

INDEX

A

- abstract steps, 104
- AbstractClass class, 104
- abstractions
 - dependencies, 128
 - DIP and, 128
- account controllers, Agatha's Clothing Store, 590–600
- AccountActive property, 122
- AccountService class, 201
- ActionCommands, 322–326
- ActionResult base class, 350
- Active Record pattern, 8
 - auto-generating business model, 58
 - business objects, 58
 - Castle ActiveRecord project, 58
 - one-to-one mapping, 58
 - Ruby on Rails framework, 58
- adapter patterns, refactoring to, 22–25
- Add method, 206
- Address class, 95, 562
- AdjustPriceWith method, 129
- ADO.NET
 - commands, populating with queries, 221
 - Entity Model, creating, 257
 - raw repository, 283
- AgathaHtmlHelper class, 489–490
- Agatha's Clothing Store
 - account controllers, 590–600
 - architecture, 424–427
 - application configuration settings, 438–439
 - domain layer supertype, 428–432
 - e-mail service, 441–443
 - helper classes, 443–444
 - infrastructure, 427–444
 - logging, 439–441
 - query object pattern, 433–438
 - authentication views, 604–609
 - AutoMapper, 427
 - basket
 - controller, 538–558
 - creating, 520
 - delivery options, 518
 - domain model, 515–525
 - JSON, 545–546
 - persisting, 524–525
 - retrieving, 524–525
 - tables, 525
 - validation, 520
 - views, 538–558
 - Basket screen, 418–420
 - BasketService, 529–538
 - caching, 424
 - checkout, order model, 611–626
 - Checkout screen, 418, 423–424
 - controllers
 - checkout, 649–658
 - customer, 586–590
 - order, 649–658
 - payment, 649–658
 - Customer Account screen, 418, 421–423
 - customer membership
 - authentication service, 578–586
 - customer model, 561–567
 - customer service, 570–578
 - data tables, 567
 - NHibernate mappings, 567–570
 - views, 600–603
 - Delivery Address Management page, 422

Agatha's Clothing Store (*continued*)

- NHibernate, 427
 - mapping, 525–529
 - order data tables, 627
 - Order Detail page, 422
 - order NHibernate mappings, 627–630
 - order service, 631–640
 - orders placed, 422
 - payment, PayPal, 640–648
 - Payment screen, 418
 - Product Catalog, product catalog model, 446–450
 - Product Catalog screen, 417, 418–420
 - Product Detail page, 420
 - requirements, 417–418
 - Unit of Work pattern, 432–433
 - User Details Management page, 421
 - views
 - checkout, 658–668
 - order, 658–668
- aggregate root, 90, 428
- aggregates, 90
- AJAX (Asynchronous JavaScript and XML)
 - Firefox browser, 388–390
 - overview, 371
 - patterns, 372
 - Periodic Refresh, 372–390
 - Timeout, 374
 - URLs, unique, 390
- AJAX JavaScript library, 372
- Alexander, Christopher, 4
- algorithms
 - boolean, business logic, 118
 - encapsulating at objects, 113–114
 - skeleton structure, 103
 - swapping dynamically at run time, 113–114
- aliases, NHibernate, 463
- AmountToRefund property, 106
- AndSpecification class, 122–123
- Anemic Domain Model, 86–88
- antipatterns, Smart UI, 29, 30–35
- API key, 582
- APIs (application programming interfaces), Facade pattern and, 155

- application configuration settings, 438–439
- Application service, 90
- ApplyExtraDiscountsTo method, 33
- architecure
 - Agatha's Clothing Store, 424–427
 - application configuration settings, 438–439
 - domain layer supertype, 428–432
 - e-mail service, 441–443
 - helper classes, 443–444
 - logging, 439–441
 - query object pattern, 433–438
 - infrastructure, 427–444
 - planning, 29
- ASP.NET
 - membership, local authentication, 579–580
 - Web forms, MVP pattern, 293–315
- ASP.NET MVC framework, 345–361
- asynchronous client callbacks, 371
- atomic transactions, persistence, 200
- authentication
 - cookies, 585–586
 - forms, 578
 - Janrain, 581–585
 - local, ASP.NET membership, 579–580
 - views, Agatha's Clothing Store, 604–609
 - web-based, 581–585
- AutoMapper, 345
 - ViewModels, 356–361
- AutoMapper, Agatha's Clothing Store, 427
- autonomy of service methods, 155

B

- BankAccountView, 79
- base, behaviorial, and structural patterns
 - Gateway, 10
 - Null Object, 10, 25–26
 - Separated Interface, 10
- base classes
 - ActionResult, 350
 - Address, 562
 - controller, 350
 - downcasting, 144
 - inheriting from, 171

- Layer Supertype pattern, 125–128
 - LSP and, 139
 - ViewResult, 350
 - base controller, creating, 482
 - base types, substituting subtypes, 144
 - BaseAccountController class, 594
 - BaseController class, 542–543
 - BasePageView class, 539–540
 - BasePrice class, 100
 - BaseProductCatalogPageView class, 479
 - BaseView class, 346–347
 - Basket class, 114
 - creating orders, 636–637
 - mapping, 527
 - updating, 117
 - Basket folder, 299
 - basket view, implementing, 310
 - BasketController class, 547–550
 - BasketDiscountFactory class, 117
 - BasketDiscountMoneyOff class, 115
 - BasketItem class, 521–522
 - mapping, 527
 - BasketMapper class, 531, 545
 - BasketPresenter class, 303
 - BasketService class, 529–538, 537–538
 - BasketSummary.ascx, 551
 - BDD (Behavior-driven Design), 12
 - behavioral patterns, 4, 16–17
 - Bibeault, Bear, 372
 - Boodhoo, J.P., 317
 - bookmarks, user, 405
 - BookRepository class, 254–255
 - BookTitleRepository class, 254–255
 - boolean algorithms, business logic and, 118
 - BootStrapper class, 48, 331, 352–353
 - Brand class, 447
 - Brinkman, Joe, 372
 - browsers, favorite icon, 512–513
 - btnReserveTickets click event, 188
 - business entities
 - creating, 39
 - Entity Framework, 262
 - pending changes, 200
 - persistence, 194
 - retrieval, 194
 - business layer, 36–42
 - business logic and, 55
 - queries, Repository pattern, 194
 - business logic
 - boolean algorithms, 118
 - DAL, 194
 - UI and, 32
 - business objects, Unit of Work pattern, 196
 - business organizational patterns
 - Active Record pattern, 58–68
 - auto-generating business model, 58
 - business objects, 58
 - Castle ActiveRecord project, 58
 - one-to-one mapping, 58
 - Ruby on Rails framework, 58
 - Anemic Domain Model, 86–88
 - DDD (Domain-Driven Design), 88–91
 - Domain Model pattern, 68–86
 - Transaction Script pattern, 56–58
 - problems with, 56
 - small applications, 56
 - understandability, 56
 - business requirements, adding, 32
 - business rules, OrderItem entities, 617
 - business use cases, 237–240
- ## C
- C#, Class Library, adding, 18
 - caching, Agatha’s Clothing Store, 424
 - CalculateRefundFor method, 106
 - canceled orders, 111
 - CancelOrderState class, 111
 - case statements, State pattern and, 108
 - case study. *See* Agatha’s Clothing Store
 - Castle ActiveRecord project, 58
 - Castle MonoRail, 361–368
 - Castle MonoRail framework, downloading, 362
 - CatalogBaseController, 480–481
 - CategoryBrandView class, 397–398
 - CategoryList.ascx user control, 304
 - CategoryMapperExtensionMethods class, 348
 - CategoryProductPresenter class, 298–299
 - CategoryProductsPageHandler class, 338
 - CategoryView, 346

