

Index

A

- Abstract, 430**
- abstraction, 18**
 - C#, 446–448
 - Factory, 139
 - MVC, 323
 - MVP, 330
 - ORMs, 253
 - ReturnOrderService, 64
 - TDD, 111
 - TouristInformationService, 139
- AcceptVerbs, 378**
- AccountController, 374**
- Action, 324**
- Action Pack, 332**
- ActionForward, 324**
- ActionLink, 397–398, 407–408**
 - LoanApplication, 410, 413
- actions**
 - Controller (MVC), 370, 421
 - Java Struts, 324
 - parameters, 370–371
- ActionServlet, 324**
- Active Record pattern, 188–190, 419**
 - DAO, 254
 - vs. Domain Model pattern, 191
 - LinqToSQL, 284
 - Order, 189–190
 - uses for, 194
- Active Server Pages (ASP), 169, 369**
 - COM, 169, 325–326
- Active Template Library (ATL), 169**
- ActiveX, 169**
 - code, 417
 - DCOM, 183
 - VB, 183, 325
- Adabas, 166**
- Add Reference, 76**
- AddEmployee(), 446**
- AddItemToCart, 419**
- AddModelError, 395**
- AddRef, 10**
- Address, 206**
 - Borrower, 199, 216–219
 - interfaces, 211
- AddRuleViolations, 438**
- ADO.NET, 18–19**
- ADO.NET, 12**
 - DAL, 252
 - EF, 270
 - System.Data, 451–452
- aggregate roots**
 - entities, 199–200
 - LoanApplication, 244–245
- aggregates**
 - DDD, 199–200, 203–204
 - NHibernate, 312–313
 - Repository, 243–244
- Agile, 5**
- AJAX, 373**
 - System.Web.Mvc.Ajax, 451
- AlwaysValidHolidayRequestValidator, 121**
 - CanGrant, 122
- Amount, 57**
- anonymous delegates, 437**
- Apache, 168**
 - modules, 169
- App.Config, 314–315**
- App_Data, 256, 270, 272**
 - NHibernate, 290, 302
- APP_Data, 373**
- AppendixC_proj, 435–436**
- Apple. See Mac OS X**
- Application, 275**
- Application folder, NHibernate, 288**
- application servers**
 - Internet, 168
 - middleware, 169
- ApplicationDAO, 260, 276**
- ApplicationDetail.aspx, 297–298**
- ApplicationDetails, 275**
 - EF, 270

ApplicationDetails, 273, 275

ApplicationDetails.aspx

 LinqToSQL, 264–265

 Web Forms, 264–265, 280–281

Application.hbm.xml, 291

ApplicationId, 257, 270

ApplicationRepository, 294–295

Applications, 273

 Application, 275

Applications and Bugs, 272

appSetting, 384

AreEqual, 78

.asmx, ASP.NET, 171–172

ASP. See Active Server Pages

ASP.NET, 8, 325–328

 .asmx, 171–172

 Click, 326

 code, 417

 Code-Behind Model, 343

 enterprise, 328

 HTML, 326

 IDE, 326

 life cycle, 326–327, 378

 MVP, 329, 331, 343

 .NET Framework, 325

 ServerClick, 350

 StructureMap, 159

 UI, 368

 Web Forms, 343

 web.config, 173

 Xaml, 364

ASP.NET MVC, 325, 326, 328, 334–335

 DDD, 393

 EF, 382

 .NET Framework, 334

 REST, 378

 routing, 370–371

 SP1, 371

 Visual Web Developer, 371

 VS.NET, 371, 373

 XML, 382

.aspx, 378

assemblies, 434

Assert, 78

Asset, 219–220

Asset.hbm.xml, 306

Assets, 198, 387

ATL. See Active Template Library

attributes

 HTML, 368

 HTTP, 368

 LinqToSQL, 285

 NHibernate, 285

 StructureMap, 153–157

authentication, 169

 LDAP, 170

authorization, 170

automation, unit testing, 113, 114–116

AverageCreditScore, 393

ayende.com, 123

B

BankAccount, 206

 Borrower, 216–219

 interfaces, 211

BasePrice, 21

BaseRepository, 312–313

Biztalk, 183

bool, 428

Borrower, 198

 Address, 199, 216–219

 BankAccount, 216–219

 Employer, 216–219

Borrower Aggregation, 204

BorrowerController, 370, 402

 Create, 406

BorrowerRepository, 243

Borrowers, 387

Borrowers folder

 credit score, 215–216

 Employer, 213–215

BorrowerService, 247–248

Bottom Up, 254

boundary conditions

 TDD, 111

 triangulation, 115

BrokenBusinessRules, 206

 ModelStateDictionary, 395

btnGetDate, 330

Bug, 275

Bug folder, NHibernate, 288

BugDAO, Model folder, 261–262, 276–277

Bug.hbm.xml, 292–293

BugId, 257

Bugs, 275

BugTrackerDataContext, 260, 261

BugTrackerDB

 LinqToSQL, 256–261

 Server Explorer, 256

BugTracker.dbml, 257, 260
 XML, 259
BugTracker.Designer.cs
 DataContext, 276
 LinqToSQL, 276
BugTracker.designer.cs, 260
BugTracker.edmx, 270, 275, 276
BuildReturnMessageFrom, 52
business logic layer, 185–186
 DAL, 251
business object-focused query
 language, 253
Button.Click, 433
byte, 428

C

C, 169

C#, 427–428
 abstraction, 446–448
 classes, 440–442
 declaration, 429–430
 delegates, 431
 encapsulation, 442, 443–446
 entities, 428
 events, 431
 extension methods, 437–438
 inheritance, 442–443
 lambdas, 437
 .NET, 425–453
 .NET SDK, 448–452
 objects, 440–442
 OO, 440
 polymorphism, 446–448
 properties, 430
 scope, 429–430
 types, 428
 XML, 384

C++, 9, 169
 ISAPI, 326
 MFC, 324

caching
 ORMs, 253
 System.Web, 449
 ViewState, 326

calculatePaymentButton, 349

Call Web Method, 182

Can_Add_Product_To_Repository(),
 NHibernate, 317

Cancel, Order, 190

CanGrant, AlwaysValidHoliday
 RequestValidator, 122
CanGrant(), HolidayRequest, 124
CanPlaceMarketAt, 84
 TicTacToeGame, 85–86
CanProcess, 193
Capability Maturity Model, 5
Castle Project, 160, 333
CGI. See **Common Gateway Interface**
ChangeCurrentPlayer, 91
ChangeCustomerAddress, 419
ChangePassword, 450
char, 428
CheckChanged, 326
CheckOrderStatus, 419
Class Library, 76
Class1.cs, 34
classes
 BorrowerService, 247–248
 C#, 440–442
 declaration, 345
 DI, 64–72
 emancipating, 33–72
 LoanApplication, 224–228
 LoanApplicationService, 246–247
 ProductIsAvailableSpecification, 235
 ProductService, 246–247
 user-defined, 429
CLI. See **Common Language Infrastructure**
Click
 ASP.NET, 326
 MVP, 330
client-server architecture, 166–167
code, 417–418
 ActiveX, 417
 ASP.NET, 417
 comments, 60
 dependencies, 33–46
 DI, 418
 enterprise, 15–30
 Factory, 140
 IoC, 418
 loose coupling, 17–23
 modules, 16–17
 refactoring, 47–48
 TDD, 417–418
 TouristInformationService, 137
 trivial, 111
Code-Behind Model, 343
Codeplex.com, 13

