

Применение DDD и шаблонов проектирования

Проблемно-ориентированное
проектирование приложений
с примерами на C# и .NET

Applying Domain-Driven Design and Patterns

with Examples in C# and .NET

Jimmy Nilsson



Addison-Wesley

Применение DDD и шаблонов проектирования

Проблемно-ориентированное
проектирование приложений
с примерами на C# и .NET

Джимми Нильссон



Москва • Санкт-Петербург • Киев
2008

ББК 32.973.26-018.2.75
Н66
УДК 681.3.07

Издательский дом “Вильямс”

Зав. редакцией *С.Н. Тригуб*

Перевод с английского и редакция *И.В. Берштейна*

По общим вопросам обращайтесь в Издательский дом “Вильямс” по адресу:
info@williamspublishing.com, <http://www.williamspublishing.com>
115419, Москва, а/я 783; 03150, Киев, а/я 152

Нильссон, Джимми.

Н66 Применение DDD и шаблонов проектирования: проблемно-ориентированное проектирование приложений с примерами на C# и .NET. : Пер. с англ. — М. : ООО “И.Д. Вильямс”, 2008. — 560 с. : ил. — Парал. тит. англ.

ISBN 978-5-8459-1296-1 (рус.)

Эта книга о разработке корпоративных программных приложений в среде .NET с применением шаблонов проектирования. В ней описаны: проблемно-ориентированные методы проектирования (DDD, или Domain Driven Design), разработка посредством тестирования (TDD, или Test-Driven Development), объектно-реляционное преобразование, т.е. методы, которые многие относят к ключевым технологиям разработки программного обеспечения. По мере развития и усложнения технологий все большее значение приобретают вопросы правильного применения методов проектирования. Ценность этой книги в том и состоит, что она помогает разобраться в этих вопросах. Хотя большинство примеров кода представлено на языке C#, материал книги может оказаться полезным и для тех, кто работает на платформе Java.

Книга адресована опытным разработчикам архитектуры и прикладного программного обеспечения уровня предприятий, в том числе и в среде .NET.

ББК 32.973.26-018.2.75

Все названия программных продуктов являются зарегистрированными торговыми марками соответствующих фирм.

Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотокопирование и запись на магнитный носитель, если на это нет письменного разрешения издательства Addison-Wesley Publishing Company, Inc.

Authorized translation from the English language edition published by Addison-Wesley Publishing Company, Inc, Copyright © 2006 by Pearson Education, Inc.

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means—electronic, mechanical, photocopying, recording, or otherwise—without written permission from the publisher, except for the inclusion of brief quotations in a review.

Russian language edition published by Williams Publishing House according to the Agreement with R&I Enterprises International, Copyright © 2008

ISBN 978-5-8459-1296-1 (рус.)
ISBN 0-321-26820-2 (англ.)

© Издательский дом “Вильямс”, 2008
© Pearson Education, Inc., 2006

Оглавление

Предисловия	27
Вступление: наведение мостов	31
Часть I. Основы	39
Глава 1. Переоценка ценностей	41
Глава 2. Переход к шаблонам	85
Глава 3. РПТ и рефакторинг кода	115
Часть II. Прикладное ППО	149
Глава 4. Новая используемая по умолчанию архитектура	151
Глава 5. Проектирование на основе модели предметной области	181
Глава 6. Подготовка к инфраструктуре	221
Глава 7. Порядок устанавливаются правила	271
Часть III. Применение шаблонов архитектуры корпоративных программных приложений	319
Глава 8. Инфраструктура для сохраняемости	321
Глава 9. Приведение в действие преобразователя NHibernate	353
Часть IV. Что дальше	389
Глава 10. Рекомендуемые методы проектирования	391
Глава 11. Акцент на пользовательском интерфейсе	449
Часть V. Приложения	489
Приложение А. Другие виды моделей предметной области	491
Приложение Б. Перечень рассмотренных шаблонов	529
Список литературы	536
Предметный указатель	543

Содержание

Отзывы о книге	17
Об авторе	26
Предисловия	27
Вступление: наведение мостов	31
Благодарности	37
От издательства	38
Часть I. Основы	39
Глава 1. Переоценка ценностей	41
Общие ценности	41
Оценка стилей архитектуры	42
Акцент на модели	42
Акцент на прецедентах использования	43
Если сделан акцент на модели, следует применять шаблон модели предметной области	48
База данных требует аккуратного обращения	53
Потеря соответствия модели предметной области и реляционной базы данных	58
Распределение требует аккуратного обращения	63
Акцент на обмене сообщениями	65
Оценка составляющих процесса	67
Предварительное проектирование архитектуры	68
Проектирование на основе предметной области	71
Разработка посредством тестирования	71
Рефакторинг кода	75
Отдельные составляющие или их сочетание	77
Непрерывная интеграция	78
Проблема интеграции	79
Решение проблемы или радикальный шаг в правильном направлении	79
Уроки, которые я извлек из опыта работы в своей организации	80