- CategoryView product view, 466–467
- centralization, Layer Supertype pattern and, 125
- Certification property, 137
- Chain of Responsibility pattern, 337–344
 - UML representation, 337
- chaining objects, 337
- checkout, order model, 611–626
- Checkout class, 649
- checkout controllers, 649–658
- checkout view, 658–668
- CheckoutController class, 652–655
- Checkout.master view, 658
- Class Library (C#), adding, 18
- classes
 - AbstractClass, 104
 - AccountService, 201
 - Address, 95
 - AgathaHtmlHelper, 489–490
 - AndSpecification, 122–123
 - BaseAccountController, 594
 - BaseController, 542–543
 - BasePageView, 539–540
 - BasePrice, 100
 - BaseProductCatalogPageView, 479
 - BaseView, 346–347
 - Basket, 114
 - BasketController, 547–550
 - BasketDiscountFactory, 117
 - BasketDiscountMoneyOff, 115
 - BasketItem, 521–522
 - BasketMapper, 531, 545
 - BasketPresenter, 303
 - BasketService, 529–538, 537–538
 - BookRepository, 254–255
 - BookTitleRepository, 254–255
 - BootStrapper, 48, 331, 352–353
 - Brand, 447
 - CancelOrderState, 111
 - CategoryBrandView, 397–398
 - CategoryMapperExtensionMethods, 348
 - CategoryProductPresenter, 298–299
 - CategoryProductsPageHandler, 338
 - Checkout, 649
 - Client, 120
 - Comment, 408
 - Component, 120
 - Composite, 120
 - CompositeSpecification, 124
 - ConcreteClassA, 104
 - ConcreteClassB, 104
 - ConcreteDecoratorA, 98
 - ConcreteStateA, 109
 - ConcreteStateB, 109
 - ConcreteStrategy, 114
 - Context, 108, 114
 - ControllerFactory, 352
 - CookieStorageService, 541
 - Courier, 516–517
 - Criterion, 215
 - CurrencyPriceDecorator, 101
 - Customer, 561–567
 - CustomerAccount, 119, 120
 - CustomerBusinessRules, 565
 - CustomerProxy, 209–210
 - CustomerRepository, 210
 - CustomerType, 40
 - CustomerView, 345
 - CustomHttpHandler, 330
 - DefaultProduct, 98
 - DeliveryAddress, 561–567
 - DeliveryAddressMapper, 572
 - dependent, creation, 94
 - DiscountType, 116–117
 - Display404PageCommand, 329
 - domain service class, 218
 - DomainEvents, 615
 - EFUnitOfWork, 265–266
 - EmployeeRepository, 212
 - EntityBase, 126, 516
 - ErrorLog, 175
 - Event, 161, 375
 - EventRepository, 178, 375–376, 377
 - Factory, 94
 - Factory Method pattern, 95
 - FaultyReturnProcess, 106
 - FrontController, 330
 - GetCategoryCommand, 323–324
 - GetCategoryProductsCommand, 324–325
 - GetFeaturedProductsResponse, 469–470
 - GetOrderRequest, 633

GetOrderResponse, 633
GetProductDetailCommand, 325
GetTopSellingProductsCommand, 323
HandlerFactory, 342
HasReachedRentalThresholdSpecification, 119
helper classes, 443–444
HomeController, 350
HomePagePresenter, 297–298
HtmlHelper, 490
IdentityMap, 212
InvalidAddressException, 565
ITicketService, 173
JsonModelBinder, 483–484
JsonProduct, 484
JsonRefinementGroup, 484
Leaf, 120
library domain model, 227
LibraryService, 240–243
Loan, 229
Member, 228
MemberRepository, 254–255
ModelRegistry, 291
ModifyBasketRequest, 533
ModifyBasketResponse, 533
NoBasketDiscount, 116
NotSpecification, 122–123
NullDeliveryOption, 519
NullDiscountStrategy, 38
NullObjectCache, 25–26
NWUnitOfWork, 252–253, 458
Order, 110–111, 620–621
OrderByClause, 215
OrderController, 651
OrderItem, 617
OrderItemView, 631
OrderNewState, 112
OrderQueryTranslator, 219
OrderRepository, 209
OrderService, 95
OrderServiceFactory, 286
OrderShippedState, 112
OrderState, 624–625
OrderSummaryView, 631
PaymentController, 656–657
PaymentFactory, 617
PaymentServiceBase, 141
Presentation models, 184–186
presenter, 47
Price, 36
Product, 36
ProductCollectionExtensionMethods, 101–102
ProductColor, 448
ProductController, 351, 363–364, 485–488
ProductDecorator, 98
ProductListExtensionMethods, 41
ProductListPresenter, 47
ProductListRequest, 43
ProductMapperExtensionMethods, 43
ProductRepository, 18, 45
ProductService, 18, 41, 44, 99, 102–103, 129–130
ProductSize, 448
ProductTitle, 449–450
ProductViewModel, 42
Query, 216
QueryName, 216
QueryTranslator, 254, 255–256
RefundRequest, 140
RefundResponse, 140
RefundService, 139
reply, 532
Repository, 253–254, 461
request, 532
RequestHandler, 337, 338
ResponseBase, 237
ReturnAction, 104–105
ReturnOrder, 105
ReturnProcessFactory, 107
ReturnProcessTemplate, 105–106
Route, 322
Service, 104, 107
SessionFactory, 250–252, 457–458
SetOrderPaymentRequest, 633
SetOrderPaymentResponse, 633
ShippingService, 517–518
ShopService, 349–350
State, 109
Strategy, 114
SubSystemA, 156
SubSystemB, 156
TicketReservation, 161

classes (*continued*)