- Cold Fusion, 369**
- collection initializers, 438–440**
- Column, 84**
- column, 92**
- COM. See Component Object Model**
- command-line utility, 168**
 - WCF, 172
- comments, code, 60**
- Common Gateway Interface (CGI), 168–169**
- Common Language Infrastructure (CLI), 11**
- Common Object Request Broker Architecture (CORBA), 171**
- ComplexClass, 26**
- ComplexObject, 27–28**
- Component Object Model (COM), 9, 10**
 - ASP, 169, 325–326
 - HTML, 326
 - .NET, 427
 - .NET SDK, 427
 - VB, 325
- conceptual layer, EF, 284**
- Conceptual Schema Definition Language (CSDL), 275**
- configSections, 159**
- Constructor Injection, 64**
 - ReturnOrderService, 65–66, 68–69
 - TouristInformationService, 135
- Control, 449**
- Controller (MVC), 370**
 - actions, 370, 421
 - Create, 378
 - Index, 378
 - ModelStateDictionary, 394
 - mortgage loan application, 376–378
- ControllerHelper, 396**
- Convention over Configuration, 332**
- cookies, 326**
- CORBA. See Common Object Request Broker Architecture**
- create, read, update, and delete (CRUD), 190**
 - DAL, 251
 - MVC, 376
- Create, 370**
 - BorrowerController, 406
 - Controller (MVC), 378
 - LoanApplication, 394, 396, 407
- Create.aspx, 370, 378**
 - LoanApplication, 394, 396
- CreateUserWizard, 450**

- credit score, 215–216**
- CreditScores, 387**
- CRUD. See create, read, update, and delete**
- CSDL. See Conceptual Schema Definition Language**
- custom exceptions, 224**
- Customer, 35**
- CustomerProvider, 419**

D

- DAL. See data access layer**
- DAO. See Data Access Object**
- data access layer (DAL), 251–253, 420**
 - ADO.NET, 252
 - automation, 285
 - business logic layer, 251
 - CRUD, 251
 - EF, 269–283
 - interfaces, 251
 - LinqToSQL, 254–268
 - ORMs, 252–253
 - persistence, 420
- Data Access Object (DAO), 254**
 - LinqToSQL, 255–261
 - Repository pattern, 284
- Data First, 254**
 - Model First, 284
- Data Transfer Object (DTO), 184, 422**
- database servers, 167**
 - Internet, 168
- DataColumn, 452**
- DataColumnCollection, 452**
- DataContext, 253–268**
 - Identity Map, 253
 - LinqToSQL, 254–255, 285
 - NHibernate, 285
- DataContext, 268**
 - BugTracker.Designer.cs, 276
- [DataContract], 172**
- [DataMember], 172**
- DataRowCollection, 23**
- DataSet, 451**
- DataTable, 451–452**
- DataToTextFormatter, 24**
- DateTime, 442**
 - .NET, 330
- DCOM. See Distributed Component Object Model**
- DDD. See Domain-Driven Design**
- Debt, 220–222**

- Debt.hbm.file, 307**
 - Debts, 198, 387**
 - debugging**
 - default.aspx, 268
 - EF, 283
 - LinqToSQL, 268
 - NHibernate, 299
 - unit testing, 113
 - decimal, 428**
 - declaration**
 - C#, 429–430
 - classes, 345
 - interfaces, 63, 172
 - WCF, 172
 - Default.aspx**
 - HTML, 278–279
 - LinqToSQL, 278–279
 - default.aspx**
 - debugging, 268
 - LinqToSQL, 262–264
 - NHibernate, 295–296
 - Visual Studio, 295–296
 - delegates**
 - anonymous, 437
 - C#, 431
 - DeleteCustomer, 419**
 - dependencies**
 - code, 33–46
 - creating, 131–138
 - Factory, 139
 - INotificationService, 66
 - MOM, 182
 - ReturnOrderService, 44, 60, 64
 - Service Locator pattern, 142
 - stub objects, 118
 - TDD, 118–127
 - TouristInformationService, 134, 137, 139
 - unit testing, 113, 316
 - Dependency Injection (DI), 25**
 - classes, 64–72
 - code, 418
 - INotificationService, 65–66
 - IoC, 64
 - vs. IoC, 146–149
 - LoanApplication, 246
 - loose coupling, 28, 135
 - modules, 64
 - runtime, 64
 - StructureMap, 146
 - Dependency Inversion Principle (DIP), 61–64**
 - ProductService, 61–62
 - DespatchAddress, 193**
 - Details, 370**
 - Details.aspx, 370**
 - Developer folder, 288**
 - DeveloperFirstName, 290**
 - DeveloperSecondName, 290**
 - DHTML, 370**
 - DI. See Dependency Injection**
 - Dictionary, 21**
 - DIP. See Dependency Inversion Principle**
 - DiscountList, 21–22**
 - DiscountOffer, 193**
 - Distributed Component Object Model (DCOM), 171**
 - ActiveX, 183
 - <div>, 329**
 - .dll, 434**
 - Document Type Definition (DTD), 170**
 - Document/View, 324**
 - Domain Model pattern, 190–194, 419–420**
 - vs. Active Record pattern, 191
 - language of, 197–198
 - Process, 193
 - Repository, 194
 - vs. Transaction Script, 194
 - uses for, 194
 - Domain-Driven Design (DDD), 198–249**
 - aggregates, 199–200, 203–204
 - ASP.NET MVC, 393
 - entities, 198–199
 - Model First, 284
 - persistence, 420
 - Repository, 242–244
 - value objects, 199
 - double, 428**
 - double-precision floats, 170**
 - DTD. See Document Type Definition**
 - DTO. See Data Transfer Object**
- ## E
- Edit, 370**
 - GetWorkingApplication(), 410
 - IStateManager, 410
 - LoanApplication, 408, 412
 - LoanApplication.Update(), 400
 - Edit.aspx, 370, 378**
 - LoanApplication, 397

