grids, 389–390
- Snippet Editor, 327–330
- SOA (Service-Oriented Architecture), 67–68
- Solution Explorer, 103–108
  - projects
    - adding to solutions, 107–108
    - adding solutions to, 106–108
    - project items, 111
    - solution folders, 108–109
  - Show All Files option, 161–162
- solutions, 120–124
  - Blank Solution template, 106
  - creating, 100–106
  - folders, 108–109
  - naming conventions, 102
  - solution folders, 108–109
  - Visual Studio 2003, 105
- SourceSafe, 91
- splash screen application property, 133–134
- Splash Screen template, 109, 111, 133
- SQL Server 2005
  - Express Edition, 440–442
  - compatibility with other editions, 441
  - versus SQL Server, 441

- Windows
    - Authentication, 442
  - installing multiple
    - versions, 441
  - versus other
    - applications, 444
    - versus SQL Server 2000, 455
  - SQL Server databases,
    - 91. *See also* databases.
    - connecting to, 444–445
    - stored procedures, 457
    - T-SQL statements, 463–467
  - SqlDataAdapter object, 485
  - SqlParameter object, 486–487
  - standards. *See* strategies for construction
  - Startup Form/Startup
    - Object application property, 129
  - startup projects, 121–123
  - state, object, 296–299
  - stateful versus stateless
    - classes, 282
  - static class data. *See* shared data
  - static class methods. *See* shared methods
  - StatusStrip controls, 204
  - stored procedures, 82, 457–458
    - calling with DAC methods, 482–487
    - creating, 458–461, 468–472
    - with scripts, 475
  - database projects, 474–475
  - definition, 457
  - displaying results, 462
  - executing, 468–470
  - storing string names as constants, 488, 490
  - T-SQL statements, 463–467
  - testing, 461
  - versus QueryDefs, 457
  - versus SQL statements, 458
  - stories. *See* scenarios
  - Strategies for Construction, 42, 85–86. *See also* GUIDS Methodology
    - development plans, 86–88
    - schedules, 27–28, 92–93
    - selecting tools, 91–92
    - standards, 89–91
  - String class
    - Format method, 236
    - IsNullOrEmpty method, 189
  - string data types, 7–8, 169
  - StringBuilder class, 428
  - Structured Exception Handling (SEH). *See* exception handling
  - Sub Main, 129
  - summary tags, XML
    - comments, 163–164, 256–257, 275–276, 288
  - SuppressFinalize method, 266–267
  - syntax errors, 120
  - syntax standards, 90
  - system columns, 453–454
  - System namespace, 18
  - System.Data namespace, 400, 410, 477
  - System.Data.Common namespace, 478
  - System.Data.DataTable namespace, 400, 410
  - System.Data.Odbc namespace, 478
  - system.Data.OleDb namespace, 478
  - System.Data.SqlClient namespace, 478
  - System.Diagnostics namespace, 141
  - System.Windows.Forms namespace, 16, 18, 162, 251
- ## T
- T-SQL (Transact-SQL)
    - statements, 458–467
    - insert operations, 471–472
  - Tab Order feature, 158–160
  - Table Designer, 448–449
  - TableLayoutPanel
    - controls, 224
  - tables, 76–81
    - columns, 77–78, 447–449
    - creating, 447–449
    - foreign keys, 78
    - Identity column, 451
    - joins, 81
    - normalization, 79–80
    - primary keys, 78–79, 451–452

