

Index

SYMBOLS

<c> tag, 352
^ (carat operator), 467
<code> tag, 352
>>, (double greater than sign), unfolding dialog, 105–106
<< (double less-than sign), unfolding dialog, 106
<example> comment tag, 357
<example> tag, 352
<exception> comment tag, 357
<exception> tag, 352–353
<include> tag, 353
<list> tag, 353
<para> tag, 353
<param> comment tag, 357
<paramref> tag, 354
<permission> tag, 354
<remarks> comment tag, 357
<remarks> tag, 354
<returns> comment tag, 357
<returns> tag, 354
<see> tag, 354
<seealso> tag, 354
[] (square brackets), 416
<summary> comment tag, 357
<summary> tag, 354
<typeparam> comment tag, 357
<typeparam> tag, 354
- (unary operator), 467
<value> tag, 354

A

abbreviations
 names, 415–416
 UI design, 122
About dialog box, 496–497, 500
acceptance testing, 59
Access (Microsoft), 139
accidents, happy, 16
activity diagrams, UML, 86–87
adapter class relation patterns, 171
AddConstructorData subroutine, 563–564
add-ins
 code, adding, 227–230
 creating, 222–226
 DelegationAddIn project, 231–236
 described, 222
 improving, 230–231
advise, don't act philosophy, 99–100
Agile and Iterative Development (Larman), 50
agile programming
 Crystal Clear methodology, 51–52
 DBC, 62–67
 described, 49–51
 Microsoft tools, 67–68
 strengths and weaknesses, 61–62
 tools, using all, 405–406
 traditional lifecycles versus, 60–61
XP
 acceptance testing, 59
 coding standard, 54
 collective code ownership, 54
 continuous integration, 54

agile programming (continued)

XP

- customer team member, 59
- design improvement, 54
- drawbacks, 53
- key values, 52
- open workspace, 59
- pair programming, 55–56
- planning game, 53
- simple design, 53
- small releases, 54
- sustainable pace, 59
- system metaphor, 55
- test-driven development, 56–59
- user story, 59
- weaknesses, 59–60
- whole team, 53

agile releases, 480

Aguanno, Kevin (*Managing Agile Projects*), 50

algorithms, classes representing, 199–200

AmbientValue, 332

AmbientValue attribute, 355

AndAlso operator, 7

Andres, Cynthia (*Extreme Programming Explained: Embrace Change*), 52, 59

Anti Patterns, Refactoring Software, Architectures, and Projects in Crisis (Brown), 201

applications

- inside testing, 461–464
- libraries tools for another, 571–575
- metafiles, opening, 531–532
- .NET dependencies, 8
- outside testing, 464–465
- prototyping and simple, 10
- throwaway, 34
- updates, 478–479
- views, 76

arithmetic expressions, 259–261

array, XML, 345

assembly

reflection

- AddConstructorData subroutine, 563–564
- declaration, obtaining, 564–565
- described, 559–560
- method, obtaining information about, 565–566
- path to, 560–563
- representing itself while running, 566–567
- type, checking, 563
- signing, enabling, 142

assertion, 63

AssignJob subroutine, 329–330

attacks, SQL, 240

attributes

- AssignJob subroutine, 329–330
- code editor
 - AttributeUsage, 340
 - ComClassAttribute, 340
 - Conditional, 340–341
 - DebuggerHidden, 342
 - DebuggerStepThrough, 342
 - EditorBrowsable, 342
 - Flags, 342
 - HideModuleName, 343
 - Obsolete, 343
 - ProvideProperty, 343

described, 330–332

form designer

- DefaultEvent, 339
- Docking, 339
- ToolboxBitmap, 340
- ToolboxItem, 340

Properties window

- AmbientValue, 332
- Browsable, 332
- Category, 332–333
- DefaultProperty, 333
- DefaultValue, 333
- Description, 333–334
- Designer, 334
- DisplayName, 334
- Editor, 335
- Localizable, 335
- NotifyParentPropertyAttribute, 335–336
- ParenthesizePropertyName, 336
- PasswordPropertyText, 336
- PropertyTab, 336–338
- ReadOnly, 335
- RefreshPropertiesAttribute, 338–339
- TypeConverter, 339

recommendations, 355–356

serialization

- OnDeserializing, OnDeserialized, OnSerializing, and OnSerialized, 344
- OptionalField, 344–345
- Serializable, 344
- XmlArray, 345
- XmlArrayItem, 345
- XmlAttribute, 345–346
- XmlElement, 346
- XmlAttribute, 346

XmlIgnore, 346–347
 XmlText, 347
 XML comments
 automatic documentation, 348–350
 described, 347–348
 design-time support, 350
 recommendations, 357
 recommended tags, 351–355
 AttributeUsage, **340**

B

background, shaped forms, 502–503
 BackgroundGraphs **program, 543–546**
 BackgroundWorker **component**
 BackgroundGraphs program, 543–546
 described, 542
 MandelbrotBackground program, 548–549
 MandelbrotWorkers program, 551–554
 threading, 542, 545–546
backups, 403–404
Beck, Kent (*Extreme Programming Explained: Embrace Change*), 52, 59
Beginning Visual Basic 2005 Databases (Willis), 134
 BeginPrint **event, 512, 514**
behavior patterns
 Abstract Base Class, 173–176
 Chain of Events, 178–181
 Chain of Responsibility, 176–178
 Command design pattern, 181–183
 delegation, 183–185
 described, 172–173
 MVC, 185–193
black box testing, 460
blank string, converting to
 (PolyPolylineConverter), **309–310**
bleeding edge, avoiding, 404–405
block endings, 429–430
 BoldStyleToggler **class, 572–573**
Boolean names, 415
box, list. See list box
Brown, William (*Anti Patterns, Refactoring Software, Architectures, and Projects in Crisis*), 201
 Browsable **attribute, 332, 355**
bug hunter, 69–70
bug proofing
 catching bugs
 described, 435–436
 global error handling, 438–441
 hunting for, 441–444
 waiting for, 436–438

previous development, tracking, 36, 38
 responding to, 447–451
 Try Catch blocks, 445–447

bugs. See also testing
 garbage collection, 11–12
 handling, 436–438
 threading, 537

builds, frequency of, 396

button
 Add-In, displaying, 227
 Click Me, 147–148
 OK button event handler, 431
 Preview, printing, 513