Дополнительные сведения	81
Не следует забывать об эксплуатации	81
Когда механизм нужен	81
Некоторые примеры механизмов эксплуатации	82
Это не только оплошность разработчиков	83
Резюме	84
Глава 2. Переход к шаблонам	85
Вкратце о шаблонах	86
Предостережения в отношении шаблонов	88
Шаблоны проектирования	90
Пример применения шаблона State	90
Архитектурные шаблоны	98
Пример применения шаблона Layers	99
Шаблон Domain Model в качестве другого примера	100
Шаблоны проектирования для конкретных типов приложений	100
Пример применения шаблона Query Object	101
Шаблоны предметной области	107
Пример применения шаблона Factory	109
Резюме	114
Глава 3. РПТ и рефакторинг кода	115
Разработка посредством тестирования	115
Порядок выполнения РПТ	116
Пора приступать к демонстрации	116
Следствия для проектирования	123
Недостатки проектирования методом РПТ	126
Следующая стадия	127
Заглушки и имитации	128
Типичный блочный тест	128
Объявление независимости	129
Работа с трудными объектами	130
Замена взаимодействующих объектов тестовыми заглушками	130
Замена взаимодействующих объектов имитирующими объектами	133
Последствия для проектирования	135
Выводы	136
Дополнительные сведения	136

Рефакторинг кода	136
Чистка недоброкачественного кода	137
Резюме	148

Часть II. Прикладное ППО **149**

Глава 4. Новая используемая по умолчанию архитектура **151**

Основа новой используемой по умолчанию архитектуры	151
Смещение акцента с базы данных на модель предметной области	153
Более конкретный акцент на ППО	153
Разделение на слои в соответствии с ППО	154
Первый набросок	155
Задачи и возможности для примера предметной области	156
Решение задач по очереди	158
Модель предметной области на данном этапе	171
Первая попытка привязать пользовательский интерфейс к модели предметной области	173
Основная цель	173
Акцент на простой пользовательский интерфейс	173
Составление списка заказов клиента	173
Ввод заказа	175
Что получается в итоге	175
Еще одно измерение	176
Расположение модели предметной области	178
Изолирование или совместное использование экземпляров	178
Реализация модели предметной области с сохранением или без сохранения состояния	179
Полная или частичная реализация модели предметной области	179
Резюме	180

Глава 5. Проектирование на основе модели предметной области **181**

Уточнение модели предметной области простым экспериментированием с РПТ	181
Первая задача: создание классов Order и OrderFactory	182
Немного логики для предметной области	187
Вторая задача: создание класса OrderRepository и ввод свойства OrderNumber	188

Восстановление объекта-сущности из состояния сохраняемости: задание значений извне	193
Выборка списка заказов	198
Пора обсудить объекты-сущности	200
Назад к основному ходу рассуждений	200
Общая перспектива	202
Подделка класса <code>OrderRepository</code>	203
Несколько слов о сохранении	205
Общая сумма каждого заказа	206
Сведения о предыстории клиента	211
Жизненный цикл экземпляра объекта	214
Тип заказа	216
Ответственное лицо для заказа	216
Плавный интерфейс	218
Резюме	220

Глава 6. Подготовка к инфраструктуре 221

РОСО как образ жизни	222
НС для объектов-сущностей и объектов-значений	223
Соблюдать НС или не соблюдать	228
НС во время выполнения или компиляции	228
Во что обходится НС для объектов-сущностей и объектов-значений	229
НС для хранилищ	231
Во что обходятся хранилища единственного ряда	237
Сценарии сохранения	238
Основания для принятых решений	240
Построение механизма подделки	243
Дополнительные свойства механизма подделки	244
Реализация подделки	246
Воздействие подделки на блочные тесты	247
Тестирование базы данных	251
Установка базы данных в исходное состояние перед каждым тестом	253
Сохранение состояния базы данных во время выполнения тестов	255
Сброс данных для теста перед его выполнением	255
Не забывайте о развитии схемы базы данных	256