EF. See Entity Framework

email, 451

- BuildReturnMessageFrom, 52
- MailMessage, 41

EmailNotificationService, 63

- ReturnOrderService, 55–57
- Services folder, 55

Embedded Resource

- NHibernate, 293, 311
- XML, 293

Employee, 439–440

Employer

- Borrower, 216–219
- Borrowers folder, 213–215

Employers, 387

encapsulation, 18

- C#, 442, 443–446
- separation of concerns, 7

EnglishPlacesOfInterestDictionary, 134, 137, 147–148

- Pluggable, 156

enterprise, 9–14

- architecture, 5–6
- ASP.NET, 328
- code, 15–30
- design, 3–14
- development, 6–9
- middleware, 169–172

Enterprise Library, 160

entities

- aggregate roots, 199–200
- C#, 428
- DDD, 198–199
- EF, 273
- LinqToSQL, 273
- NHibernate, 300

Entity Designer, 273

Entity Framework (EF), 16, 269–283

- ADO.NET, 270
- ApplicationDetails, 270
- ASP.NET MVC, 382
- conceptual layer, 284
- debugging, 283
- entities, 273
- LinqToSQL, 269
- vs. LinqToSQL, 284–299
- Model folder, 270
- .NET Framework, 270
- PI, 285
- query, 269

- relational databases, 269

- SP1, 270

- Visual Studio, 270

EntityRef, 285

EntitySet, 285

enumeration, 224

ET.NET, 15

events, C#, 431

ExecutePaymentCalculation(), 350, 364

ExpectedException, 86

Extensible Application Markup Language (XAML), 177

- MVP, 361–365

Extensible Markup Language (XML), 160

- ASP.NET MVC, 382

- Asset.hbm.xml, 306

- BugTracker.dbml, 259

- C#, 384

- Debt.hbm.file, 307

- Embedded Resource, 293

- Java Struts, 324

- LinqToSQL, 254, 285

- middleware, 170–171

- Model folder, 291–293

- NHibernate, 285, 291, 306–308

- SOAP, 171

- Spring.Net, 160

- web services, 171

extension methods, 128, 395

- C#, 437–438

- inheritance, 437

Extract Class, 47

Extract Interface, 63

Extract Method, 47

- IsAWinner, 101

- PlaceMarketAt, 91

- ReturnOrderService, 48–50

- SendReturnOrderNotificationFor, 53

- Status, 95

Extract SuperClass, 47

Extreme Programming, 5

F

Factory

- abstraction, 139

- code, 140

- dependencies, 139

- FactoryPatternTests, 139

- TouristInformationService, 138, 139

Factory pattern, 138–141

- Gang of Four, 138
- Service Locator pattern, 144

FactoryPatternTests, 139**Fake implementation, 66****favoriteColor, 368****Find, 384**

- Repository, 382

FindInterestingPlacesNear, 131**FindLoanApplicationBy (Guid
loanApplicationId, 246****firstName, 368****flexibility, 6, 416**

- MVP, 367
- ReturnOrderService, 45, 70
- WPF, 423

float, 428**fluent interface**

- NHibernate, 285
- Registry, 157
- StructureMap, 151–153

foreach, 381**ForeignKeyConstraint, 452****FormCollection, 378****Fowler, Martin, 242–244, 286****front end, 321–336**

- ORMs, 319
- Web Forms, 262

G**GameStatus, 93****Gang of Four, 138****garbage collection, 426–427****Generate From Database, 270****GenerateOffer, 239****GermanPlacesOfInterestDictionary, 139****GET, 368****GetAllLoans, 384**

- LoanApplication, 385, 386
- Repository, 382

GetBrokenBusinessRules, 393**GetBrokenRules, 401****GetCredentials(), 448****GetCustomer, 419****GetData, 175**

- QLSql, 20

GetDateAndTimeOfDay, 442**GetFormattedText, 24****GetFriendlyTime, 175, 176****GetInformation(), 445****GetList, 23****GetMortgagePayment(), 350****GetMortgagePayment, 351****GetNewApplicationID, 395****GetOfferInPrincipal, 240, 242****GetOfferInPrincipal(), 241****GetOfferInPrincipalFor (Guid
loanApplicationId), 246****GetPaymentLabel(), 350****GetPrice, 21****GetTimeOfDay, 443****GetTimeOfDay(), 443****GetTimeStamp(), 443****GetTouristServiceSatelliteMap
English, 139****GetWorkingApplication(), 406**

- Edit, 410

Global.asax, 374

- LoanApplication, 400
- Solution Explorer, 385

graphical user interfaces (GUIs), 321–322

- HTML, 321
- Internet, 321
- RAD, 322

GUIs. See graphical user interfaces**H****HandlerForMyDelegateClas1, 433****Hanson, David Heinemer, 331****HasOfferExpired, 242****helper method**

- GetOfferInPrincipal(), 241
- LoanApplicationTests, 232–233

Hibernate, 12. See also NHibernate

- open source, 285

HolidayRequest, 119

- CanGrant(), 124
- HolidayRequestService, 120–121, 122
- IHolidayRequestRepository, 126
- Save, 125

**HolidayRequestService, HolidayRequest,
120–121, 122****HolidayRequestServiceHandRolledTests,
122, 125–126****HolidayRequestServiceRhinoMockTests,
123–124**

- Rhino Mocks, 126–127

Hollywood principle, 144

HomeController, 374

HQL, 286

HTML, 168

- ASP.NET, 326
- attributes, 368
- COM, 326
- Default.aspx, 278–279
- GUIs, 321
- HTTP, 368
- MVC, 326, 421, 451
- MVP, 343, 346
- View (MVC), 370

Html, 349

Html.ActionLink, 407

HtmlControls, 349

HTMLHelper, 381

HtmlHelper, 397–398

HTTP, 168, 326, 368–369

- attributes, 368
- HTML, 368
- MVC, 369
- SOAP, 171

HttpModules, 334

HttpRequest, 449

HttpResponse, 449

HttpServerUtility, 449

IApplicationContext, 29

IBankAccount, 209

iCalcList, 21–22, 24

IDE. See interactive development environment

Identity Map

- DataContext, 253
- ORMs, 253

Iesi.Collections.dll, 291, 301

IHolidayRequestRepository, 120, 121

- HolidayRequest, 126

IHolidayRequestValidator, 120, 121, 122

IIS. See Internet Information Server

ILoanApplicationRepository, 385

IMap

- SatelliteMap, 147
- TouristInformationService, 135
- TouristInformationService
Factory, 140
- using, 155