C

C#
 escaped names, using in library, 416
 IL, 5
 memory allocation bugs, 442
C++, C#
 pointers and, 3
 power and flexibility, 9
 verbosity, 9
 <c> **tag, 352**
carat operator (^), 467
cascade model, 34–36
case view, application, 76
 Category **attribute, 355**
CD-ROM installation with Web updates, ClickOnce
 method, 483–486
central processing unit. See CPU
changing requirements, tracking, 44
classes
 available, displaying list of, 233
 diagrams, UML, 83–85
 escaped names, containing, 416
 events, listing available, 234
 general, looking for, 575–577
 internal, adding in object-oriented design, 74
 invariant, 63
 libraries, 7
 names, 416
 object-oriented design
 candidates, picking, 70–71
 converting to, 71–74
 need for improving, 76
 reflection, loading dynamically, 571–575
 representing strategies or algorithms, 199–200
 structure versus, in Visual Basic, 72
cleverness, avoiding, 396

Click Me button, 147–148

ClickOnce deployment

- CD-ROM installation with Web updates, 483–486
- described, 30, 480–481
- ease of use, 32
- installation, 481–482
- update, 482–483

clone pattern, 164–168

closing forms, 432

Cockburn, Alistair (*Crystal Clear: A Human-Powered Methodology for Small Teams*), 49–50, 51–52

code changes, testing

- described, 29–30, 455
- philosophies, 455
- technique, 459

code editor, 340–343

code, reviewing for bugs. See testing

<code> tag, 352

coding standards

- block endings, 429–430
- Continue, 431–432
- described, 409
- End, 432
- Exit Do, 431–432
- Exit For, 431–432
- exit points, 430–431
- IfThenElseIf statements, 6, 427–429
- names
 - abbreviations, 415–416
 - Boolean, 415
 - class, 416
 - components, 411–413
 - constant, 413
 - enumerated type, defining, 414
 - escaped, 416
 - event, 418–419
 - Hungarian notation, 410–411
 - interface, 414
 - method, 417–418
 - Pascal case, 414
 - property, 417
 - public variables, 414
 - routine variable, 413
- overloading and optional parameters, 422–426
- ParamArray, 426–427
- usefulness, 409–410
- variables, declaring, 419–421
- XP, 54

collection editor, 268

collective code ownership, XP, 54

color

- custom controls, building, 264–265
- drawing program, displaying available, 569–570
- GroupBox caption, 109
- property support, 299
- PropertyGrid control, 108–109
- splash screen, 496
- users setting, 111

columns, database table, 134

ComClassAttribute, 340

ComClassAttribute attribute, 356

command verbs, 300

commands, running, 247–250

comma-separated value (CSV) files, 159–160

comments

- block endings, 429–430
- inline, 371–373
- XML, 400, 423

commercial installation products, 490–491

communication, XP, 52

compiled-in data, DLL, 140, 144–145

CompileExpression_Test program, 465–467

components

- creating, 282–284
- described, 263–264
- diagrams, UML, 89–90
- extender providers and dialogs
 - building, 287–289
 - described, 285–287
 - ExtraTag Provider example, 289–290
 - RegexValidationProvider example, 291–294
- names, 411–413
- prefixes listed, 412–413
- TickOnce example, 284–285

consistency, UI design, 131

constant

- custom to hide test menu, 462
- names, 413

constructors, building derived controls (StyledListBox from ListBox), 268

Contact class, 322–327

context menus, 121

context-sensitive help, user documentation, 366–367

Continue, 431–432

continuous integration, XP, 54

controls, adding to page, 322

Core!DRAW!, 108

cost cutoff, 45–46

courage, XP, 52

CPU (central processing unit)

- multiple threads, running, 536–537, 545
- multiple units, 545
- performance issues with threads, 541
- switching, 535

CrashProof program, 437

creation patterns

- clone pattern, 164–168
- factory class, 168–170

Crystal Clear: A Human-Powered Methodology for Small Teams (Cockburn), 49–50, 51–52

Crystal Clear methodology, 51–52

CSV (comma-separated value) files, 159–160

culture, user interpretation of system, 103

custom classes, disposing, 590–591

custom controls

- derived controls, building (StyledListBox from ListBox)
 - collection editor, 268
 - color, text, font, and image values, 264–265
 - constructors, 268
 - described, 264
 - displaying items with event handler, 268–269
 - drawing items, 269
 - Items property, 267–268
 - serializable, making, 265–267
- deriving from existing control (UserControls),
 - MapViewFinder example
 - corners, defining where shrunken map should be drawn, 278
 - described, 275–276
 - graphical transformation, 277–278
 - MapToolsTest program, 281–282
 - mouse event handlers, 279–281
 - Paint event, drawing with, 278–279
 - described, 263–264
 - testing controls, 271–272
 - toolbox bitmap, setting, 270–271
 - UserControl, building, 272–275

custom tags, XML documentation, 374–375

customer champion, 18

customer team member, XP (Extreme Programming), 59

D

data, compiled-in DLL, 140, 144–145

data storage design

- CSV files, 159–160
- described, 133

INI (.ini) files, 154–156

object databases or object store, 160–161

relational databases

- compiled-in data, 140–145
- information, finding additional, 133–134
- products, 139–140
- relationships among tables, defining, 134–136
- resource files, 145–148
- SQL, 136
- strengths and weaknesses, 136–137
- tables, organization of, 134
- tree and nodes, building, 137–139

satellite assemblies (resource files), 148–149

System Registry (small amounts of data), 149–154

XML, 157–159

data view, application, 76

database

- layers separating, 75
- relational
 - compiled-in data, 140–145
 - information, finding additional, 133–134
 - products, 139–140
 - relationships among tables, defining, 134–136
 - resource files, 145–148
 - SQL, 136
 - strengths and weaknesses, 136–137
 - tables, organization of, 134
 - tree and nodes, building, 137–139
- storage problems, waterfall model and, 36