- TicketService, 175
- TradeDiscountPriceDecorator, 100–101
- TradeDiscountStrategy, 37–38
- TransactionScope, 200
- UKShippingCourierFactory, 97
- UnitOfWork, 200
- ValueObjectBase, 431
- ViewStorage, 318
- ViewStorageFactory, 318, 333
- WebBasket, 300
- WebCommand, 326–327
- WebRequest, 320
- Client class, 120
- client proxy, 182–186
- clients
 - asynchronous client callbacks, 371
 - SOA example, 186–190
- code-behind files, 30
- Command pattern
 - parts, 315
 - requests, 315
 - UML representation, 316
- command registry, WebCommands list, 327
- Comment class, 408
- Commit method, 200
- compile-time error, 162
- Component class, 120
- Composite class, 120
- Composite pattern, 119–125
 - UML representation, 120
- CompositeSpecification class, inheriting from, 124
- ConcreteClassA class, 104
- ConcreteClassB class, 104
- ConcreteDecoratorA class, 98
- ConcreteStateA class, 109
- ConcreteStateB class, 109
- ConcreteStrategy class, 114
- concurrency control
 - optimistic, 203
 - pessimistic, 203
- ConfigureAutoMapper method, 359, 471
- ConfigureStructureMap method, 49
- ConfirmOrder.aspx view, 658–659
- constructor injection, ProductCatalogService class, 478
- containers, IoC, 288–292
- Context class, 108, 114
- controller base class, 350
- controller factories, creating, 481
- ControllerFactory class, 352
- controllers
 - account controllers, Agatha’s Clothing Store, 590–600
 - Agatha’s Clothing Store
 - checkout, 649–658
 - order, 649–658
 - payment, 649–658
 - base controllers, creating, 482
 - basket (Agatha’s), 538–558
 - CatalogBaseController, 480–481
 - CheckoutController, 652–655
 - constructors, services, 481
 - customer controllers, Agatha’s Clothing Store, 586–590
 - inheritance, 540
 - instantiation, 352
 - Product Catalog (Agatha’s), 479–488
 - routes, mapping to, 356
 - services, 481
- CookieDataKeys enumeration, 541–542
- cookies
 - authentication, 585–586
 - storage, 540–541
- CookieStorageService class, 541
- correlation ID, 160, 175
 - service response messages, 177
- Courier class, 516–517
 - mapping, 526
- CreateQueryAndObjectParameters method, 269
- CreateShippingCourier method, 97
- creational patterns, 4, 15
- CriteriaOperator enumeration, 214
- Criterion class, 215
- CRUD (create, read, update, delete) methods, 193
- CSDL (Conceptual Schema Definition Language), 262
- CurrencyPriceDecorator class, 101
- Customer class, 561–567
 - business rules, 564–565
 - inheriting from, 209
- customer controllers, 586–590

- Customer entity, converting to CustomerView, 571
 - customer membership
 - authentication service, 578–586
 - cookies, 585–586
 - Janrain, 581–585
 - customer model, 561–567
 - customer service, 570–578
 - data tables, 567
 - NHibernate mappings, 567–570
 - views, 600–603
 - CustomerAccount class, 119
 - properties, 120–121
 - CustomerBusinessRules class, 565
 - CustomerProxy class, 209–210
 - CustomerRepository class, 210
 - CustomerType class, 40
 - CustomerType enumeration, 43
 - CustomerView class, 345
 - CustomHttpHandler class, 330
- D**
- Dahan, Udi, 613
 - DAL (data access layer), 45–46
 - business logic, 194
 - Data Concurrency Control, 203–206
 - description, 193
 - Entity Framework, 224–225
 - Identity Map, 211–213
 - Lazy Loading pattern, 206–211
 - Linq to SQL, 45
 - ORM (object relational mapper), 223
 - code example, 225–283
 - Proxy pattern, 206–211
 - Query Object pattern, 213–223
 - Unit of Work pattern, 196–203
 - XML files, 193
 - DAO (Data Access Objects) pattern, 195
 - Active Record pattern, 195
 - IProductDAO interface, 195
 - Transaction Script Business pattern, 195
 - data access strategies
 - DAO (Data Access Objects) pattern, 195
 - Repository pattern, 194–195
 - Data Concurrency Control, 203–206
 - data integrity, Unit of Work pattern, 196
 - Data Mapper pattern, 9
 - data sharing, Messaging patterns, 157
 - data store, 193
 - data tables
 - customer membership, 567
 - order data tables, 627
 - Product Catalog (Agatha's), 450–451
 - databinding, jTemplates, 390–408
 - DDD (Domain-driven Design), 11, 88–89, 428
 - aggregate roots, 90
 - aggregates, 90
 - Application service, 90
 - domain services, 90
 - entities, 89
 - layering, 90
 - repositories and, 194
 - Repository pattern, 90
 - ubiquitous language, 89
 - value objects, 89
 - Decorator pattern, 98–103
 - classes extending state, 98
 - UML representation, 98–99
 - DefaultProduct class, 98
 - delivery options, 518, 529–530
 - DeliveryAddress class, 561–567
 - DeliveryAddressMapper class, 572
 - DeliveryOption class, mapping, 528
 - dependencies
 - applying, 188
 - BootStrapper class, 331
 - configuration, 291
 - Facade pattern, 156
 - IoC containers, 178
 - presenters, registering, 311–312
 - ProductService, resolving, 307
 - Repositories, 243
 - StructureMap, 511–513
 - Unit of Work implementation, 243
 - Dependency Injection principle, 22
 - discount strategy and, 39
 - Dependency Inversion principle, 20–22
 - dependent classes, creation, 94
 - design patterns
 - Alexander, Christopher, 4

design patterns (*continued*)