_IMort, 349

IMortgageCalculator, 364

IMortgageCalculator.cs, 347

impedance mismatch issue, 252

implicitly local typed variables, 436–437

Index

- Controller (MVC), 378
- LoanApplication, 386–387
- LoanApplicationController, 379

Index.aspx, 380

- LoanApplication, 397

inheritance, 395

- C#, 442–443
- extension methods, 437
- OO, 443

Initialize, 152

INotificationService, 64

- dependencies, 66
- DI, 65–66
- Fake implementation, 66
- Process, 118

In_Progress, 394

instances, 440–441

int, 428

integers, 170

interactive development environment (IDE), 16

- ASP.NET, 326
- WCF, 172

Interface, WCF, 172

interfaces. See also fluent interface; graphical user interfaces; user interface

- Address, 211
- BankAccount, 211
- DAL, 251
- declaration, 63, 172
- NHibernate, 301
- TouristInformationService, 134
- user-defined, 429

Internal, 429

Internet

- application servers, 168
- database servers, 168
- GUIs, 321
- middleware, 168–169

Internet Information Server (IIS), 9, 169

Internet Information Services, 168

Internet Server Application Programming Interface (ISAPI), 9, 169

- C++, 326

InvalidArgumentException, 111

Inversion of Control container (IoC), 25–29, 131–161, 144–146

code, 418
 DI, 64
 vs. DI, 146–149
 Spring.Net, 422
 StructureMap, 146
 TouristInformationService, 145

invoke, 433

IoC. See Inversion of Control container

IPaymentGateway, 118

IPlacesOfInterestDictionary, 133–134
 GermanPlacesOfInterestDictionary,
 139
 StructureMap, 155–156
 TouristInformationService, 135

**ISAPI. See Internet Server Application
 Programming Interface**

IsAWinner, 98–102
 Extract Method, 101
 TicTacToeGame, 98

IService1, 173, 175–176

IService1.cs, 173

ISession, 285

ISimpleMortgageCalculator, 345

**ISimpleMortgageCalculatorView, 345,
 347, 364**
 WPF, 362

**ISimpleMortgageCalculatorView _View;,
 347**

**ISimpleMortgageCalculatorView.cs,
 358–359**

IStateManager.
 SetWorkingApplicationID, 412

IStateManager, 410

IsValidInterestRate, 213

IUnknown, 10

_IView, 349

IWorker, 447–448

J

J2EE. See Java 2 Enterprise Edition

Java, 9, 10–11
 NHibernate, 285
 RMI, 171

Java 2 Enterprise Edition (J2EE), 11, 169, 324

Java Message Service (JMS), 11
 middleware, 183

Java Runtime Environment (JRE), 333

Java Server Pages (JSP), 169, 325, 369

Java Struts, 324–325

actions, 324
 MVC, 324
 XML, 324

Java Swing, 324

JavaBeans, 324

JMS. See Java Message Service

jquery, 373

JRE. See Java Runtime Environment

JSP. See Java Server Pages

Just-in-Time, 426

K

keywords, 435–436
 instances, 440–441

L

lambdas
 C#, 437
 Moq, 128

**Language Integrated Query (Linq). See also
 LinqToSQL**
 LinqToSQL, 255
 Moq, 128
 relational databases, 255

lazy loading, 253
 NHibernate, 291, 301

lblTime, 330

LDAP. See Lightweight Directory Access Protocol

lib folder, 76, 119, 132
 NHibernate, 301
 NUnit.dll, 301
 NUnit.Framework.dll, 229
 StructureMap.dll, 150

life cycle, ASP.NET, 326–327, 378

**Lightweight Directory Access Protocol
 (LDAP), 170**

LinFu.DynamicProxy.dll, 291, 301

Linq. See Language Integrated Query

LINQ to Entities, 269

LINQ to SQL Designer, 258–260

LinqToSQL, 254–268
 Active Record pattern, 284
 ApplicationDetails.aspx, 264–265
 attributes, 285
 BugTrackerDB, 256–261
 BugTracker.Designer.cs, 276
 DAO, 255–261
 DataContext, 254–255, 285

LinqToSQL (continued)

- debugging, 268
- Default.aspx, 278–279
- default.aspx, 262–264
- EF, 269
- vs. EF, 284–299
- entities, 273
- EntityRef, 285
- EntitySet, 285
- Linq, 255
- metadata, 254, 285
- NHibernate, 290
- OO, 285
- PI, 285
- Solution Explorer, 257
- SQL Server, 256, 284
- Visual Studio, 255, 258
- Web Forms, 262, 264–265, 278–279
- XML, 285
- List, 200, 439–440**
- ListBox, 449**
- List<LoanApplication> FindAll(), 246**
- Load, MVP, 331**
- Loan, 198**
- Loan Application Aggregation, 204**
- LoanAmount, 345**
- LoanApplication, 204, 374**
 - ActionLink, 410, 413
 - aggregate roots, 244–245
 - classes, 224–228
 - Create, 394, 396, 407
 - Create.aspx, 394, 396
 - custom exceptions, 224
 - DI, 246
 - Edit, 408, 412
 - Edit.aspx, 397
 - enumeration, 224
 - GenerateOffer, 239
 - GetAllLoans, 385, 386
 - Global.asax, 400
 - Index, 386–387
 - Index.aspx, 397
 - metadata, 309–311
 - NUnit.Framework.dll, 229
 - Prequalify, 238
 - ProductIsAvailableSpecification, 237
 - solution folder, 229
 - Stubs folder, 231–232
 - unit testing, 229–242

- LoanApplicationController, 378, 385**
 - Index, 379
- LoanApplicationController.cs, 395**
- LoanApplication.hbm.xml, 309–311**
- LoanApplicationRecord, 397**
- LoanApplicationRepository, 382–384, 385**
 - NHibernate, 313–314
- LoanApplications folder**
 - Asset, 219–220
 - Debt, 220–222
 - Offer, 222–223
- LoanApplicationService, 244–245, 246–247**
- LoanApplicationTests, 230, 238**
 - helper method, 232–233
- LoanApplication.Update(), 400**
- LoanController, routing, 385**
- LoanMonthlyPaymentCalculator, 234–235**
- LoanRepository, 243**
- Loans.xml, 384**
- Loan-to-Value (LTV), 392**
- Location, 135**
- log4net.dll, 291, 301**
- LogForJ, 12**
- Login, 450**
- LoginView, 450**
- long, 428**
- loose coupling**
 - code, 17–23
 - DI, 28, 135
 - MVC, 323
 - MVP, 328
 - TouristInformationService, 138
- LTV. See Loan-to-Value**