- records, 77
  - saving, 448–449
  - system columns, 453–454
  - Task List window,
    - 118–119
  - TDD (Test-Driven Development),
    - 88, 331
  - Team System (Visual Studio), 91
  - templates
    - Blank Solution, 106
    - exporting, 113–115, 119
    - project items, 110–112
      - About Box, 109
      - Class, 254, 296
      - creating templates,
        - 116–119
      - Dialog, 109
      - Login Form, 109
      - Splash Screen,
        - 109, 111, 133
    - projects, 111–115
      - Class Library, 107–108, 112
      - Windows Application,
        - 102–103, 112, 121
    - types, 103
    - versus inheritance, 181
  - Terminate events, 265
  - Test-Driven Development (TDD), 88, 331
  - Test Project template, 336
  - Test Results window,
    - 342–343
  - testing. *See also* unit testing/
    - unit tests
    - regression testing, 331
    - static methods, 316–317,
      - 338–341
    - stored procedures,
      - 461–462
  - TestMethod attribute, 339
  - Text property, 370–372
    - Button class, 10
  - TextBox control, 84,
    - 154–157, 370,
      - 375–377
    - event handlers, 167–168
      - Enter, 170
      - Leave, 171
      - sharing, 174
    - properties
      - Anchor, 157
      - BackColor, 171
  - three-tiered architecture, 64
    - business logic layer,
      - 64–66, 71
    - data access layer,
      - 64–67, 71
    - presentation layer,
      - 64–66, 70
    - transferring data layer to
      - layer, 83–85
  - Throw statement, 239
  - throwing exceptions,
    - 238–240
  - tiers, architecture, logical
    - and physical, 64
  - Timer object, 6
  - TimeStamp type, 453
  - TODO tokens, 117–118
  - ToolStrip controls, 204
  - ToString method, 428
    - casting variables, 169
    - Integer object, 6
  - Transact-SQL (T-SQL)
    - statements, 458–467
      - insert operations, 471–472
  - Try/Catch blocks, 135, 242
  - TryCast method, 169–170,
    - 211–212
  - typeof operator, 176–177
  - types, handling, 71, 398–405
- ## U
- UIs (user interfaces). *See*
    - Web-based UIs;
    - Windows-based UIs
  - UNDONE tokens, 118
  - UnhandledException event,
    - 135–137
  - Unified Modeling Language. *See* UML
  - Unit Test Wizard, 334–335
  - unit testing/unit tests,
    - 88, 330
    - Add New Test dialog, 334
    - creating, 332–337
    - code coverage, 343
    - coding, 337–341
    - Create Unit Tests dialog,
      - 332–333
    - methods, 338–341
    - New Test Project dialog, 333
    - NUnit, 91, 331
    - organizing, 341
    - properties, 337–338
    - running, 342–343
    - Test Project template, 336
    - test projects, 332–337
    - Test Results window,
      - 342–343

- Unit Test Wizard, 334–335
- units of work, 48–49
- unmanaged resources, 224, 264–267
- use cases, 45–47. *See also* scenarios
  - applications into feature sets, 47–49
  - scenarios, 50–51
  - UML, 51
  - versus scenarios, 45
- UseMnemonic
  - property, 159
- user controls, 194–196
- user interface design, 42, 55–56. *See also* GUIDS Methodology
  - conceptual design, 57–59
  - form layouts, 60–62
  - navigation techniques, 60, 189–193
  - prototypes, 62
- Using statement, 224–225, 267

## V

- validation
  - class, 424–435
  - clearing, 429
  - empty values, 423–424
  - ErrorProvider control, 421
  - length check, 430
  - null values, 423–424
- Value property, 281
- value tags, XML comments, 275–276

- value types, 7
  - Boolean, 7, 279
  - boxing, 7
  - converting to reference types, 7
  - date, 279–281
  - not nullable, 278
- ValueMember property, 396, 404–405, 407–408, 412
- variable scope, 90
- .vb extension, 161–162
- View Application Events
  - button, 134
- View menu
  - Class View, 316
  - Other Windows, Class Details, 313
  - Server Explorer, 442
- visual inheritance, 181
- Visual SourceSafe, 91
- Visual Studio, 1, 91
  - Add Connection dialog, 445–447
  - Add Database Reference dialog, 474
  - Add New Item dialog, 110, 112, 119
  - Add New Test dialog, 334
  - Add Reference dialog, 139, 364
  - Choose Data Source dialog, 445
  - Create New SQL Server Database dialog, 442–443
  - Create Unit Tests dialog, 332–333

- Data Source
  - Configuration Wizard, 363–365
- Error List window, 120
- Export Template Wizard, 113–114, 119
- Find and Replace dialog, 117, 145
- image library, 204
- Import and Export Settings Wizard, 145–147
- Inheritance Picker dialog, 186
- Intellisense, 6, 120, 163–164, 198, 211, 257, 276, 288, 291, 300
- Invoke Method dialog, 317
- Method Call Result dialog, 318–319
- naming classes, 255
- New Project dialog, 101–103, 106, 112, 114–115
- New Test Project dialog, 333
- Options (Tools) dialog, 142–145
- Query Builder dialog, 463–467
- Run Stored Procedure dialog, 461
- settings, exporting, 142–144, 145–147
- support for C# and Visual Basic, 1

Test Results window,  
342–343  
Unit Test Wizard,  
334–335  
Visual Studio, Express  
Edition, 308  
Visual Studio 2003, 105  
Visual Studio Team  
Developer, 331  
Visual Studio Team  
System, 91

**W**

Web-based UIs, 57, 65, 152.  
*See also* Windows-  
based UIs  
Width, Pen objects, 225  
Windows-based UIs, 57,  
65, 152. *See also*  
Web-based UIs  
Windows Application  
projects, 104, 121  
Windows Application  
template, 102–103,  
112  
Windows Authentication,  
442  
Windows XP, themes and  
visual styles, 131  
wrapped controls, 194  
WriteLine method, 292  
WriteOnly keyword, 277

**X – Z**

XML comments, 162  
classes, 256–257  
event handlers, 178  
forms, 162–164  
methods, 287–289  
properties, 275–276  
XML data storage  
mechanisms, 76  
XP (Extreme  
Programming), 24, 41