Тестирование блока кода отдельно от проверки обращения к базе данных	257
Формирование запросов	261
Объекты запросов единственного ряда	262
Во что обходятся объекты запросов единственного ряда	263
Размещение запросов	266
Агрегаты как инструмент формирования запросов	267
Описания в качестве запросов	269
Другие методы формирования запросов	269
Резюме	270

Глава 7. Порядок устанавливаются правила **271**

Разделение правил на категории	271
Принципы для правил и их применения	272
Двухсторонняя проверка правил: необязательная (профилактическая) и обязательная (задним числом)	272
Сохранению подлежат все состояния — даже ошибочные	272
Правила должны применяться продуктивно	273
Правила должны допускать поправки для ввода специальных правил	273
Правила должны размещаться вместе с состоянием	273
Правила должны быть очень удобными для тестирования	273
Система должна препятствовать переходу в недействительное состояние	274
Начало создания интерфейса API	274
Контекст, контекст и еще раз контекст!	275
Ограничения, связанные с базой данных	276
Привязка правил к переходам, связанным с моделью предметной области или же с инфраструктурой	277
Уточнение принципа “Сохранению подлежат все состояния — даже ошибочные”	278
Требования к интерфейсу API для основных правил, связанных с сохраняемостью	280
Возврат к проблемам интерфейса API	281
Какая именно проблема возникла	281
Мы допустили неправильный переход	282
Если мы забыли выполнить проверку	282
Акцент на правилах, связанных с предметной областью	283

Правила, требующие взаимодействия	285
Поиск методов обработки на основе множеств	286
Обслуживание проверки достоверности по шаблону Service	289
Попытка сделать непредусмотренный переход	289
Идентификационный номер клиента	292
Как избежать осложнений	294
Агрегаты как инструмент проверки правил	295
Расширение интерфейса API	296
Запрос правил для настройки пользовательского интерфейса	296
Возможность для внесения правил	297
Уточнение реализации	297
Наивная реализация	297
Переход от самой наивной стадии к созданию классов правил	303
Составление списка правил	306
Использование списка правил	307
Составление подсписков	308
Усовершенствование интерфейса API	309
Специальная настройка	310
Предоставление пользователю метаданных	310
Проблема подстать шаблону	311
Как обращаться со сложными правилами	311
Привязка к абстрактному представлению сохраняемости	313
Внедряемый интерфейс проверки достоверности	313
Альтернативные решения для проверки достоверности задним числом при сохранении	314
Повторное использование метаданных преобразования	315
Обобщения и анонимные методы как спасение	315
Другие подходы	317
Резюме	317

Часть III. Применение шаблонов архитектуры корпоративных программных приложений 319

Глава 8. Инфраструктура для сохраняемости 321	321
Требования к инфраструктуре сохраняемости	322
Место для хранения данных	324
Оперативная память	324
Файловая система	327

12 СОДЕРЖАНИЕ

Объектная база данных	327
Реляционные базы данных	328
Один или несколько администраторов ресурсов	329
Другие факторы	329
Выбор решения и переход к его воплощению	330
Подход	330
Написание вручную заказного кода	330
Генерация заказного кода	331
Преобразование метаданных (объектно-реляционное преобразование)	333
Повторный выбор	335
Классификация	335
Вид модели предметной области	336
Вид объектно-реляционного преобразователя	336
Отправная точка	337
Акцент на интерфейсе API	338
Тип языка запросов	339
Улучшенная поддержка базы данных	340
Другие функциональные возможности	341
Классификация по шаблонам инфраструктуры	342
Тип метаданных для преобразования	343
Поля идентификации	344
Преобразование внешних ключей	346
Внедренное значение	346
Решения проблемы наследования	347
Карта идентичности	348
Единица работы	349
Загрузка по требованию или энергичная загрузка	349
Управление параллельным выполнением операций	350
Резюме	351
Глава 9. Приведение в действие преобразователя NHibernate	353
Основания для выбора преобразователя NHibernate	353
Краткое введение в NHibernate	354
Подготовка	354
Метаданные для преобразования	356
Небольшой пример интерфейса API	362
Транзакции	365