M

- Mac OS X, 324**
- MailMessage, email, 41**
- Main, 18, 19**
- mainframes, 166**
- maintainability, 7–9, 416–417**
 - ReturnOrderService, 45–46, 70–72
- ManualWireUpTest, 135**
- Mapping Specification Language (MSL), 275**
- MappingFiles, 306–308**
- MapRoute, 385–386**
- Martin, Robert C., 61**
- MbUnit, 110**
- McCarthy, Tim, 198**

- McLanahan, Craig, 324**
- Membership, 450**
- Membershipuser, 450**
- Message Oriented Middleware (MOM), 11, 182–183**
 - dependencies, 182
 - middleware, 182–183
- metadata**
 - LinqToSQL, 254, 285
 - LoanApplication, 309–311
 - NHibernate, 306
 - StructureMap, 157–159
- Method Injection, 64**
 - ReturnOrderService, 67–68
- MFC. See Microsoft Foundation Classes**
- Microsoft Foundation Classes (MFC), 169**
 - C++, 324
- Microsoft Intermediary Language (MSIL), 426**
- middleware, 165–184, 419–420**
 - application servers, 169
 - creating, 185–250
 - DDD, 198–249
 - enterprise, 169–172
 - Internet, 168–169
 - JMS, 183
 - MOM, 182–183
 - tiered design, 166–167
 - XML, 170–171
- MLXML, 28**
- MockHolidayRequestRepository, 124–125**
- mocking, 127–128**
 - MVP, 331, 351–352
 - stub objects, 123
 - TDD, 331
 - unit testing, 113
- MockRepository, 124**
- mocks.Stub<IHolidayRequestValidator>(), 124**
- Model (MVC), 369–370**
 - mortgage loan application, 374–376, 401–413
 - UI, 421
- Model (MVP), 338**
 - UI, 421
- Model First, 284**
 - NHibernate, 299
 - persistence, 420
- Model folder, 35–37**
 - ApplicationRepository, 294–295
 - BugDAO, 261–262, 276–277
 - EF, 270
 - NHibernate, 288, 291, 294–295
 - Repository, 294–295
 - SessionProvider, 293–294
 - XML, 291–293
- ModelState, 395**
- ModelState.AddModelError, 401**
- ModelStateDictionary**
 - AddRuleVilocations, 438
 - BrokenBusinessRules, 395
 - Controller (MVC), 394
 - ModelState, 395
 - View (MVC), 395
- Model-View-Controller pattern (MVC), 322–324, 367–414. See also ASP.NET MVC**
 - abstraction, 323
 - CRUD, 376
 - HTML, 326, 421, 451
 - HTTP, 369
 - Java Struts, 324
 - loose coupling, 323
 - Mac OS X, 324
 - Monorail, 333
 - mortgage loan application, 371–382
 - Observer pattern, 323
 - parameters, 370–371
 - reusability, 328
 - RoR, 368, 421
 - routing, 421
 - System.Web.Mvc, 451
 - UI, 328, 421
- Model-View-Presenter (MVP), 328–332, 337–365**
 - abstraction, 330
 - ASP.NET, 329, 331, 343
 - Click, 330
 - flexibility, 367
 - HTML, 343, 346
 - Load, 331
 - loose coupling, 328
 - mocking, 331, 351–352
 - Model, 338
 - Mortgage Calculator Model, 339–360
 - NUnit, 339
 - Page_Load(), 345
 - Page_Load, 349
 - reusability, 328
 - TDD, 328
 - testability, 367
 - UI, 328, 343, 421
 - unit testing, 350–360

Model-View-Presenter (continued)

- View, 338–339
- VS.NET, 339–341
- Web Forms, 343
- WPF, 361–365
- XAML, 361–365

modifiers, 429

modules

- Apache, 169
- code, 16–17
- DI, 64
- separation of concerns, 46–60
- third-party TDD, 111

MOM. See Message Oriented Middleware

Mono project, 12

Monorail, 333–335

- MVC, 333
- open source, 333

MonthlyPayment, 392, 393

monthlyPayment, 350

Moq, 128

MoreDetails, 274

Mortgage Application Processor, 423

Mortgage Calculator Model, 423

- MVP, 339–360

mortgage loan application, 197–249

- Controller (MVC), 376–378
- Model (MVC), 374–376, 401–413
- MVC, 371–382
- NHibernate, 299–318
- View (MVC), 378–382

MS Test, 110

MS Unity, 160

msi installer, 75

MSIL. See Microsoft Intermediary Language

MSL. See Mapping Specification Language

MVC. See Model-View-Controller pattern

MVP. See Model-View-Presenter

MyDelegateClass1, 433

MyHomePage.aspx, 334

MyinnerNamespaceClass, 435

N

namespaces, 434–436

.NET, 11–14. See also ADO.NET; ASP.NET; Visual Studio.NET

- ADO.NET, 18–19
- C#, 425–453
- COM, 427

- DateTime, 330
- Moq, 128
- NHibernate, 382
- RAD, 417
- SourceForge.NET, 12–13, 286
- Spring.Net, 28–29, 160
- WSDL, 171

.NET Domain Driven Design with C#: Problem - Design - Solution (McCarthy), 198

.NET Framework

- ASP.NET, 325
- ASP.NET MVC, 334
- EF, 270
- PI, 285

.NET Runtime Environment, 426

.NET SDK, 171

- C#, 448–452
- COM, 427
- svcutil.exe, 180

.NET Winforms, 324

new, 429

New Solution Folder, 76

NHibernate, 255, 285–299

- aggregates, 312–313
- App.Config, 314–315
- App_Data, 290, 302
- Application folder, 288
- ApplicationDetail.aspx, 297–298
- ApplicationRepository, 294–295
- attributes, 285
- BaseRepository, 312–313
- Bug folder, 288
- Can_Add_Product_To_Repository(), 317
- DataContext, 285
- debugging, 299
- default.aspx, 295–296
- Developer folder, 288
- Embedded Resource, 293, 311
- entities, 300
- fluent interface, 285
- HQL, 286
- interfaces, 301
- ISession, 285
- Java, 285
- lazy loading, 291, 301
- lib folder, 301
- LinqToSQL, 290
- LoanApplicationRepository, 313–314
- MappingFiles, 306–308
- metadata, 306