- applying, 17–18
- behavioral patterns, 4, 16–17
 - Chain of Responsibility, 16
 - Command, 16
 - Interpreter, 16
 - Iterator, 16
 - Mediator, 16
 - Memento, 16
 - Observer, 16
 - State, 16
 - Strategy, 16
 - Template Method, 16
 - Visitor, 16
- blueprints for solutions, 4
- Chain of Responsibility, 337–344
- creational patterns, 4, 15
- Decorator, 98–103
- enterprise (Fowler)
 - base, behavioral, and structural, 10
 - domain logic patterns, 8–9
 - layering, 8
 - object relational mapping, 9–10
 - web presentation, 10
- Facade pattern, 155–156
- Factory Design pattern, 286–287
- Factory Method, 94–98
- language agnostic, 5
- limitations, 5
- MVP (Model-View-Presenter), 292–315
- necessity, 4–5
- origins, 4
- reading, 13–15
- selecting, 17–18
- Singleton, 626
- solutions, 5
 - reuse, 5
- State pattern, 108–113, 612
- Strategy, 113–118
- structural patterns, 4, 15–16
 - adapter, 16
 - bridge, 16
 - composite, 16
 - decorator, 16
 - facade, 16
 - flyweight, 16
 - proxy, 16
 - Template method, 103–108
 - usefulness, 5
- design principles
 - DI (Dependency Injection), 128
 - DIP (Dependency Inversion Principle), 128
 - DRY (Don't Repeat Yourself), 6
 - Interface Segregation, 135–139
 - KISS (Keep It Simple Stupid), 6
 - LSP (Liskov Substitution principle), 139–148
 - SoC (Separation of Concerns), 6–7
 - S.O.L.I.D.
 - DI (Dependency Injection), 8
 - DIP (Dependency Inversion Principle), 7
 - IoC (Inversion of Control), 8
 - ISP (Interface Segregation Principle), 7
 - LSP (Liskov Substitution Principle), 7
 - OCP (Open-Closed Principle), 7
 - SRP (Single Responsibility Principle), 7
 - Tell, Don't Ask, 6
 - YAGNI (You Ain't Gonna Need It), 6
- Detail method, 488
- DI (Dependency Injection), 8, 128
 - Constructor Injection, 129, 133
 - Method Injection, 129, 134
 - Setter Injection, 129
- dictionaries, UnitOfWork class, 200
- DIP (Dependency Inversion Principle), 7, 128
 - definition, 128
 - dependent classes and, 94
- discount strategies, 115
- DiscountType class, 116–117
- Dispatch method, 95
- Display method, 302, 307
- displayComments method, 412
- DisplayDiscount method, 32
- displayEvents method, 384
- displayPage method, 510
- Display404PageCommand class, 329
- DisplaySavings method, 32
- Document Message pattern, 157–158
 - service methods, 157
- Dojo Toolkit, 372
- DOM (Document Object Model), data retrieval,
 - XMLHttpRequest object, 371

- domain entities
 - converting to ViewModels, 347–348
 - mapping to view models, 470
 - domain events, 613–615
 - domain logic patterns
 - Active Record, 8
 - Domain Model, 9
 - Transaction Script, 8
 - domain model, SOA example, 161–169
 - Domain Model pattern, 9, 36, 68–86
 - PI (persistent ignorance), 69
 - POCO (plain old common runtime object), 69
 - domain service class, 218
 - domain services, 90
 - DomainEvents class, 615
 - Dorrans, Barry, 580
 - downcasting, base classes, 144
 - drop-down list, 32–33
 - DRY (Don't Repeat Yourself), 6
 - duplication, Layer Supertype pattern and, 125
- E**
- e-mail service, Agatha's Clothing Store, 441–443
 - EFOUnitOfWork class, 265–266
 - Embedded Resource, 246
 - Product Catalog (Agatha's), 452
 - EmployeeRepository class, 212
 - enterprise design patterns (Fowler)
 - base, behavioral, and structural
 - Gateway, 10
 - Null Object, 10, 25–26
 - Separated Interface, 10
 - Composite Design pattern, 119–125
 - domain logic
 - Active Record, 8
 - Domain Model, 9
 - Transaction Script, 8
 - Layer Supertype pattern, 125–128
 - layering, 8
 - leveraging, 25–26
 - object relational mapping
 - Data Mapper, 9
 - Identity Map, 9
 - Lazy loading, 9
 - Query Object, 9–10
 - Repository, 9
 - Unit of Work, 9
 - Specification pattern, 118–119
 - Web presentation patterns, 10
 - entities, 89
 - Linq to SQL, creating, 45
 - entity classes, Layer Supertype pattern, 127
 - Entity Data Model Wizard, 257–260
 - Entity Framework, 224–225
 - business entities, 262
 - CSDL (Conceptual Schema Definition Language), 262
 - DAL automation, 223
 - DataContext, 263
 - current, 265
 - EFOUnitOfWork class, 265–266
 - MSL (Mapping Specification Language), 262
 - presentation layer, 274
 - querying, 266–267
 - repository, 257–274
 - building, 262–263
 - SSDL (Store Schema Definition Language), 262
 - storage containers, 264–265
 - Entity SQL String builder, 266–267
 - EntityBase class, 126
 - inheritance, 516
 - EntityModelCodeGenerator, 260
 - ErrorLog class, 175
 - Evans, Eric, 91
 - Event class, 161, 375
 - mehods, 165
 - EventRepository class, 178, 375–376, 377
 - events
 - checking for, 376
 - click, btnReserveTickets, 188
 - domain events, 613–615
 - Page_Init, 307
 - raising, 615
 - exceptions, custom, 620
 - expiration date, state, 158
 - expression of intent, 4
 - extension methods, 41
 - converting domain entities, 234
 - ProductService class, 102
 - ExtractFrom method, 320

F

Facade pattern, 155–156
 SOA, 156
 UML representation, 156
 Factory class, 94
 couriers and, 97
 Strategy pattern, 117
 Factory Design pattern, 286–287
 Factory Method pattern
 classes, 95
 code centralization, 94
 Factory class, 94
 UML representation, 94–95
 FaultyReturnProcess class, 106
 favorite icon requests, 512–513
 fetching, scrolling-based, 408
 filterProductsBy method, 404
 FindAll method, 46, 102, 130
 FindAllSince method, 376
 FindBy method, 206, 211
 Firebug, 388–390
 AJAX call, 413
 Firefox browser
 AJAX and, 388–390
 Firebug, 388–390
 Four Tenets of SOA, 154–155
 Fowler, Martin, 3, 206
 Fowler’s enterprise patterns, 13
 Front Controller pattern, 10
 Chain of Responsibility pattern, 337–344
 Command pattern, 315–336
 passivity, 315
 requests, 315
 FrontController class, 316, 330

G

Gang of Four, 4
 Gateway pattern, 10
 GenerateReturnTransactionFor method, 107
 GetAllProductsIn method, 18
 GetBrokenRules method, 430
 GetCategories method, 482
 GetCategoryBrandViews method, 398

GetCategoryCommand class, 323–324
 GetCategoryProductsCommand class, 324–325
 getComments methods, 412
 GetControllerInstance method, 481
 GetFeaturedProductsResponse class, 469–470
 GetHandlers method, 343
 getLatestEvents method, 383
 GetOrder method, 360
 GetOrderRequest class, 633
 GetOrderResponse class, 633
 GetProductDetailCommand class, 325
 GetProducts method, 134
 getProducts method, 510
 GetTopSellingProductsCommand class, 323
 Global.asax file, 336
 GoF pattern template, 13
 Also Known As, 14
 Applicability, 14
 Classification, 14
 Collaborations, 14
 Consequences, 14
 Implementations, 14
 Intent, 14
 Known Uses, 14
 Motivation, 14
 Participants, 14
 Pattern Name, 14
 Related Patterns, 14
 Sample Code, 14
 Structure, 14
 GoF patterns, 13
 Google Web Toolkit, 372
 GridView1_RowDataBound method, 32