DBC (design by contract) approach

- agile programming, 62–67
- bugs, searching for, 442

deadline

- cutoff and cost cutoff, 45–46
- documentation and, 362

deadlocks, preventing, 540

debug build, 63–65

DEBUG constant, 462

Debug.Assert statements, 442, 444

DebuggerHidden, 342

DebuggerStepThrough, 342

declaration, 564–565

deep copy, 165–166

DefaultEvent, 339

DefaultEvent attribute, 356

DefaultProperty, 333

DefaultValue, 333

DefaultValue attribute, 355

DelegationAddIn project, 231–236

dependencies, Java, 8

deployment

- ClickOnce method
 - CD-ROM installation with Web updates, 483–486
 - described, 30, 480–481
 - ease of use, 32
 - installation, 481–482
 - update, 482–483
 - Windows Installer, 486–490
- commercial installation products, 490–491
- described, 30–31, 477
- diagrams, UML, 90–91
- models
 - agile releases, 480
 - described, 477
 - major and minor releases, 479–480
 - user types, 478–479
- Xcopy command, Windows, 491–492

Description **attribute**, 355

design

- scripting, 239
- views, studying different, 75–76
- XP, 54

design by contract (DBC) approach

- agile programming, 62–67
- bugs, searching for, 442

design patterns

- behavior patterns
 - Abstract Base Class, 173–176
 - Chain of Events, 178–181
 - Chain of Responsibility, 176–178
 - Command design pattern, 181–183
 - delegation, 183–185
 - described, 172–173
 - MVC, 185–193
- creation patterns
 - clone pattern, 164–168
 - factory class, 168–170
- described, 163–164
- property procedure
 - described, 193–195
 - information, finding more, 201
 - snapshot, 195–199
 - strategy, 199–200
- relation patterns
 - adapter class, 171
 - described, 170
 - facade interface, 171–172
 - interface, 172

Design Patterns (Gamma), 163, 201

Designer, **334**

Designer **attribute**, 356

design-time

- features, implementing, 303
- support, XML comments, 350

developer documentation

- described, 369
- high-level design, 369
- inline comments, 371–373
- module-level documentation, 370–371
- specification, 369
- system-level documentation, 369–370

developers

- layers separating database, 75
- number of Visual Basic, 12

development environment, strong, 52

development philosophy

- agile programming tools, using all, 405–406
- bleeding edge, avoiding, 404–405
- errors, making program find, 394–395
- first, make it work, 395–397
- goals, 391
- good job, do and then move on, 398
- look before you leap, 397–398
- object-oriented principles, 399–400
- program for people, 392–393
- save everything, 403–404
- user, keeping in charge, 393–394
- Visual Studio, taking advantage of, 400–401
- weakest link, beware of, 401–403

device, deployment diagrams, 90

dialog box

- editing (PolyPolylineEditor), 310–313
- folding (<<), 106
- forms, 104
- PolyPolyline, displaying and editing, 310–313
- unfolding (>>), 105–106

disk drives, external, 403–404

displaying items, event handler, 268–269

DisplayName, **334**

DLL (dynamic link library)

- classes for invoking methods dynamically, 575–576
- compiled-in data, 140, 144–145
- discovering resources, 577–580
- encapsulation, 78

Do **loop**, 445–446

Docking, **339**

document view, application, 76

document, XML

- converting to HTML
 - elements to be handled by function, 388
 - HTML table, building, 385–387
 - index page, generating, 383–384
 - links, 381–383
 - Web page, building, 384–385
- converting to text, 376–381

documentation

- described, 361
- developer
 - described, 369
 - high-level design, 369
 - inline comments, 371–373
 - module-level documentation, 370–371
 - specification, 369
 - system-level documentation, 369–370
- safeguarding, 404
- timing
 - frequent changes to, 363–364
 - training, 361–362
- user
 - context-sensitive help, 366–367
 - described, 364
 - help files, 365
 - historical, 369
 - overview, 364
 - specification, 364
 - support materials, 369
 - tooltips, 368
 - training materials, 368–369
 - use cases, 365
 - user manual, 365
- Visual Basic, self, 10
- XML
 - comments, automatic, 348–350
 - custom tags, 374–375
 - program XmlDocToHtml, 381–388
 - program XmlDocToText, 376–381
 - syntax, 373
 - using, 375–376

double greater than sign (>>), 105–106**double less-than sign (<<), 106****downloads**

- user interface, keeping active while, 542
- user types, 478

drawing

- commands, printing procedurally, 528–531
- derived controls, building (StyledListBox from ListBox), 269

Paint event, 278–279

before printing, 515

drawing program

- colors, displaying available, 569–570
- ellipse, 589–590
- hatch brushes, displaying available, 567–569

dynamic link library (DLL)

- classes for invoking methods dynamically, 575–576
- compiled-in data, 140, 144–145
- discovering resources, 577–580
- encapsulation, 78

E**early and often testing, 454–455****ease of use, ClickOnce deployment, 32****easy access to expert users principle, 51****editing dialog box (PolyPolylineEditor), 310–313****editing macros, 214****editor**

- class, 304
- code
 - AttributeUsage, 340
 - ComClassAttribute, 340
 - Conditional, 340–341
 - DebuggerHidden, 342
 - DebuggerStepThrough, 342
 - EditorBrowsable, 342
 - Flags, 342
 - HideModuleName, 343
 - Obsolete, 343
 - ProvideProperty, 343
- form, 107–108

Editor **attribute, 335, 356**

EditorBrowsable **attribute, 342, 356**

elements to be handled by function, XmlDocToHtml program, 388

ellipse drawing program, 589–590

emailing bug information to developers, 448, 450

encapsulation. See also components; custom controls

- described, 399–400
- improving classes, 77–78
- interface, 77

End, **432**

EndPrint **event, 512, 514**

enumerated type, defining, 414

enumerations, 567–571

environment, understanding users, 102

error handling, global, 438–441

ErrorFallback **example program, 448**

errors. *See also* testing

- drawing Mandelbrot sets, 553–554
- flagging with wiggly underline and red rectangle, 99–100
- garbage collection, 11–12
- handling, 436–438
- making program find, 394–395
- threading, 537

escaped names, using in C# library, 416

The Essential Guide to User Interface Design (Galitz), 131

event handler

- bugs, handling, 436–438
- displaying items, 268–269
- GUI testing, 473–474
- memory problems, protecting from, 448

event-driven printing, 512–514

events

- class, listing available, 234
- names, 418–419

everything testing philosophy, 455

evolutionary prototyping, 41–43

<example> **comment tag**, 357

<example> **tag**, 352

exception handling

- described, 66
- needless use of, 447

<exception> **comment tag**, 357

<exception> **tag**, 352–353

execution, multiple threads of. *See* threading

exhaustive testing, 460

Exit Do, 431–432

Exit For, 431–432

exit points, 430

ExitAndEnd **program**, 432

expansion, macro, 7

Expert One-on-One Visual Basic 2005 Database Programming (Jennings), 134

ExpressionLib **example**, 465–467

ExpressionLibTester **class**, 468–469

expressions

- arithmetic, evaluating, 259–261
- variable, 10

extender providers and dialogs

- building, 287–289
- described, 285–287
- ExtraTag Provider **example**, 289–290
- RegexValidationProvider **example**, 291–294