- Model First, 299
 - Model folder, 288, 291, 294–295
 - mortgage loan application, 299–318
 - .NET, 382
 - NUnit.dll, 301
 - PI, 285
 - Product.hbm.xml, 307–308
 - ProductRepository, 313–314
 - Query Object, 286
 - relational databases, 301
 - Repository, 294–295
 - Server Explorer, 302
 - SessionProvider, 293–294, 304, 314–315
 - unit testing, 315–318
 - value objects, 300, 306
 - Visual Studio, 301
 - Web Forms, 297–298
 - web.config, 298–299
 - XML, 285, 291, 306–308
 - NHibernateBugTracker, 288**
 - NHibernate.ByteCode.LinFu.dll, 291, 301**
 - NHibernate.dll, 291, 301**
 - NHibernateProductTests, 315**
 - NMock, 128**
 - NonGraphicalMap, 140**
 - normalization, 167**
 - NotificationService, 65**
 - n-tiered architectures, 168, 169–170**
 - NullReferenceException, 241**
 - NUnit, 75**
 - ExpectedException, 86
 - MVP, 339
 - SimpleIoCContainer_Can_Create_An_Instance_Of_SatelliteMap(), 148
 - TDD, 110
 - Visual Studio, 79
 - NUnit.dll, 301**
 - NUnit.Framework.dll, 76, 132**
 - lib folder, 229
 - LoanApplication, 229
 - nvarchar(50), 257, 270, 290**
 - nVelocity, 334, 378**
- O**
- ObjA, 26, 28**
 - ObjB, 26, 28**
 - ObjC, 26, 28**
 - ObjD, 26, 28**
 - object, 428**
 - object initializers, 438–440**
 - object relational mappers (ORMs), 252–253**
 - abstraction, 253
 - caching, 253
 - front end, 319
 - Identity Map, 253
 - persistence, 420
 - ObjectDataSource, 449**
 - object-oriented language (OO), 15**
 - C#, 440
 - inheritance, 443
 - LinqToSQL, 285
 - ObjectRelational Designer, 259**
 - objects**
 - C#, 440–442
 - user-defined, 429
 - Observer pattern, 323**
 - Offer, 222–223**
 - Offer property, 240**
 - OO. See object-oriented language**
 - open source, 9**
 - Castle Project, 160
 - Hibernate, 285
 - Java Struts, 324
 - Monorail, 333
 - NMock, 128
 - StructureMap, 146
 - [OperationService], 172**
 - Order**
 - Active Record pattern, 189–190
 - Cancel, 190
 - Place, 190
 - Process, 190
 - ProcessOrderBy, 196
 - OrderDTO, 197**
 - ORMs. See object relational mappers**
 - overrides, 430**
- P**
- Page_Load, 329**
 - MVP, 349
 - SimpleMortgageCalculatorView.aspx.cs, 350
 - this, 349
 - Page_Load(), MVP, 345**
 - parameters, 370–371**

parent-child relationships, 170

passwords, 450

PaymentGateway, 63

Process, 59

ReturnOrderService, 59

Services folder, 58

PaymentTransactionID, 37

PayPal, 34

PayPalGateway, 38–39, 41, 58

ReturnOrderService, 41, 44, 46

PayPalPayment, 38

Perl, 169

persistence, 420

DAL, 420

DDD, 420

Model First, 284, 420

ORMs, 420

Repository, 284

persistence ignorant (PI), 194, 285

EF, 285

LinqToSQL, 285

.NET Framework, 285

NHibernate, 285

PHP, 169, 369

PI. See persistence ignorant

PIC, 166

Place, 190

PlaceMarketAt, 86, 88

Extract Method, 91

PlaceOfInterest, 135

PlaceOrder, 197

PlacesOfInterest, 133

PlaceOfInterest, 135

**plain old common runtime object (POCO),
194, 285**

Player.x, 81–82

player.x, WhoseTurn(), 90

Pluggable

EnglishPlacesOfInterestDictionary,
156

Plugin, 159

StructureMap, 155–156

Plugin, 159

PluginFamily, 155–156

POCO. See plain old common runtime object

polymorphism, C#, 446–448

POST, 368

Potel, Mike, 328

Prequalify, 230, 234

LoanApplication, 238

Presenter (MVP), 339

UI, 421

**PrincipleInterestTaxesAnd
Insurance, 393**

Private, 429

Process

Domain Model pattern, 193

INotificationService, 118

IPaymentGateway, 118

Order, 190

PaymentGateway, 59

ReturnOrder, 41

ReturnOrderService, 59, 118

ProcessOrder, 419

ProcessOrderBy, 196

Product, 35, 41, 204, 211–213

Product Aggregation, 204

Product.hbm.xml, 307–308

ProductIsAvailableSpecification, 235

LoanApplication, 237

ProductRepository, 41, 63, 243

NHibernate, 313–314

ReturnOrderService, 44

save, 41

ProductService, 246–247

DIP, 61–62

ReturnStockFor, 50

Services folder, 50

Visual Studio, 62

Program, 19

properties, C#, 430

Protected, 429

proxy classes, 171–172

Public, 429

Q

QLSql, GetData, 20

Query Object, 286

QueryInterface, 10

R

**RAD. See rapid application development
rapid application development (RAD), 4, 9**

GUIs, 322

.NET, 417

RoR, 332

Record, 124

- Reenskaug, Trygve, 322**
- Refactor Pro, 117**
- refactoring**
 - code, 47–48
 - ReturnOrderService, 54–55
 - TDD, 116–117
 - Visual Studio, 47–48
- RegisterRoutes, 385**
- Registry**
 - fluent interface, 157
 - StructureMap, 151
- relational databases, 166–167**
 - Data First, 254
 - EF, 269
 - Linq, 255
 - NHibernate, 301
- Release, 10**
- reliability, 6, 416**
- Remote Method Invocation (RMI), 171**
- remote procedure call (RPC), 171**
- Repositories folder, 38–39**
- Repository, 41**
 - ApplicationRepository, 294–295
 - DAO, 284
 - Find, 382
 - GetAllLoans, 382
 - Model folder, 294–295
 - NHibernate, 294–295
 - persistence, 284
 - Save, 382
- Repository**
 - aggregates, 243–244
 - DDD, 242–244
 - Domain Model pattern, 194
- __repository, 385**
- Representational State Transfer (REST), 378**
- ReSharper, 117**
- Resolve, 142**
- REST. See Representational State Transfer**
- ReturnItem, 36**
- ReturnOrder, 37–38**
 - Process, 41
 - ServiceProductRepository, 51
- ReturnOrderRepository, 38, 41, 63**
 - ReturnOrderService, 44
 - save, 41
- ReturnOrderService, 39–44**
 - abstraction, 64
 - Constructor Injection, 65–66, 68–69
 - dependencies, 44, 60, 64
 - EmailNotificationService, 55–57
 - Extract Method, 48–50
 - flexibility, 45, 70
 - maintainability, 45–46, 70–72
 - Method Injection, 67–68
 - NotificationService, 65
 - PaymentGateway, 59
 - PayPalGateway, 41, 44, 46
 - Process, 59, 118
 - ProductRepository, 44
 - refactoring, 54–55
 - ReturnOrderRepository, 44
 - ReturnStockFor, 50
 - reusability, 45, 70
 - rigidity, 45, 70
 - separation of concerns, 45, 70
 - setter injection, 66–67
 - SRP, 47
 - WorldPayGateway, 42–44
- ReturnStockFor, 48–50**
 - ProductService, 50
 - ReturnOrderService, 50
- reusability, 7, 416**
 - MVC, 328
 - MVP, 328
 - ReturnOrderService, 45, 70
- Rhino Mocks, 123, 128**
 - HolidayRequestServiceRhinoMockTests, 126–127
 - MockRepository, 124
- rigidity, ReturnOrderService, 45, 70**
- RMI. See Remote Method Invocation**
- RoadMap, 140**
- roleManager, 450**
- Roles, 450**
- RoR. See Ruby on Rails**
- RouteCollection, 385**
- routing**
 - ASP.NET MVC, 370–371
 - LoanController, 385
 - MVC, 421
- Row, 84**
- row, 92**
- RPC. See remote procedure call**
- Ruby on Rails (RoR), 331–332**
 - MVC, 368, 421
 - RAD, 332
 - UI, 421