H

Handle method, 337
 HandlerFactory class, 342
 handlers, ordering, 343
 Handlers folder, 338
 hash tables, sessions, 250
 HasReachedRentalThresholdSpecification class, 119
 Helm, Richard, 4
 helper classes, Agatha’s Clothing Store, 443–444
 Hibernate framework, 223

hideOverlay method, 404
 Hohpe, Gregor, 159
 HomeController, inheritance, 482
 HomeController class, 350
 inheritance, 363
 HomePagePresenter class, 297–298
 HomePageView, 479
 HQL (NHibernate), 256
 HtmlHelper class, 490
 HTTP, items collection, 455
 HTTP Context Cache API, 23

I

IAccountRepository interface, 201
 IActionCommand interface, 322
 IAggregateRoot interface, 198
 IBankAccount Repository, 75–79
 IBasket interface, 299
 IBasketDiscountStrategy interface, 115, 117
 IBasketRepository interface, 524–525
 IBasketView interface, 302–303, 310–311
 ICacheStorage interface, 23
 ICategoryProductView interface, 307
 ICriteria interface, 256
 ICustomerRepository interface, 210, 569
 ICustomerService interface, 574–576
 IDeliveryOption interface, 518
 Idempotent Messaging pattern, 159–160
 correlation ID, 180
 idempotent operations, 159
 Identity Map pattern, 9, 211–213
 NHibernate, 459
 unit tests, 213
 versions, 211
 IdentityMap class, 212
 IDiscountStrategy interface, 37
 IDomainEventHandlerFactory interface, 614
 IEmployeeRepository interface, 212
 IEventRepository interface, 165, 375
 IHandlerFactory interface, 342
 IHomePagePresenter interface, 298
 IHomeView interface, 306–307
 IHttpHandler interface, 330
 impedence mismatch, 223
 Index method, 350
 indexOfEventToAdd parameter, 384
 inheritance
 controllers, 540
 EntityBase class, 516
 HomeController, 482
 HomeController class, 363
 Layer Supertype pattern, 125–128
 persistence ignorance, 223
 response messages, 237
 response objects, 171
 Inherits keyword, 353
 intent, expression, 4
 Interface Segregation principle, 135–139, 198
 interfaces
 controller services, 481
 IAccountRepository, 201
 IActionCommand, 322
 IAggregateRoot, 198
 IBasket, 299
 IBasketDiscountStrategy, 115, 117
 IBasketRepository, 524–525
 IBasketView, 302–303, 310–311
 ICacheStorage, 23
 ICategoryProductView, 307
 ICriteria, 256
 ICustomerRepository, 210, 569
 ICustomerService, 574–576
 IDeliveryOption, 518
 IDiscountStrategy, 37
 IDomainEventHandlerFactory, 614
 IEmployeeRepository, 212
 IEventRepository, 165, 375
 IHandlerFactory, 342
 IHomePagePresenter, 298
 IHomeView, 306–307
 IHttpHandler, 330
 IOrderRepository, 358, 626
 IOrderService, 637
 IOrderState, 111, 624
 IPageNavigator, 300–301, 319
 IPaymentService, 644–645
 IPrice, 100
 IProduct, 95, 98
 IProductAttribute, 447

interfaces (*continued*)

- IProductCatalogService, 470
- IProductDetailView, 301, 309–310
- IProductDiscountStrategy, 131
- IProductListView, 46–47
- IProductRepository, 40–41, 48, 102
- IQuery, 256
- IRepository, 194
- IShippingCourier, 96
- ISpecification, 118–119
- IUnitOfWork, 198
- IUnitOfWorkRepository, 198
- IViewStorage, 317
- IWebCommand, 316
- repositories, 230–231
- translating, 22–23
- internal state, State pattern, 108
- interoperability, SOA and, 155
- InvalidAddressException class, 565
- IoC (Inversion of Control), 8
 - containers, 288
 - dependencies, 178
 - StructureMap, 288–292
 - Factory Design pattern, 286–287
 - Service Locator, 287–288
 - setup, 511–513
 - StructureMap, 427
- IoCControllerFactory, configuration, 355
- IOrderRepository interface, 358, 626
- IOrderService interface, 637
- IOrderState interface, 111, 624
- IPageNavigator interface, 300–301, 319
- IPaymentService interface, 644–645
- IPN (Instant Payment Notification), 642
- IPrice interface, 100
- IProduct interface, 95, 98
 - implementing, 136
- IProductAttribute interface, 447
- IProductCatalogService interface, 470
- IProductDAO interface, 195
- IProductDetailView interface, 301, 309–310
- IProductDiscountStrategy interface, 131
- IProductListView interface, 46–47
- IProductRepository interface, 40–41, 48, 102

- IQuery interface, 256
- IRepository interface, 194
- IShippingCourier interface, 96
- IsNameQuery method, 217
- ISP (Interface Segregation Principle), 7
- ISpecification interface, 118–119
- IsSatisfied method, 124
- ITicketService class, 173
- IUnitOfWork interface, 198
- IUnitOfWorkRepository interface, 198
- IViewStorage interface, 317
- IWebCommand interface, 316

J

- Janrain, 581–585
- JavaScript
 - Dojo Toolkit, 372
 - Google Web Toolkit, 372
 - jTemplates, databinding, 390–408
 - libraries, 372
 - jQuery, 372
 - Script.aculo.us, 372
- Johnson, Ralph, 4
- jQuery, 372
 - Scripts folder, 378
- jQuery, visual Studio 2010, 372
- JSON (JavaScript Object Notation)
 - AJAX and, 371
 - events, 384
- json2.js file, 495
- JsonModelBinder classes, 483–484
- JsonProduct class, 484
- JsonRefinementGroup class, 484
- jTemplates, databinding, 390–408

K

- Katz, Yehunda, 372
- keywords
 - Inherits, 353
 - this, 307
- KISS (Keep It Simple, Stupid), 6
- Kuate, Pierre, 224