Extensible Markup Language. *See* XML; XML comments; XML documentation

ExtraTag Provider **example**, 289–290

Extreme Programming Explained: Embrace Change (Beck and Andres), 52, 59

Extreme Programming. *See* XP

F

facade interface, 171–172

factory class creation patterns, 168–170

feasibility analysis

- high-level, 21–22
- low-level, 22–23

“feauritis,” 53

feedback

- idea formulation, 16
- splash screens, 500–501
- threading, 547–549
- UI design, providing appropriate, 129–130
- XP, 52

fields, database table, 134

Filev, Andrew (*Professional UML with Visual Studio .NET*), 80

filled text, drawing splash screens, 504–505

finalization, memory management, 587–588

Finalize **methods**, 598

first, make it work development philosophy, 395–397

flagging

- errors, wiggly underline and red rectangle, 99–100
- improper SQL, 400–401

Flags **attribute**, 342, 356

focus principle, 51

font

- custom controls, building, 264–265
- sizes, supporting different in UI design, 115–119

form

- closing, 432
- components, prefixes listed, 412–413
- described, 104
- dialogs, 104
- editors, 107–108
- flow, promoting, 110–113
- MDI (multiple-document interface) versus SDI (single-document interface), 105
- printing, 514–518
- property sheets, 108–109
- resizing, 105–107

form designer

- DefaultEvent, 339
- Docking, 339
- ToolboxBitmap, 340
- ToolboxItem, 340

frequent delivery principle, 51
function, XmlDocumentToHtml program, 388

G

Galitz, Wilbert (*The Essential Guide to User Interface Design*), 131

Gamma, Erich (*Design Patterns*), 163, 201

garbage collection

- bugs, 11–12
- memory management, 585–586, 597–599
- techniques, 6–7

generalize groups, 72

generalizing, 396

global error handling, 438–441

good job, do and then move on development philosophy, 398

Grand, Mark (*Visual Basic Design Patterns*), 201

graphical feedback, threading, 550–551

graphical user interfaces. See GUIs

graphics. See also splash screen

- custom controls, building, 264–265
- printing
 - form, drawing before, 515
 - with metafiles, 531–532
 - text around, 522–527

graphs

- BackgroundGraphs program, 543–546
- values, calculating with threads, 537–540

greater than sign, double (>>), unfolding dialog, 105–106

GroupBox caption, 109

GUIs (graphical user interfaces)

- building, 8–9
- mouse, testing, 471–475
- view, application, 76

H

hackers, 5

handler

- event
 - bugs, handling, 436–438
 - displaying items, 268–269
 - GUI testing, 473–474
 - memory problems, protecting from, 448
- exception
 - described, 66
 - needless use of, 447

hardened prototypes, 43

hatch brushes, displaying available, 567–569

headers, UI design, 122

help files, 365

hidden screens, displaying, 506–508

HideModuleName, 343

high-level design

- described, 23
- developer documentation, 369

high-level feasibility analysis, 21–22

historical user documentation, 369

HTML (HyperText Markup Language), creating from document file

- elements to be handled by function, 388
- HTML table, building, 385–387
- index page, generating, 383–384
- links, 381–383
- Web page, building, 384–385

Hungarian notation, 410–411

I

icon, toolbox, 270–271

IDE (integrated development environment), 11

idea formulation, 16

IDisposable interface, 590–591

If tests, exit points, 430

If-Then style branches, 87

IfThenElseIf statements, 6, 427–429

IIF statement, 6

IL (Intermediate Language) code, compiled executable versus, 4–5

images. See graphics

implementation stage, 25–26

<include> tag, 353

information

- finding additional, 133–134
- hiding, 77
- property procedure, finding more, 201

inheritance

- described, 399
- diagram, 83
- multiple, in Visual Basic, 7
- object-oriented design, 78–79

INI (.ini) files, data storage design, 154–156

inline code, 7

inline comments, 371–373

inserting snippets, 206–208

inside application testing, 461–464

installation

- ClickOnce deployment, 481–482
- snippets, 211

integers, Hungarian notations, 411

integration tests, 458

IntelliSense

- overloading and optional parameters, 422
- taking advantage of, 400
- UI prototype, building, 40

interface. See UI design

Intermediate Language (IL) code, compiled executable versus, 4–5

intermittent crashes, controls, 471

internal state of object, property procedure, 195–199

internationalizing, 396

Invoke method, threading, 542–543

item placement, splash screens, 501–502

iterative prototyping, 41–43

J

Java

- dependencies, 8
- power and flexibility, 9

Jennings, Roger (*Expert One-on-One Visual Basic 2005 Database Programming*), 134

JIT (just-in-time) compiler, 4

K

key values, XP, 52

L

language relatives, 11

language, resource files, 146

language selection, 3–4

Larman, Craig (*Agile and Iterative Development*), 50

layers separating database, 75

libraries

- class, 7
- escaped names, using in C#, 416
- run-time, installing, 5
- tools for another application, building, 571–575

lifeline, sequence diagram, 85

lifestyle methodologies

- deployment, 30–31
- mix and match, 46
- models
 - listed, 33
 - overlapping waterfall, 38
 - serial waterfalls, 36–37
 - throwaway application, 34