runat=server, 349
runtime, DI, 64

S

SatelliteMap, 134, 137

IMap, 147
StructureMap, 155

Save, 384

HolidayRequest, 125
Repository, 382

save

ProductRepository, 41
ReturnOrderRepository, 41

Save (LoanApplication

loanApplication), 246

sbyte, 428

scope, C#, 429–430

script files, 169, 326

Sealed, 430

security, 170, 450

DTO, 184

SendReturnOrderNotificationFor, Extract Method, 53

SendWorkerDataToWebService(), 448

separation of concerns, 7, 416

DTO, 184
encapsulation, 7
modules, 46–60
ReturnOrderService, 45, 70

Separation of Concerns Principle (SoC), 47

server, 345

Server Explorer

BugTrackerDB, 256
NHibernate, 302

ServerClick

ASP.NET, 350
HtmlControls, 349

service contract, WCF, 172

service layer, 195–196

Service Locator pattern, 142–144

dependencies, 142
Factory pattern, 144
TouristInformationService, 144

Service Oriented Architecture (SOA), 183–184

Service Pack 1 (SP1), 16

ASP.NET MVC, 371
EF, 270

[ServiceContract], 172

ServiceLocatorPatternTest, 143–144

ServiceProductRepository, 51

Services folder

EmailNotificationService, 55
PaymentGateway, 58
ProductService, 50

Service.svc, 175–176

Servlets, 169, 325

SessionProvider, 293–294, 304, 314–315

Model folder, 293–294

setter injection, 64

ReturnOrderService, 66–67

SetUp, StructureMap, 152, 157

SetUpResult.For, 124

SetWorkingApplication(), 406

SharpDevelop, 16

ShippingMethod, 193

short, 428

Simple Mail Transfer Protocol (SMTP), 41, 451

Simple Object Access Protocol (SOAP)

HTTP, 171
XML, 171

SimpleIoCContainer, 146–149

SimpleIoCContainer_Can_Create_An_Instance_Of_SatelliteMap(), 147
NUnit, 148

SimpleIoCContainerTests, 147

TouristInformationService, 148–149

SimpleMortgageCalculator, 392

SimpleMortgageCalculator.cs, 348, 360

SimpleMortgageCalculatorPresenter, 348

SimpleMortgageCalculatorPresenter.cs, 359

SimpleMortgageCalculator_Test, 353–354

SimpleMortgageCalculatorView, 346

SimpleMortgageCalculatorView.aspx, 356–357

SimpleMortgageCalculatorView.aspx.cs, 345, 357–358

Page_Load, 350

SimpleMortgageCalculator_View

.xaml, 362

SimpleMortgageCalculator_View.xaml.cs, 363

SimpleWebReference, 181

Single Responsibility Principle (SRP), 45, 71

ReturnOrderService, 47

Site.css, 400

site.css, 373

Six Sigma, 5

SMTP. See Simple Mail Transfer Protocol

- SOA. See Service Oriented Architecture**
- SOAP. See Simple Object Access Protocol**
- SoC. See Separation of Concerns Principle**
- Solicitor, 213**
- Solicitor.hbm.xml, 308–309**
- Solution Explorer**
 - AccountController, 374
 - Global.asax, 385
 - HomeController, 374
 - LinqToSQL, 257
 - Window1.Xaml, 181
- solution folder, 76, 132**
 - LoanApplication, 229
- SourceForge.NET, 12–13, 286**
- sourceforge.net, 150**
- SP1. See Service Pack 1**
- Spark, 378**
- SpecialHandler, 435**
- SpecialMethod, 433**
- SpecialMethod(), 433**
- SpecialWorker, 448**
- springframework.net, 160**
- Spring.Net, 28–29**
 - IoC, 422
 - XML, 160
- SQL Server, 450**
 - LinqToSQL, 256, 284
- SqlDataSource, 449**
- SRP. See Single Responsibility Principle**
- SSDL. See Stored Schema Definition Language**
- StateManager, 406**
- _statemanager.**
 - GetWorkingApplication(), 407
- Static, 429**
- static void Main(), 442**
- Status, 95**
- Stored Schema Definition Language (SSDL), 275**
- string, 428**
- strings, 170**
- structs, user-defined, 429**
- StructureMap, 150–159**
 - ASP.NET, 159
 - attributes, 153–157
 - fluent interface, 151–153
 - Initialize, 152
 - IPlacesOfInterestDictionary, 155–156
 - metadata, 157–159
 - Pluggable, 155–156
 - PluginFamily, 155–156
 - Registry, 151
 - SatelliteMap, 155
 - SetUp, 152, 157
 - TouristInformationService, 153, 159
- StructureMap, 146**
 - configSections, 159
- StructureMap_Attribute, TestsTouristInformationScanningRegistry, 156–157**
- StructureMap_AttributeTests, 153–157**
- StructureMap_Can_Create_An_Instance_Of_EnglishPlacesOfinterestDictionary(), 152**
- StructureMap.Config, 157**
- StructureMap_ConfigurationFileTests, 157–159**
- StructureMap.dll, 155**
 - lib folder, 150
- StructureMap_FluentTests, 151**
- StructureMap_Should_Create_A_Valid_SatelliteMap(), 152**
- Struts. See Java Struts**
- stub objects**
 - dependencies, 118
 - mocking, 123
 - unit testing, 113
- StubAddress, 231–232**
- StubBankAccount, 231–232**
- StubHolidayRequestRepository, 121**
- Stubs folder, 231–232**
- SubmitChanges(), 261**
- SubmitForPrequalification(LoanApplication loanApplication), 246**
- svcutil.exe, 180**
- System.Collections.IEnumerable, 439**
- System.Data, 451–452**
- System.Data.OleDb, 452**
- System.Data.SqlClient, 452**
- %systemdrive%\%programfiles\directory%\NUnity 2.4.8\bin, 76, 119**
- System.EventHandler, 433**
- System.IO, 452**
- System.NET, 452**
- System.NET.Mail, 451**
- System.Net.Mail, 44**
- System.Web, 449**
- System.Web.Mobile, 450–451**
- System.Web.Mvc, 451**
- System.Web.Mvc.Ajax, 451**