L

LateFees property, 122

Layer Supertype pattern, 125–128

- entity classes, 127
- inheritance, 125

layering applications

- business layer, 36–42
- data access layer, 45–46
- presentation layer, 46–47
- Service layer, 42–45
- SoC principles, 30
- user experience layer, 48–51

layering design patterns, 8

- DDD (Domain-driven Design), 90

Lazy Loading pattern, 9, 206–211

- Proxy pattern and, 208–210

Leaf class, 120

Lerman, Julia, 225

leveraging enterprise patterns, 25–26

libraries

- Facade pattern, 156
- JavaScript, 372
 - jQuery, 372

library database schema, 244–245

library domain model

- business rules, 230
- classes, 227
- validation checking, 230

LibraryService class, 240–243

LINQ (Language Integrated Query), repositories and, 194

Linq to SQL, 45

- DAL automation, 223
- entities, creating, 45

LinqProductRepository, 129, 132–133

loading, objects, Identity Map pattern, 211

Loan class, 229

local authentication, ASP.NET membership, 579–580

logging mechanism, Agatha’s Clothing Store, 439–441

LSP (Liskov Substitution Principle), 7, 139–148

- base classes and, 139
- subclasses and, 139

M

magic strings, avoiding, 541

Mappers folder, 234

mapping

- AutoMapper, 345
- domain entities to view models, 470
- Entity Framework, 224–225
- NHibernate
 - Agatha’s, 525–529
 - order mappings, 627–630
 - Product Catalog (Agatha’s), 453–454
 - routes to controllers, 356
 - ViewModels with AutoMapper, 356–361

Mapping folder, 346

MappingFiles folder, 245

master pages, 305–306

- customer membership views, 600

MasterPage, display controls, populating, 307

McCarthy, Tim, 196

Member class, 228

- methods, 229

MemberRepository class, 254–255

message objects, 237

Message property, 43, 237

messages, service interfaces, 155

Messages folder, 237

Messaging patterns, 157

- data sharing, 157
 - Document Message pattern, 157–158
 - Idempotent Messaging pattern, 159–160
 - Request-Response pattern, 157–158
 - Reservation Messaging pattern, 158–159

method, getComments, 412

methods

- Add, 206
- AdjustPriceWith, 129
- ApplyExtraDiscountsTo, 33
- CalculateRefundFor, 106
- Commit, 200
- ConfigureAutoMapper, 359, 471
- ConfigureStructureMap, 49
- CreateQueryAndObjectParameters, 269
- CreateShippingCourier, 97
- Detail, 488

methods (*continued*)

- Dispatch, 95
- Display, 302, 307
- displayComments, 412
- DisplayDiscount, 32
- displayEvents, 384
- displayPage, 510
- DisplaySavings, 32
- Event class, 165
- extension, 41
- ExtractFrom, 320
- filterProductsBy, 404
- FindAll, 46, 102, 130
- FindAllSince, 376
- FindBy, 206, 211
- GenerateReturnTransactionFor, 107
- GetAllProductsIn, 18
- GetBrokenRules, 430
- GetCategories, 482
- GetCategoryBrandViews, 398
- GetControllerInstance, 481
- GetHandlers, 343
- getLatestEvents, 383
- GetOrder, 360
- GetProducts, 134
- getProducts, 510
- GridView1_RowDataBound, 32
- Handle, 337
- hideOverlay, 404
- Index, 350
- IsNameQuery, 217
- IsSatisfied, 124
- Member class, 229
- OrderExtensionMethods, 359
- pauseUpdates, 386
- ProcessRequest, 330
- refineSearch, 510
- Resolve, 287
- Resolve, 490
- serviceSuccessful, 384
- SessionFactory class, 251–252
- SetNextHandler, 337
- showOverlay, 404
- storeComments, 412
- Submit, 625

- ToString, 290
- TranslateInto, 221
- Model-View-Controller pattern, 10
- Model-View-Presenter pattern, 10
 - presentation logic and, 46
- ModelBinder, 483
- ModelRegistry class, 291
- ModifyBasketRequest class, 533
- ModifyBasketResponse class, 533
- modules, dependencies, 128
- MonoRail framework, 361–368
- MSL (Mapping Specification Language), 262
- multiple users, Identity Map pattern, 211
- MVC (Model-View-Controller) pattern, 344–345
 - ASP.NET MVC framework, 345–361
 - HomeController, 398
 - ModelBinder, 483
 - ProductController, 401
 - ViewModel pattern, 345
- MVP (Model-View-Presenter) pattern, 292–293
 - ASP.NET Web forms, 293–315
 - classes, 314
 - updated diagram, 314

N

- named queries, Query Object, creating, 217
- namespaces, SoC and, 35
- NameValueCollection, query arguments, 321
- Navigator, 316
- .NET Framework, patterns, identifying, 17
- NHibernate, 223–224
 - Agatha’s Clothing Store, 427
 - aliases, 463
 - HQL, 256
 - Identity Map pattern, 459
 - interfaces
 - ICriteria, 256
 - IQuery, 256
 - mapping
 - Agatha’s, 525–529
 - customer membership, 567–570
 - order mappings, 627–630
 - persistance ignorance, 223
 - presentation layer, 274

- Product Catalog repositories, 451–464
 - repository, 245–257
 - sessions, 248–250, 454
 - Unit of Work pattern, 252–253, 458–459
 - XML configuration file, 223–224, 451
- NHibernate framework, XML files, 245
- NHibernate Repository, programming, 454–462
- Nilsson, Jimmy, 91
- NoBasketDiscount class, 116
- NotSpecification class, 122–123
- Null Object pattern, 10, 25–26, 519
 - NullDiscountStrategy class, 38
 - Strategy pattern and, 116
- NullDeliveryOption class, 519
- NullDiscountStrategy class, 38
- NullObjectCache class, 25–26
- NVelocity, 361–362
- NWUnitOfWork class, 252–253, 458

O

- object-oriented programming, design patterns and, 4
- object relational mapping patterns
 - Data Mapper, 9
 - Identity Map, 9
 - Lazy Loading, 9
 - Query Object, 9–10
 - Repository, 9
 - Unit of Work, 9
- objects
 - chaining, 337
 - as single object, 119–125
- OCP (Open-Closed Principle), 7
- online chat, Periodic Refresh pattern, 374
- optimistic concurrency, 203
- Order Class, 620–621
- Order class, 110–111
- order controllers, 649–658
- order data tables, 627
- order model, 611–626
- order NHibernate mappings, 627–630
- order service, 631–640
- order view, 658–668
- OrderByClause class, 215
- OrderController class, 651
- OrderExtensionMethods method, 359

- OrderItem class, 617
- OrderItemView class, 631
- OrderNewState class, 112
- OrderQueryTranslator class, 219
- OrderRepository, 358
- OrderRepository class, 209
- orders, canceled, 111
- OrderService class, 95
 - queries, 219
- OrderServiceFactory class, 286
- OrderShippedState class, 112
- OrderState class, 624–625
- OrderStatus enumeration, 110
- OrderSummaryView class, 631
- ORM (object relational mapper), 223
 - code example
 - database, 244–245
 - Entity Framework repository, 257–274
 - library domain model, 225–233
 - NHibernate repository, 245–257
 - presentation layer, 274–283
 - raw ADO.NET repository, 283
 - service layer, 233–244
 - impedance mismatch, 223