waterfall model, 34–36

waterfall with feedback, 38–40

prototyping

deadline cutoff and cost cutoff, 45–46

described, 40–41

iterative prototyping, 41–43

staged delivery, 43–45

throwaway prototyping, 41

stages

high-level design, 23

high-level feasibility analysis, 21–22

idea formulation and refinement, 16–17

implementation, 25–26

lower-level design, 24–25

low-level feasibility analysis, 22–23

requirements gathering, 19–21

support, 31–33

team building, 17–19

testing

described, 26

regression testing, 29–30

system testing, 28–29

unit testing, 26–27

user acceptance testing, 30

line width, property support, 299

LineColor property, 307–308

LineWidth

LineWidthEditor class, 304–306

LineWidthListBox class, 306–307

links, XmlDocumentToHtml program, 381–383

list box

line width, 306–307

resizing, 114

styled, building

collection editor, 268

color, text, font, and image values, 264–265

constructors, 268

described, 264

displaying items with event handler, 268–269

drawing items, 269

Items property, 267–268

<list> tag, 353

Localizable attribute, 335, 356

locks, preventing race conditions with, 539–540

look before you leap development philosophy, 397–398

loops

Continue, 431–432

Do, 445–446

Exit Do, 431–432

Exit For, Exit Do, and Continue, 431–432
 For loop
 Next statement ending, 10
 variables, 420–421

loose coupling, 77**lower-level design, 24–25****low-level feasibility analysis, 22–23****M****macros**

customizing Visual Studio, 218–221
 described, 212
 editing, 214
 expansion, 7
 modifying, 214–217
 recording, 213
 sample, 218
 sharing, 221

main forms, 104**major and minor releases, 479–480****managed heap, 585****management patron, 18–19****Managing Agile Projects (Aguanno), 50**

MandelbrotBackground **program, 547–549**

MandelbrotWorkers **program, 551–554**

ManyReferences **program, 598**

MapToolsTest **program, 281–282**

MapViewFinder **example**

 corners, defining where shrunken map should be drawn, 278
 described, 275–276
 graphical transformation, 277–278
 MapToolsTest program, 281–282
 mouse event handlers, 279–281
 Paint event, drawing with, 278–279

margins, printing

 with metafiles, 531
 without metafiles, 521

max, **415**

MDI (multiple-document interface), 105**memory**

 allocation bugs, 442
 C# allocation bugs, 442
 garbage collection bugs, 11–12
 management
 custom classes, disposing, 590–591
 finalization, 587–588
 garbage collection, 585–586, 597–599
 objects, pre-allocating, 591–595

 resources, disposing, 589–590

 weak references, 595–597

 running out of, 448

 weak references, 595–597

menu

 hiding test, 462

 UI design, 119–121

message boxes, 107**message, displaying bug to user, 443–444****metafiles**

 printing with, 531–532

 printing without, 521

method

 class, listing available, 234

 invariants, 63

 names, 417–418

 obtaining information about, 565–566

 postcondition, 63

 precondition, 63

 reflection, invoking dynamically, 575–577

methodologies, lifestyle

 deployment, 30–31

 mix and match, 46

 models

 listed, 33

 overlapping waterfall, 38

 serial waterfalls, 36–37

 throwaway application, 34

 waterfall model, 34–36

 waterfall with feedback, 38–40

 prototyping

 deadline cutoff and cost cutoff, 45–46

 described, 40–41

 iterative prototyping, 41–43

 staged delivery, 43–45

 throwaway prototyping, 41

 stages

 high-level design, 23

 high-level feasibility analysis, 21–22

 idea formulation and refinement, 16–17

 implementation, 25–26

 lower-level design, 24–25

 low-level feasibility analysis, 22–23

 requirements gathering, 19–21

 support, 31–33

 team building, 17–19

 testing

 described, 26

 regression testing, 29–30

 system testing, 28–29

 unit testing, 26–27

 user acceptance testing, 30

Meyer, Bertrand (*Object-Oriented Software Construction*), 63

Microsoft Access, 139

Microsoft agile programming tools, 67–68

Microsoft Paint, 532

min, **415**

models, lifestyle methodologies

listed, 33

overlapping waterfall, 38

serial waterfalls, 36–37

throwaway application, 34

waterfall model, 34–36

waterfall with feedback, 38–40

module-level documentation, 370–371

Mono project, 8, 31

mouse

event handlers, deriving from existing control
(UserControls), MapViewFinder example,
279–281

GUI testing, 471–475

selection, easing, 111

moving or blinking objects, 130

MSF for Agile Software Development, 67–68

multiple CPUs, 545

multiple inheritance, 7

multiple threads, running, 536–537, 545

multiple-document interface (MDI), 105

N

names

abbreviations, 415–416

class, 416

coding standards

constant, 413

enumerated type, defining, 414

interface, 414

Pascal case, 414

public variables, 414

routine variable, 413

components, 411–413

event, 418–419

Hungarian notation, 410–411

method, 417–418

property, 417

system metaphor, 55

variables, 392–393

navigator, pair programming, 55

.NET (Visual Basic)

application dependencies, 8

memory management

custom classes, disposing, 590–591

finalization, 587–588

garbage collection, 585–586, 597–599

objects, pre-allocating, 591–595

resources, disposing, 589–590

weak references, 595–597

reflection as part of, 559

upgrading from Visual Basic 6 to, 8

Next **statement, 10**

nodes, deployment diagrams, 90

non-deterministic finalization, 6, 589

NUnit testing tool, 467–470

O

obfuscators, 5

object

databases or object store data storage design,
160–161

memory management, pre-allocating, 591–595

model, exposing while scripting Visual Basic, 254–257

object, characteristic of. See property

object view, application, 76

object-oriented design

bug hunter example, 69–70

candidate classes, picking, 70–71

candidates, converting to classes, 71–74

database classes, adding, 75

design views, studying different, 75–76

first steps, 70

improving classes

encapsulation, 77–78

need for, 76

polymorphism and inheritance, 78–79

internal classes, adding, 74

principles, 399–400

UML

activity diagrams, 86–87

class diagrams, 83–85

component diagrams, 89–90

deployment diagrams, 90–91

described, 79–80, 92

sequence diagrams, 85–86, 91

state chart diagrams, 87–89

tools, 92–95

use case diagrams, 80–82, 91

Object-Oriented Software Construction (Meyer), 63
Objects by Design Web site, 92
observer, pair programming, 55
Obsolete attribute, 343, 356
ObsoleteAttribute class, marking method as no longer supported, 40
OnDeserializing, OnDeserialized, OnSerializing, and OnSerialized attribute, 344
OnEvent methods, 472
open workspace, XP, 59
operating system versions, large installation problem caused by, 31
Option Explicit, 6
Option Strict, 6
OptionalField attribute, 344–345, 356
order, preventing deadlocks, 540
OrElse operator, 7
osmotic communication principle, 51
outside application testing, 464–465
overlapping waterfall, 38
overloading and optional parameters, IntelliSense, 422