Требования к инфраструктуре сохраняемости	366
Высокий уровень неведения сохраняемости	366
Некоторые желательные свойства жизненного цикла постоянных объектов-сущностей	367
Аккуратное обращение с реляционной базой данных	369
Классификация	370
Вид модели предметной области	371
Вид объектно-реляционного преобразователя	371
Отправная точка	372
Акцент на интерфейсе API	372
Тип языка запросов	372
Улучшенная поддержка базы данных	374
Другие функциональные возможности	375
Классификация по шаблонам инфраструктуры	377
Тип метаданных для преобразования	377
Поля идентификации	377
Преобразование внешних ключей	379
Внедренное значение	380
Решения проблемы наследования	381
Карта идентичности	382
Единица работы	383
Загрузка по требованию или энергичная загрузка	383
Управление параллельным выполнением операций	384
Дополнительные возможности: перехватчики проверки достоверности	384
NHibernate и ППО	385
Краткий обзор сборок	385
Интерфейс ISession и хранилища	386
Интерфейс ISession , хранилища и транзакции	386
Чего мы достигли	387
Резюме	388
Часть IV. Что дальше	389
Глава 10. Рекомендуемые методы проектирования	391
Все дело в контексте	392
Слои и разделы	392
Основания для разбиения на разделы	393
Ограниченный контекст	393

Взаимосвязь между ограниченными контекстами и разделами	394
Расширение масштабов проектов ППО	395
Основание для разбиения модели предметной области на разделы — ориентация на службы	395
Введение в АОС	396
Что собой представляет АОС	396
Назначение АОС	396
Отличия АОС	397
Что такое служба	397
Что входит в службу	398
О значении четырех руководящих принципов АОС	399
Еще раз о том, что такое служба	400
Место объектной ориентации в АОС	401
Архитектура “клиент-сервер” и АОС	401
Односторонний асинхронный обмен сообщениями	402
Как АОС улучшает масштабируемость	403
Структура службы АОС	403
Взаимодействие службы с другими службами	404
АОС и недоступные службы	407
Сложные процессы обмена сообщениями	408
Масштабирование служб	408
Выводы	409
Инверсия управления и внесение зависимостей	410
Ни один из объектов не изолирован	410
Фабрики, реестры и указатели служб	412
Внесение зависимостей в конструктор	415
Внесение зависимостей в механизм установки	418
Инверсия управления	419
Внесение зависимостей в среде Spring.NET	420
Автосвязывание в среде PicoContainer.NET	422
Вложенные контейнеры	424
Сравнение шаблонов Service Locator и Dependency Injection	426
Выводы	427
Аспектно-ориентированное программирование	429
Из-за чего столько шума	429
Терминология АОП	433
АОП в .NET	433
Выводы	447
Резюме	447

Глава 11. Акцент на пользовательском интерфейсе	449
Предварительный эпилог	449
Шаблон Model-View-Controller	451
Пример: обувной магазин Джо	452
Упрощение интерфейсов представления с помощью адаптеров	459
Развязка контроллера от представления	459
Сочетание представлений и контроллеров	460
Чего все это стоит	460
Разработка Web-формы посредством тестирования	461
Предпосылки	461
Пример	462
Модель предметной области	463
РПТ графического пользовательского интерфейса	463
Реализация Web-формы	470
Выводы	472
Имитация средствами NMock	472
Преобразование и заключение в оболочку	474
Преобразование и заключение объектов в оболочку	475
Заключение МПО в оболочку МП	476
Преобразование МП в МПО	477
Организация отношений между объектами	480
Вопросы состояния	484
Заключительные соображения	485
Резюме	486
Эпилог	486
Часть V. Приложения	489
Приложение А. Другие виды моделей предметной области	491
Объектно-ориентированная модель данных, уровень развитого обслуживания и документы	493
В начале	493
Объектная ориентация и реляционные структуры данных	494
Модель предметной области и объектно-реляционное преобразование	494
Уровень обслуживания	495
Объединение на одном уровне	496
Приложение Джимми и АОС	496

16 СОДЕРЖАНИЕ

Структура уровня обслуживания	499
Предоставление данных	501
Степень структурирования	503
Несколько слов о транзакциях	505
Выводы	505
Модель базы данных как модель предметной области	506
Место обитания сущностей	506
Функциональный анализ как основа разработки приложений	511
Прагматизм и нетрадиционный подход	514
Предпосылки и история вопроса	515
Краткий обзор архитектуры	515
Пример приложения Джимми для обработки заказов	519
Бизнес-правила размещения заказов	522
Некоторые заключительные соображения	527
Резюме	527
Приложение Б. Перечень рассмотренных шаблонов	529
Список литературы	536
Предметный указатель	543