P

- Page Controller pattern, 10, 368
- page layout, master pages, 305–306
- Page_Init event, 307
- partial views, 353
 - Shared folder, 365
- pattern, request-response messaging, 572
- pattern templates, 13
- patterns. *See also* specific pattern types
 - AJAX, 372
 - Periodic Refresh, 372–390
 - Timeout, 374
 - Messaging patterns, 157
 - MVC (Model-View-Controller), 344–368
 - Predictive Fetch, 408–413
 - Request/Response messaging pattern, 43
 - studying, 17
 - UML diagrams, 15
 - ViewModel, 345

- pauseUpdates method, 386
- payment
 - controllers, 649–658
 - merchant, 611
 - PayPal, 640–648
- PaymentController class, 656–657
- PaymentFactory class, 617
- PaymentServiceBase class, 141
- PaymentType enumeration, 140
- PayPal, 611, 640–648
- Periodic Refresh pattern, 372–390
 - implementing, 377–378, 383
 - script, 383–386
 - Timeout pattern, 374
- persistence
 - atomic transactions, 200
 - enabling, IAccountRepository interface, 201
 - ignorance, 223
 - Unit of Work pattern, 196
- Person repository, 204
- pessimistic concurrency, 203
- PI (persistent ignorance), 69
 - objects, 194
- POCO (plain old common runtime object), 69, 194
- Predictive Fetch pattern, 408–413
- presentation layer, 46–47
 - Entity Framework, 274
 - NHibernate, 274
- Presentation models, 42
 - classes, 184–186
- presenter class, responsibilities, 47
- presenters, dependencies, registering, 311–312
- Price class, 36
- principles
 - Dependency Injection, 22
 - Dependency Inversion, 20–22
 - refactoring to, 20–22
- ProcessRequest method, 330
- Product Catalog (Agatha's)
 - categories, 447–448
 - controllers, 479–488
 - data tables, 450–451
 - product catalog model, 446–450
 - product detail, 480
 - product services, 464–478
 - product views, 465–467
 - ProductCatalogService, 467–478
 - repositories, 451–464
 - Embedded Resource, 452
 - mapping files, 453–454
 - views, 489–511
- Product class, 36
 - decorating behavior, 101
- Product entity, converting to
 - ProductViewModel, 43
 - ViewModels, 348
- product retrieval, 40
- ProductCatalogService, operations, 467–468
- ProductCatalogService class, constructor injection, 478
- ProductCollectionExtensionMethods class, 101–102
- ProductColor class, 448
- ProductController class, 351, 363–364, 401, 485–488
- ProductDecorator class, 98
- ProductDetailView, 346
- ProductList.aspx user control, 304–305
- ProductListExtensionMethods class, 41
- ProductListPresenter class, 47
- ProductListRequest class, 43
- ProductMapperExtensionMethods class, 43
- ProductRepository class, 18, 45
- ProductService class, 18, 41, 44, 102–103, 129–130
 - dependencies, resolving, 307
- ProductSize class, 448
- ProductSummaryView product view, 466
- ProductTitle class, 449–450
- ProductView, 346
- ProductView product view, 466
- ProductViewModel class, 42
 - converting Product entity, 43
- properties
 - AccountActive, 122
 - AmountToRefund, 106
 - Certification, 137
 - CustomerAccount class, 120
 - LateFees, 122
 - Message, 43, 237
 - Response object, 43
 - RunningTime, 137

proxies

- client proxy, 182–186
- remote proxy, 207
- virtual proxy, 207

Proxy pattern, 206–211

- UML representation, 207

Q

queries

- Entity Framework, 266–267
- generating, 478
- NameValueCollection, 321
- OrderService class, 219
- Query Object pattern, 213
 - named queries, 217

Query class, 216

Query Object pattern, 9–10

- converting to SQL command, 219
- model, 214
- queries, named, 217
- repositories, 194

query object pattern, 433–438

query objects, 433

QueryName class, 216

QueryOperator enumeration, 215

QueryTranslator, 214, 217

QueryTranslator class, 254, 255–256

R

raw ADO.NET repository, 283

reading design patterns, 13–15

red-green-refactor, 11

refactoring

- to adapter pattern, 22–25
- design patterns and, 4
- to principles, 20–22

references

- services, remote proxy, 207
- StructureMap.dll, 290–291

refinements, 510

refineSearch method, 510

RefundRequest class, 140

RefundResponse class, 140

RefundService class, 139

registering dependencies, presenters, 311–312

registering services, 287

remote proxy, 207

reply classes, 532

repositories

DDD, 194

Entity Framework, 257–274

building, 262–263

implementations, 209

in-memory collection, 194

interfaces, 194, 230–231

LinqProductRepository, 129, 132–133

NHibernate, 245–257

Person, 204

Product Catalog (Agatha's), 451–464

Query Object pattern, 194

querying, 213

raw ADO.NET, 283

Repository class, 253–254, 461

repository implements, 462

Repository pattern, 9, 40–41, 194–195

DDD (Domain-driven Design), 90

implementations, 253–254

request classes, 532

request messages, 243

Request-Response Messaging pattern, 43, 157–158, 572, 633–634

RequestHandler class, 337, 338

requests

Command pattern, 315

Front Controller, 315

Reservation Messaging pattern, 158–159

Resolve method, 287, 490

resource-intensive objects, 207

response messages, inheritance, 237

response objects

inheritance, 171

properties, 43

ResponseBase class, 237

retrieving customers

deferring execution, 206–211

Repository contract, 209

retrieving products, 40

ReturnAction class, 104–105

- ReturnOrder class, 105
- ReturnProcessFactory class, 107
- ReturnProcessTemplate class, 105–106
- Route class, 322
 - WebRequest, 339
- routes, mapping to controllers, 356
- Ruby on Rails framework, 58
- RunningTime property, 137