P

Paint event, drawing with, 278–279
Paint (Microsoft), 532
pair programming, XP (Extreme Programming), 55–56
<para> tag, 353
paragraphs, printing
 around graphics, 523–527
 without graphics, 518–522
<param> comment tag, 357
ParamArray, 426–427
parameters, problem with overloaded methods, 424–425
<paramref> tag, 354
ParenthesizePropertyName, 336
Pascal case, 414
PasswordPropertyText, 336
PasswordPropertyText attribute, 356
path
 to assembly, 560
 Bug Hunter program, 73
patterns, relation
 adapter class, 171
 described, 170
 facade interface, 171–172
 interface, 172
performance
 requirements, 20
 threads, 541
<permission> tag, 354
personal safety principle, 51
pictures. See graphics
planning game, XP, 53
platform dependencies, Visual Basic, 8
pointers
 C++, C#, 3
 VB, awkwardness of using, 6
political infighting and enemies, 19
polylines, 298–300
polymorphism
 described, 399
 object-oriented design, 78–79
PolyPolyline, displaying and editing
 blank string, converting to (PolyPolylineConverter), 309–310
 described, 308–309
 editing dialog box (PolyPolylineEditor), 310–313
prefixes, components listed with, 412–413
Preview button, printing, 513
previous development, tracking, 36, 38
PrintDocument event handlers, 516–517
printing
 described, 511
 event-driven, 512–514
 flowing text around pictures, 522–527
 forms, 514–518
 paragraphs without graphics, 518–522
 procedural method, 528–532
 wrapping text across pages, 518–522
PrintPage event, 512, 514
priority, thread's, 537
procedural method, printing, 528–532
processor, computer
 multiple threads, running, 536–537, 545
 multiple units, 545
 performance issues with threads, 541
 switching, 535
products, relational database, 139–140
Professional UML with Visual Studio .NET (Filev), 80
Properties window
 AmbientValue, 332
 Browsable, 332
 Category, 332–333
 customizations
 command verbs, 300
 design-time features, implementing, 303
 polylines, 298–300
 property sheets, 302–303
 smart tags, 301–302

Properties window (continued)

- DefaultProperty, 333
- DefaultValue, 333
- described, 297–298
- Description, 333–334
- Designer, 334
- DisplayName, 334
- Editor, 335
- LineColor property, 307–308
- LineWidth
 - LineWidthEditor class, 304–306
 - LineWidthListBox class, 306–307
- Localizable, 335
- NotifyParentPropertyAttribute, 335–336
- object properties, displaying
 - Contact class, 322–327
 - UserControl string representation, 327–328
- ParenthesizePropertyName, 336
- PasswordPropertyText, 336
- PolyPolyline, displaying and editing
 - blank string, converting to (PolyPolylineConverter), 309–310
 - described, 308–309
 - editing dialog box (PolyPolylineEditor), 310–313
- property sheets, displaying
 - controls, adding to page, 322
 - ScribbleControl class, 319
 - ScribbleControlEditor class, 319–320
 - ScribbleControlEditorBasicsPage class, 320–321
- PropertyTab, 336–338
- ReadOnly, 335
- RefreshPropertiesAttribute, 338–339
- smart tags, displaying, 313–319
- TypeConverter, 339
- property**
 - class, listing available, 234
 - color, 299
 - displaying
 - Contact class, 322–327
 - UserControl string representation, 327–328
 - editor class, 304
 - names, 417
 - procedure
 - described, 193–195
 - information, finding more, 201
 - snapshot, 195–199
 - strategy, 199–200
 - sheets
 - controls, adding to page, 322
 - customizations, 302–303
 - PropertyGrid control, 108–109
 - ScribbleControl class, 319
 - ScribbleControlEditor class, 319–320
 - ScribbleControlEditorBasicsPage class, 320–321
 - PropertyGrid **control**, 108–109
 - PropertyTab, **336–338**
 - prototype**
 - deadline cutoff and cost cutoff, 45–46
 - described, 40–41
 - iterative prototyping, 41–43
 - staged delivery, 43–45
 - in stages, UI design, 110
 - throwaway prototyping, 41
 - UI, building, 10, 40
 - prototype pattern**, 164–168
 - ProvideProperty, **343**
 - ProvideProperty **attribute**, 356
 - public variables, naming**, 414

Q

queries

- executing, 241–244
- generating, 244–247
- scripting, 238–239
- QueryPageSettings **event**, 512

R

race conditions, 537–540

- ReadOnly, **335**
- ReadOnly **attribute**, 356
- recording macros**, 213
- records, database table**, 134
- reflection**

assemblies

- AddConstructorData subroutine, 563–564
- declaration, obtaining, 564–565
- described, 559–560
- method, obtaining information about, 565–566
- path to, 560–563
- representing itself while running, 566–567
- type, checking, 563

- classes, loading dynamically, 571–575
- enumerations, 567–571
- methods, invoking dynamically, 575–577

resources, discovering, 577–580
 retrieving known resources, 580–583
reflective improvement principle, 51
“Reflector for .NET” program (Roeder), 4–5
 RefreshProperties attribute, 356
 RegexValidationProvider example, 291–294
registration, product, 478
regression testing
 described, 29–30, 455
 philosophies, 455
 technique, 459
related items, grouping in UI design, 113–114
relation patterns
 adapter class, 171
 described, 170
 facade interface, 171–172
 interface, 172
relational databases
 compiled-in data, 140–145
 information, finding additional, 133–134
 products, 139–140
 relationships among tables, defining, 134–136
 resource files, 145–148
 SQL, 136
 strengths and weaknesses, 136–137
 tables, organization of, 134
 tree and nodes, building, 137–139
relationship diagram, 84
 ReleaseAssert subroutine, 442–443
reliability, protecting appearance of, 32
 <remarks> comment tag, 357
 <remarks> tag, 354
requirements gathering stage, 19–21
resistance to projects, users, 17
resizing
 forms, 105–107
 list boxes, 114
 UI design, supporting, 114–115
 window, crashing, 471–472
resource
 files, relational databases, 145–148
 memory management, disposing, 589–590
respect, XP, 52
responding to bugs testing philosophy, 456
 <returns> comment tag, 357
 <returns> tag, 354
right thing, help users do, 100–101
Roeder, Lutz (“Reflector for .NET” program), 4–5