System.Web.Mvc.Html, 451
System.Web.Security, 450
System.Web.Services, 449–450
System.Web.UI, 449
System.Web.UI.HtmlControls, 449
System.Web.UI.MobileControls, 451
System.Web.UI.MobileControls.
 Adapters, 451
System.Web.UI.WebControls, 449
System.Xml, 452

T

Taligent, 328
TDD. See Test Driven Development
TemperatureChanged, 323
[Test], 77
Test Driven Development (TDD), 6, 24, 73–129
 abstraction, 111
 boundary conditions, 111
 code, 417–418
 dependencies, 118–127
 MbUnit, 110
 mocking, 331
 Model First, 284
 MS Test, 110
 MVP, 328
 NUnit, 110
 refactoring, 116–117
 testability, 115
 TestDriven.Net, 110
 third-party modules, 111
 Tic Tac Toe, 74–116
 trivial code, 111
testability, 6, 18
 MVP, 367
 TDD, 115
TestDriven.Net
 TDD, 110
 Visual Studio, 110
[TestFixture], 77
testing. See unit testing
TestsTouristInformationScanningReg,
 StructureMap_Attribute, 156–157
Text, 330
TextBox, 449
third-party modules, TDD, 111
this, 437, 443
 Page_Load, 349

Tic Tac Toe
 Class Library, 76
 requirements, 75–109
 TDD, 74–116
 Visual Studio, 76
TicTacToeGame, 80
 CanPlaceMarketAt, 85–86
 IsAWinner, 98
TicTacToeGameTests, 86
tiered design, middleware, 166–167
Top Down, 284
TouristInformationRegistry,
 151–152
TouristInformationService, 131
 abstraction, 139
 code, 137
 Constructor Injection, 135
 dependencies, 134, 137, 139
 Factory, 138, 139
 IMap, 135
 interfaces **prior**, 134
 IoC, 145
 IPlacesOfInterestDictionary, 135
 loose coupling, 138
 Service Locator pattern, 144
 SimpleIoCContainerTests, 148–149
 StructureMap, 153, 159
TouristInformationServiceFactory,
 138–139
 GetTouristServiceSatelliteMap
 English, 139
 IMap, 140
Transaction Script, 186–188, 419
 DAO, 254
 vs. Domain Model, 194
 service layer, 195
 uses for, 194
TransactionId, 57
triangulation, 85
 boundary conditions, 115
 unit testing, 114
 WhoseTurn, 88
trivial code, 111
truck factor, 116
types, C#, 428

U

UI. See user interface
uint, 428

ulong, 428
UniqueConstraint, 452
unit of work pattern, 253
unit testing, 23–24. See also Test Driven Development
 Development
 automation, 113, 114–116
 debugging, 113
 dependencies, 113, 316
 independence of, 114
 LoanApplication, 229–242
 mocking, 113
 MVP, 350–360
 NHibernate, 315–318
 repeatability of, 113
 set up for, 114
 simplicity of, 113
 speed of, 112
 stub objects, 113
 triangulation, 114
 truck factor, 116
UNIX, 169
UpdateMe, 402
user interface (UI), 7, 321–322, 421–422
 ASP.NET, 368
 Model (MVC), 421
 Model (MVP), 421
 MVC, 328, 421
 MVP, 328, 343, 421
 Presenter (MVP), 421
 RoR, 421
 View (MVC), 421
 View (MVP), 421
ushort, 428
using, 155
using StructureMap, 156
using-namespace-directive, 437–438

V

value objects
 address, 206
 BankAccount, 206
 credit score, 215–216
 DDD, 199
 NHibernate, 300, 306
 Solicitor, 213
VB. See Visual Basic
VBA. See Visual Basic for Applications
View (MVC), 370
 .aspx, 378

 DHTML, 370
 HTML, 370
 ModelStateDictionary, 395
 mortgage loan application, 378–382
 UI, 421
View (MVP), 338–339
 UI, 421
View Code, 181
ViewState, 326
Virtual, 430
Visual Basic (VB), 9–10, 169
 ActiveX, 183, 325
 COM, 325
Visual Basic for Applications (VBA), 9
Visual Studio
 Class Library, 76
 Class1.cs, 34
 default.aspx, 295–296
 EF, 270
 Entity Designer, 273
 LinqToSQL, 255, 258
 NHibernate, 301
 NHibernateBugTracker, 288
 NUnit, 79
 ProductService, 62
 refactoring, 47–48
 ReSharper, 117
 TestDriven.Net, 110
 Tic Tac Toe, 76
Visual Studio.NET (VS.NET), 16
 ASP.NET MVC, 371, 373
 MVP, 339–341
Visual Web Developer, 371
VS.NET. See Visual Studio.NET

W

WCF. See Windows Communication Foundation
Web Forms
 ApplicationDetail.aspx, 297–298
 ApplicationDetails.aspx, 264–265, 280–281
 ASP.NET, 343
 front end, 262
 LinqToSQL, 262, 264–265, 278–279
 MVP, 343
 NHibernate, 297–298
web services, 171
 WCF, 172–182
 XML, 171

Web Services Description Language (WSDL)

Web Services Description Language (WSDL), 171

.NET, 171

Web.config, 384

web.config

ASP.NET, 173

NHibernate, 298–299

WCF, 173

WebRequest, 452

WebResponse, 452

WhoseTurn, 85

triangulation, 88

WhoseTurn(), 81–82

player.x, 90

Window1.Xaml, 177

Solution Explorer, 181

Windows Communication Foundation (WCF), 16, 172, 423

web services, 172–182

web.config, 173

Windows Presentation Foundation (WPF), 16

flexibility, 423

ISimpleMortgageCalculatorView, 362

MVP, 361–365

Windsor Container, 160

Worker, 447

WorldPay, 41

WorldPayGateway, 41–44, 58

ReturnOrderService, 42–44

WPF. See Windows Presentation Foundation

WSDL. See Web Services Description Language

wsHttpBinding, 175

X

XAML. See Extensible Application Markup Language

Xaml, 364

XML. See Extensible Markup Language

XML Schema (XSD), 170

XMLDocument, 448

XSD. See XML Schema

Z

zero-based index, 83–84