S

- Script.aculo.us, 372
- Scripts folder, jQuery, 378
- scrolling-based fetching, 408
- selection criteria logic, sharing, 118
- Separated Interface pattern, 10
- separating concerns, 29
- serialization, enabling, 172
- Service Bus, 153
- Service class, 104, 107
- service compatibility, 155
- service facade, building, 184
- service interfaces, messages and, 155
- service layer, 42–45
 - database, 181
 - description, 151
 - facade, 42
 - Facade pattern, 156
 - granularity, 151
 - ORM (object relational mapper), 233–244
 - presentation model, 42
 - SOA example, projects, 170
 - Transaction Script and, 151
- Service Locator, 287–288
- service methods
 - autonomy, 155
 - Document Message pattern, 157
 - state, 155
- service response messages, correlation identifiers, 177
- services
 - references, remote proxy, 207
 - registering, 287
- Services project, ProductMapperExtensionMethods class, 43
- serviceSuccessful method, 384
- session containers, 456
- SessionFactory class, 250–252, 457–458
 - methods, 251–252
- sessions
 - HTTP items collection, 455
 - NHibernate, 248–250, 454
- SetNextHandler method, 337
- SetOrderPaymentRequest class, 633
- SetOrderPaymentResponse class, 633
- Shared folder, partial view, 353
- shipping courier, 517
- ShippingService class, 517–518
 - mapping, 526
- ShopService class, 349–350
- showOverlay method, 404
- simplified template
 - code example, 15
 - Name and Intent section, 14
 - UML diagram, 15
- Singleton pattern, 626
- Site.Master file, 492–495
- skeleton process workflow, 104
- skeleton template method, 104
- Smart UI antipattern, 29, 30–35
- Smart UI applications, benefits, 30
- SOA (service oriented architecture), 152–154
 - applying, example, 152–154
 - benefits, 153–154
 - example, 160–161
 - client, 186–190
 - client proxy, 182–186
 - domain model, 161–169
 - service layer, 169–182
 - Facade pattern, 156
 - Four Tenets, 154–155
 - interoperability and, 155
 - services, 152
- SoC (Separation of Concerns), 6–7
 - layering and, 30, 35
 - namespaces and, 35
- social networking, Periodic Refresh pattern, 374
- S.O.L.I.D. design principles, 13
- solutions, design patterns, 5
- Special Case pattern, 25–26
- Specification pattern, 118–119

SQL, commands, Query Object pattern, 219
 SQL (Structured Query Language), HQL (NHibernate), 256
 SRP (Single Responsibility Principle), 7
 SSDL (Store Schema Definition Language), 262
 state
 expiration date, 158
 service methods, 155
 state-altering requests, Idempotent Messaging pattern, 159–160
 State class, 109
 State pattern, 108–113, 612
 case statements, 108
 classes, 109
 internal state, 108
 UML representation, 108–109
 storeComments method, 412
 Strategy class, 114
 Strategy pattern, 37, 113–118
 Factory class, 117
 UML representation, 114
 structural patterns, 4, 15–16
 StructureMap, 288–292, 427
 configuration, 331
 dependencies, 511–513
 downloading, 48
 studying patterns, 17
 subclasses
 Interface Segregation, 135
 LSP and, 139
 Template method pattern, 103
 Submit method, 625
 SubSystemA class, 156
 SubSystemB class, 156
 subtypes, substituting base types, 144
 System.Transactions, 200

T

TDD (Test-driven Development), 11
 Tell, Don't Ask design principle, 6
 Template Method pattern, 103–108
 abstract, 105
 algorithm skeleton structure, 103
 creating, 105–106

skeleton, 104
 subclasses, 103, 104, 106
 UML representation, 103–104
 templates
 pattern templates, 13
 simplified, 14–15
 this keyword, 307
 TicketReservation class, 161
 TicketService class, 175
 Timeout pattern, 374
 implementing, 377–378, 383
 script, 386–388
 ToString method, 290
 TradeDiscountPriceDecorator class, 100–101
 TradeDiscountStrategy class, 37–38
 Transaction Script pattern, 8
 problems with, 56
 small applications, 56
 understandability, 56
 transactions, Identity Map pattern, 211
 TransactionScope class, 200
 TransactionView, 79
 TranslateInto method, 221

U

ubiquitous language, 89
 UI (user interface), business logic, 32
 UKShippingCourierFactory class, 97
 UML diagram, 15
 Chain of Responsibility pattern, 337
 Command pattern, 316
 Composite pattern, 120
 Decorator pattern, 98–99
 Facade pattern, 156
 Factory Method pattern, 94–95
 Proxy pattern, 207
 State pattern, 108–109
 Strategy pattern, 114
 Template method pattern, 103–104
 Unit of Work pattern, 9, 196–203
 Agatha's Clothing Store, 432–433
 data integrity and, 196
 delegating to, 202
 IAggregateRoot interface, 198

Unit of Work pattern (*continued*)

- NHibernate, 252–253, 458–459
 - persistence, 196
- UnitOfWork class, dictionaries, 200
- URLs (Uniform Resource Locators), unique, AJAX and, 390
- user controls
 - CategoryList.ascx, 304
 - ProductList.ascx, 304–305
- user experience layer, 48–51
- user inactivity, 387
- users
 - bookmarks, 405
 - multiple, Identity Map pattern, 211

V

- validation, baskets, 520
- value objects, 89
- ValueObjectBase class, 431
- versions
 - concurrency checking, 204–206
 - Identity Map pattern, 211
- view classes, inheriting from, 346–347
- view model, 42
- ViewDataDictionary, 351
- ViewModel pattern, 345
 - converting domain entities, 347–348
 - mapping, AutoMapper, 356–361
- ViewResult base class, 350
- views
 - authentication, Agatha’s Clothing Store, 604–609
 - basket (Agatha’s), 538–558
 - CategoryView, 346
 - checkout, 658–668
 - Checkout.master, 658
 - ConfirmOrder.aspx, 658–659
 - creating, 353–355
 - customer membership, 600–603
 - order, 658–668
 - partial, 353, 365
 - Product Catalog (Agatha’s), 489–511
 - ProductDetailView, 346
 - ProductView, 346

- Views folder, 346
- ViewStorage, populating, 323
- ViewStorage class, 318
- ViewStorageFactory class, 318, 333
- ViewStorageKeys enumeration, 318
- virtual proxy, 207
- Visual Studio 2010, jQuery, 372
- Vlissides, John, 4

W

- WCF Service application, projects, 181
- web-based authentication, 581–585
- Web forms, MVP pattern and, 293–315
- Web presentation patterns, 10
- web services, creating, 376–377
- WebBasket class, 300
- WebCommand class, 316, 326–327
 - route, 327
- WebCommandRegistry, 316, 327–329
 - actions, 329
- WebRequest class, 320
- WebUI project, BootStrapper class, 48
- wizards, Entity Data Model, 257–260
- Wolf, Bobby, 159
- workflow, skeleton process, 104
- WS-Policy, specifications, 155
- wsHttpBinding, 181

X

- XML configuration file, NHibernate and, 223–224
- XML (eXtensible Markup Language)
 - AJAX and, 371
 - NHibernate and, 451
- XML files
 - DAL and, 193
 - NHibernate framework, 245
- XMLHttpRequest object, 371

Y

- YAGNI (You Ain’t Gonna Need It), 6