Roques, Pascal (UML in Practice: The Art of Modeling Software Systems Demonstrated through Worked Examples and Solutions), 80
 rotating text, splash screen, 505–506
routine variable names, 413
routines
 exit points, 430–431
 unit testing, 26–27
rows, database table, 134
run-time libraries, installing, 5

S

safeguarding documentation, 404
satellite assemblies (resource files), 148–149
save everything development philosophy, 403–404
save forever testing philosophy, 455
 ScribbleControl class, 319
 ScribbleControlEditor class, 319–320
 ScribbleControlEditorBasicsPage class, 320–321
scripting
 cautions about use of, 237–238, 239–241
 design consequences, 239
 expressions, evaluating, 259–261
 other queries, 238–239
 SQL
 attacks, 240
 executing queries, 241–244
 generating queries, 244–247
 knowledge of, 238
 running commands, 247–250
 Visual Basic code
 described, 251
 object model, exposing, 254–257
 running, 251–254
 simplifying, 257–259
scrolling areas, resizing, 114
SDI (single-document interface), 105
secondary forms, 104
security, IL, 5
 <see> tag, 354
 <seealso> tag, 354
self-documenting Visual Basic, 10
sequence diagrams, UML, 85–86, 91
serial number, including in splash screen, 497
serial waterfalls, 36–37
 Serializable attribute, 344, 356

StyledListBox derived control

- collection editor, 268
- constructors, 268
- custom controls
 - color, text, font, and image values, 264–265
 - described, 264
- displaying items with event handler, 268–269
- drawing items, 269
- Items property, 267–268

subroutines, unit testing, 26–27**subsystems, 89–90****sub-unit tests (bugs in smallest units of executable code), 456–457****successive refinement, 41–43****<summary> comment tag, 357****<summary> tag, 354****support, lifestyle methodologies, 31–33****support materials, user documentation, 369****sustainable pace, XP, 59****swimlane version, 87**SyncLock **statement, 554**SyncLockTest **program, 554–557****syntax errors, highlighting, 400****system metaphor, XP, 55****System Registry**

- data storage design, 149–154
- INI files versus, 155, 156

system state snapshot, 123–127**system testing, 28–29****system tests, 458–459****system-level documentation, 369–370****systems, 89–90****T**TabControl **control, 121–122****table**

- relational databases, organization of, 134
- relationships among, defining, 134–136
- SQL Server size, 139–140

tags

- smart
 - displaying, 313–319
 - Properties window, customizing, 301–302
- XML comments, 351–355

talent pool, Visual Basic, 12**tasks**

- dividing, 551–554
- UI design, 98–99

team building, 17–19**test-driven development, XP (Extreme Programming), 56–59****testing**

- custom controls, 271–272
- described, 26
- integration tests, 458
- mechanics
 - ExpressionLib example, 465–467
 - inside application, 461–464
 - outside application, 464–465
- need for, 398
- philosophy
 - described, 453–454
 - early and often, 454–455
 - everything, 455
 - responding to bugs, 456
 - save forever, 455
- regression testing, 29–30, 459
- Staged Delivery, 45
- sub-unit tests (bugs in smallest units of executable code), 456–457
- system testing, 28–29, 458–459
- techniques
 - black box testing, 460
 - described, 459
 - exhaustive testing, 460
 - random testing, 461
 - white box testing, 461
- tools
 - GUI, 471–475
 - NUnit, 467–470
 - Visual Studio, 470–471
- unit testing, 26–27, 457–458
- user acceptance testing, 30

text

- boxes, restricting, 100
- custom controls, building, 264–265
- around graphics, 522–527

threading

- advantages, 536–537
- BackgroundWorker component, 542, 545–546
- described, 535
- disadvantages
 - confusion, 537
 - deadlocks, 540–541
 - performance issues, 541
 - race conditions, 537–540
- dividing tasks, 551–554

threading (continued)

- feedback, 547–549
- graphical feedback, 550–551
- SyncLock statement, 554–557
- UI, 542–546
- throwaway application, 34**
- throwaway prototyping, 41**
- TickOnce **component example, 284–285**
- time, splash screen display, 498–499**
- time-outs, 541**
- toolbox bitmap**
 - custom controls, setting, 270–271
 - ToolboxBitmap attribute, 340, 356
- ToolboxItem **attribute, 340, 356**
- tools**
 - agile programming, 67–68, 405–406
 - libraries for another application, 571–575
 - testing
 - GUI, 471–475
 - NUnit, 467–470
 - Visual Studio, 470–471
 - UML, 92–95
- tooltips, user documentation, 368**
- tracking, ease of, 44**
- traditional lifecycles, agile programming versus, 60–61**
- training**
 - documentation, 361–362
 - materials, 368–369
- tree and nodes, relational databases, 137–139**
- TrueType fonts, 113**
- TryCatch **block**
 - bug catching, 436–438, 440
 - bug proofing, 445–447
 - memory problems, protecting event handler from, 448
- type checking, strict, 6**
- TypeConverter **attribute, 339, 356**
- typeface**
 - custom controls, building, 264–265
 - sizes, supporting different in UI design, 115–119
- <typeparam> **comment tag, 357**
- <typeparam> **tag, 354**

U

UI (user-interface) design

- consistency, importance of, 131
- described, 97
- design principles
 - described, 109
 - feedback, providing appropriate, 129–130