Отзывы о книге

“Я не знаю, как назвать то, чем я занимался до того, как пополнил свой инструментарий проектированием на основе предметной области и разработкой посредством тестирования, но теперь я склонен называть это не более чем хаотичным хакерством. Проектирование на основе предметной области и разработка посредством тестирования — это два метода, которыми я постоянно руководствуюсь в практическом применении принципов разработки программного обеспечения, что позволило мне осуществить поставленные цели в моих проектах с помощью надежного, устойчивого программного обеспечения. С точки зрения практики гибкой разработки программного обеспечения эта книга полностью оправдывает свое назначение. Она, вероятно, является одним из самых влиятельных руководств по успешной практике разработки программного обеспечения, предлагавшихся до сих пор сообществу разработчиков программного обеспечения средствами .NET”.

Скотт Беллуэр (Scott Bellware), авторитетный профессионал (Microsoft MVP) по C#, ведущий блог по проектированию на основе предметной области и разработке посредством тестирования

“Джимми Нильсон делает своим читателям большую услугу, показывая им, как применять основополагающие принципы проектирования и разработки приложений для предприятия, выработанные Эвансом, Фаулером и другими ведущими представителями данного направления в программировании. В книге представлены примеры с решением не только для пояснения, но и для демонстрации особенностей проектирования на основе предметной области, шаблонов архитектуры приложений для предприятия и разработки посредством тестирования. Глубокое понимание предмета и практический опыт Джимми делают чтение книги приятным и оставляют у читателя уверенность в том, что он научился чему-то полезному у большого специалиста-практика. Разработчики приложений для предприятия, стремящиеся овладеть этими принципами, обнаружат в книге весьма ценные наставления”.

Джек Гринфилд (Jack Greenfield), разработчик архитектуры инструментария для предприятия, Visual Studio Team System, Microsoft

“Хорошие разработчики архитектуры программного обеспечения должны постоянно совершенствоваться. Помимо целей внедрения текущей системы, они ищут пути совершенствования разработки и создания программного обеспечения. Эта книга является своего рода путевыми заметками, сделанными Джимми по ходу освоения

самых разных шаблонов, практических методов и технологий с пояснением эволюции его представлений о системах предприятия. Если вы идете той же дорогой, эта книга будет вам хорошим попутчиком”.

Тим Евальд (Tim Ewald), главный разработчик архитектуры программного обеспечения в компании Foliage Software Systems и автор книги Transactional COM+: Building Scalable Applications

“Эта книга вносит большой вклад в осуществление крупных, но важных идей проектирования на основе предметной области”.

Флойд Маринеску (Floyd Marinescu), автор книги EJB Design Patterns и создатель Web-сайтов InfoQ.com и TheServerSide.com

“Понимание принципов и движущих сил в предметной области имеет решающее значение для успешной разработки программного обеспечения. Джимми Нильсон почерпнул вдохновение в изучении шаблонов и проектирования на основе предметной области в течение последних десяти лет и воспроизвел свой опыт работы над конкретными проектами. Эта книга содержит замечательные примеры воплощения теории на практике. Она показывает глубокое понимание автором вопросов принятия проектных решений и компромиссов, связанных с объектно-ориентированной разработкой”.

Андерс Хесселлунд (Anders Hessellund), Копенгагенский университет информационных технологий, Дания

“В этой книге рассматривается область, представляющая немалые трудности для освоения большинством разработчиков на платформе .NET. Мне как разработчику структуры шаблонов и практических методов, ставшему инициатором и приложившему немало усилий для выработки руководства по разработке приложений для предприятия в .NET, хорошо известно, насколько эта область важна для наших заказчиков, и я отчетливо осознаю те пробелы, которые все еще существуют в созданном нами руководстве.

Я был тронут тем, как Джимми делится своим опытом проектирования на основе предметной области и разработки посредством тестирования. Полагаю, что в настоящий момент эту тему лучше всего рассматривать с точки зрения простоты, шаблонов и осознания социальных аспектов создания приложений.

Я доверяю опыту Джимми и его знанию .NET, мне по душе его манера делиться принципами и примерами. Мне трудно представить, чтобы кого-то другой смог лучше раскрыть эту тему на примере платформы, с которой я работаю каждый день.

Настоятельно рекомендую книгу Джимми моим заказчикам, коллегам и всем инженерам из Microsoft.

Я надеюсь, что в конечном итоге наша отрасль только выиграет от выражения глубоких понятий в коде и достигнет прогресса в социальном процессе ясной формули-

ровки и эволюции общедоступных знаний. На пути к этому лежит проектирование на основе предметной области и разработка посредством тестирования”.