- font sizes, supporting different, 115–119
 - form flow, promoting, 110–113
 - headers, designing good, 122
 - menus, optimizing, 119–121
 - prototype in stages, 110
 - related items, grouping, 113–114
 - resizing, supporting, 114–115
 - skill levels, accommodating different, 127–129
 - space, using wisely, 121–122
 - state, preserving, 122
 - undo and redo, 123–127
 - downloading, keeping active while, 542
 - encapsulation, 77
 - forms
 - described, 104
 - editors, 107–108
 - MDI versus SDI, 105
 - property sheets, 108–109
 - resizing, 105–107
 - names, 414
 - philosophy
 - advise, don't act, 99–100
 - right thing, help users do, 100–101
 - tasks, focus on, 98–99
 - user control, 98
 - prototypes, building, 10, 40
 - relation patterns, 172
 - threading, 536, 542–546
 - users
 - environment, understanding, 102
 - job, understanding, 101
 - respecting, 101–102
 - types, understanding, 102–103
- UML in Practice: The Art of Modeling Software Systems Demonstrated through Worked Examples and Solutions (Roques), 80**
- UML (Universal Modeling Language)**
- activity diagrams, 86–87
 - class diagrams, 83–85
 - component diagrams, 89–90
 - deployment diagrams, 90–91
 - described, 76, 79–80, 92
 - sequence diagrams, 85–86, 91
 - state chart diagrams, 87–89
 - tools, 92–95
 - use case diagrams, 80–82, 91
- unary operator (-), 467**
- undo and redo, user-interface (UI) design principles, 123–127**
- unfolding dialog, 105**

`UnhandledException` **event handler**, 439, 448–450

unit testing

relatively self-contained pieces of code, 457–458
routines, 26–27

Universal Modeling Language. *See* UML

unnamed objects, 86

unusual situations, leaving the user to run, 393–394

update

application user types, 478–479
ClickOnce, 482–483

up-front design, 397–398

upgrading from Visual Basic 6 to Visual Basic .NET, 8

use case

diagrams, UML, 80–82, 91
performance requirements, 20
user documentation, 365

user

acceptance testing, 30
control philosophy, UI design, 98, 393–394
environment, understanding, 102
job, understanding, 101
manual, 365
number of Visual Basic, 12
resistance to projects, 17
respecting, 101–102
story, XP, 59
training, Staged Delivery, 45
types
deployment, 478–479
downloads, 478
understanding, 102–103

user documentation

Bug Hunter program, 73
context-sensitive help, 366–367
described, 364
help files, 365
historical, 369
overview, 364
specification, 364
support materials, 369
tooltips, 368
training materials, 368–369
use cases, 365
user manual, 365

`UserControl`

building, 272–275

`MapViewFinder` example

corners, defining where shrunken map should be drawn, 278
described, 275–276
graphical transformation, 277–278

`MapToolsTest` program, 281–282

mouse event handlers, 279–281

`Paint` event, drawing with, 278–279

string representation, 327–328

user-interface design. *See* UI design

`Using` **keyword**, 589–590

`Using` **statement**, 421

`UsingPen` **program**, 589–590

V

`<value>` **tag**, 354

variables

coding standards, declaring, 419–421
deadlocks, 540–541
expressions, 10
locking to prevent race conditions, 539–540
names, 392–393

VB (Visual Basic)

“Classic” version, 11
garbage collection, 11–12
GUIs, building, 8–9
IDE, 11
IL code, compiled executable versus, 4–5
language relatives, 11
multiple inheritance, 7
platform dependencies, 8
pointers, awkwardness of using, 6
power and flexibility, 9
prototyping and simple applications, 10
scripting
described, 251
object model, exposing, 254–257
running, 251–254
VBScript, 11
self-documenting, 10
syntactic annoyances, 6–7
talent pool, 12
upgrading from Visual Basic 6 to Visual Basic .NET, 8
verbosity, 9

VB (Visual Basic) .NET

application dependencies, 8
memory management
custom classes, disposing, 590–591
finalization, 587–588
garbage collection, 585–586, 597–599
objects, pre-allocating, 591–595
resources, disposing, 589–590
weak references, 595–597
reflection as part of, 559
upgrading from Visual Basic 6 to, 8

VBA (Visual Basic for Applications), 11

VBScript, 11

verbosity

C++, C#, 9

VB, 13

versions, operating system, 31

Visual Basic. See VB; VB .NET

Visual Basic Database Programming (Stephens), 134

Visual Basic Design Patterns (Grand), 201

Visual Basic Graphics Programming: Hands-On Applications and Advanced Color Development (Stephens), 496

Visual Basic .NET and XML (Stephens), 157

Visual Source Safe (VSS) revision control system, 362–363

Visual Studio (Microsoft)

taking advantage of, 400–401

testing tools, 470–471

VSS (Visual Source Safe) revision control system, 362–363

W

waterfall model, 34–36

waterfall with feedback, 38–40

weak references, memory management, 595–597

weakest link, beware of, 401–403

Web page, building (XmlDocToHtml program), 384–385

Web site, reporting bugs online, 451

white box testing, 461

whole team, XP (Extreme Programming), 53

width, line

LineWidthEditor class, 304–306

LineWidthListBox class, 306–307

Willis, Thearon (Beginning Visual Basic 2005 Databases), 134

window, crashing resizable, 471–472

Windows Installer, ClickOnce method, 486–490

wrapping text across pages, printing, 518–522

X

Xcopy command, Windows, 491–492

XML (Extensible Markup Language). See also snippets

comments, 400, 423

data storage design, 157–159

XmlAttribute attribute, 345

XmlAttributeItem attribute, 345

XmlAttribute, 345–346

XmlDocToHtml program

elements to be handled by function, 388

HTML table, building, 385–387

index page, generating, 383–384

links, 381–383

Web page, building, 384–385

XmlDocToText program, 376–381

XmlElement, 346

XmlAttribute, 346

XmlAttribute, 346–347

XmlAttribute, 347

XML (Extensible Markup Language) comments

AssignJob subroutine, 329–330

attributes

automatic documentation, 348–350

described, 347–348

design-time support, 350

recommendations, 357

recommended tags, 351–355

XML (Extensible Markup Language) documentation

custom tags, 374–375

program XmlDocToHtml

elements to be handled by function, 388

HTML table, building, 385–387

index page, generating, 383–384

links, 381–383

Web page, building, 384–385

XmlDocToHtml **program**

elements to be handled by function, 388

HTML table, building, 385–387

index page, generating, 383–384

links, 381–383

Web page, building, 384–385

XmlDocToText **program, 376–381**

XP (Extreme Programming)

acceptance testing, 59

coding standard, 54

collective code ownership, 54

continuous integration, 54

customer team member, 59

design improvement, 54

drawbacks, 53

key values, 52

open workspace, 59

pair programming, 55–56

planning game, 53

simple design, 53

small releases, 54

sustainable pace, 59

system metaphor, 55

test-driven development, 56–59

user story, 59

weaknesses, 59–60

whole team, 53