Эдвард Ежерски (Edward Jezierski), разработчик структуры шаблонов и практических методов Microsoft

“Джимми связал вместе самые передовые методы — проектирование на основе предметной области, разработку посредством тестирования, рефакторинг кода и шаблоны проектирования — в убедительную альтернативу стилю информационной разработки, служившему оплотом приложений Microsoft. Книга демонстрирует воплощение методов, которые пропагандируются в проектировании на основе предметной области. Она имеет ярко выраженный практический характер и дает читателю представление о мыслительном процессе, на котором основывается применение испытанных методов и технологий”.

Кристиан Кроухэрст (Christian Crowhurst), разработчик-аналитик

“Мне пришлось работать с Джимми над совместным проектом в течение 18 месяцев. Джимми действительно живет и работает по тем принципам, которые он проповедует. В книге он описывает свои действия на практике. В частности, он ежедневно пользуется объектно-реляционным преобразованием и преобразователем NHibernate, а разработка посредством тестирования помогает ему понять и вычленил бизнес-логику заказчика, а также убедиться в том, что заказчик получит проверенный, высококачественный код. Эта книга знакомит читателей с проверенными принципами. Пользуйтесь”.

— *Дан Бистрём (Dan Byström), разработчик программного обеспечения, www.visual-design.se/blog*

“Показывая, как выполнять проектирование на основе предметной области и изменять шаблоны приложений для предприятия, Джимми делает трудную тему доступной для широкого круга читателей. В этой книге найдется немало полезного как для опытного разработчика архитектуры программного обеспечения, так и для честолюбивого реализатора шаблонов. Разработчики программного обеспечения или его архитектуры, желающие усовершенствовать свои проекты и результаты внедрения программных решений, должны прочитать эту книгу”.

Пер-Ола Нильссон (Per-Ola Nilsson), руководитель отдела разработки и разработчик архитектуры программного обеспечения в компании Luvit

“Постоянно находясь в курсе новейших тенденций в области разработки программного обеспечения, но не поддаваясь на рекламные провокации, Джимми профессионально отделяет зерна от плевел. Он обладает особым даром наставить и просветить читателя, руководствуясь практической применимостью. Джимми не предлагает

помощь как дар небес — он просто пытается убедить нас в том, что мы можем научиться чему-то совершенно новому у равного нам”.

Матс Хеландер (Mats Helander), разработчик программного обеспечения, www.matshelander.com

“Как и предыдущая книга Джимми, эта книга является настоящим шедевром, идеально сочетающим в себе практические вопросы проектирования программного обеспечения с основательной теорией”.

Франс Боума (Frans Bouma), создатель средства генерации кода LLBLGen Pro, weblogs.asp.net/fbouma

“Опираясь на солидное основание, Джимми рассматривает в своей книге практические вопросы прикладного проектирования на основе предметной области. Это руководство позволит разработчикам справиться с постоянно растущей сложностью новых приложений для предприятия, основанных на модели предметной области”.

Поль Гиленс (Paul Gielens), старший консультант в компании Capgemini, weblogs.asp.net/pgielens

“Опираясь на свой опыт и наполняя в .NET конкретным содержанием такие сокращенные понятия, как DDD и TDD, можно создать весьма эффективную и адаптивную архитектуру программного обеспечения для предприятия. В книге показано, как и почему эти принципы составляют вместе столь эффективное целое. Таким образом, читатель настраивается на бесценный, но краткий путь к успешному созданию архитектуры программного обеспечения для предприятия. Эта книга, без сомнения, будет полезна разработчику такого программного обеспечения”.

Мартин Розен-Лидхольм (Martin Rosén-Lidholm), разработчик архитектуры программного обеспечения в компании Exense Healthcare

“Если ваша задача — построить качественную систему предприятия с объектной ориентацией и реляционной базой данных, не упустите возможность прочитать эту книгу. Вы узнаете, каким образом проектирование на основе предметной области может стать движущей силой всего проекта и как объектно-ориентированная конструкция приводится в соответствие с базой данных — этот этап зачастую рассматривается в литературе очень лаконично, что уже само по себе окупает расходы на данную книгу. Кроме того, здесь вы найдете немало полезных советов и рекомендаций, которым можно последовать при проектировании”.

Ингмар Лундберг (Ingemar Lundberg), разработчик программного обеспечения, www.ingolundberg.com