

Содержание

Предисловие	17
Введение	27
Часть I. Обзор	41
Глава 1. “Расслоение” системы	43
Развитие модели слоев в корпоративных программных приложениях	44
Три основных слоя	46
Где должны функционировать слои	48
Глава 2. Организация бизнес-логики	51
Выбор типового решения	55
Уровень служб	56
Глава 3. Объектные модели и реляционные базы данных	59
Архитектурные решения	59
Функциональные проблемы	64
Считывание данных	66
Взаимное отображение объектов и реляционных структур	67
Отображение связей	67
Наследование	71
Реализация отображения	73
Двойное отображение	74
Использование метаданных	75
Соединение с базой данных	76
Другие проблемы	78
Дополнительные источники информации	79
Глава 4. Представление данных в Web	81
Типовые решения представлений	84
Типовые решения входных контроллеров	86
Дополнительные источники информации	86

8 Содержание

Часть II. Типовые решения	131
Глава 9. Представление бизнес-логики	133
Сценарий транзакции (Transaction Script)	133
Принцип действия	133
Назначение	135
Задача определения зачетного дохода	135
Пример: определение зачетного дохода (Java)	136
Модель предметной области (Domain Model)	140
Принцип действия	140
Назначение	143
Дополнительные источники информации	143
Пример: определение зачетного дохода (Java)	144
Модуль таблицы (Table Module)	148
Принцип действия	149
Назначение	151
Пример: определение зачетного дохода (C#)	152
Слой служб (Service Layer)	156
Принцип действия	157
Разновидности “бизнес-логики”	157
Варианты реализации	157
Быть или не быть удаленному доступу	158
Определение необходимых служб и операций	158
Назначение	160
Дополнительные источники информации	160
Пример: определение зачетного дохода (Java)	161
Глава 10. Архитектурные типовые решения источников данных	167
Шлюз таблицы данных (Table Data Gateway)	167
Принцип действия	167
Назначение	168
Дополнительные источники информации	169
Пример: класс PersonGateway (C#)	170
Пример: использование объектов ADO.NET DataSet (C#)	172
Шлюз записи данных (Row Data Gateway)	175
Принцип действия	175
Назначение	176
Пример: запись о сотруднике (Java)	178
Пример: использование диспетчера данных для объекта домена (Java)	181
Активная запись (Active Record)	182
Принцип действия	182
Назначение	184
Пример: простой класс Person (Java)	184
Преобразователь данных (Data Mapper)	187
Принцип действия	187
Обращение к методам поиска	190

Отображение данных на поля объектов домена	191
Отображения на основе метаданных	192
Назначение	192
Пример: простой преобразователь данных (Java)	193
Пример: отделение методов поиска (Java)	198
Пример: создание пустого объекта (Java)	201
Глава 11. Объектно-реляционные типовые решения, предназначенные для моделирования поведения	205
Единица работы (Unit of Work)	205
Принцип действия	206
Назначение	211
Пример: регистрация посредством изменяемого объекта (Java)	212
Коллекция объектов (Identity Map)	216
Принцип действия	216
Выбор ключей	217
Явная или универсальная?	217
Сколько нужно коллекций?	217
Куда их поместить?	218
Назначение	219
Пример: методы для работы с коллекцией объектов (Java)	219
Загрузка по требованию (Lazy Load)	220
Принцип действия	221
Назначение	223
Пример: инициализация по требованию (Java)	224
Пример: виртуальный прокси-объект (Java)	224
Пример: использование диспетчера значения (Java)	226
Пример: использование фиктивных объектов (C#)	227
Глава 12. Объектно-реляционные типовые решения, предназначенные для моделирования структуры	237
Поле идентификации (Identity Field)	237
Принцип действия	237
Выбор ключа	238
Представление поля идентификации в объекте	239
Вычисление нового значения ключа	240
Назначение	242
Дополнительные источники информации	243
Пример: числовой ключ (C#)	243
Пример: использование таблицы ключей (Java)	244
Пример: использование составного ключа (Java)	246
Класс ключа	246
Чтение	249
Вставка	252
Обновление и удаление	256
Отображение внешних ключей (Foreign Key Mapping)	258
Принцип действия	258
Назначение	261

10 Содержание

Пример: однозначная ссылка (Java)	262
Пример: многотабличный поиск (Java)	265
Пример: коллекция ссылок (C#)	266
Отображение с помощью таблицы ассоциаций (Association Table Mapping)	269
Принцип действия	270
Назначение	270
Пример: служащие и профессиональные качества (C#)	271
Пример: использование SQL для непосредственного обращения к базе данных (Java)	274
Пример: загрузка сведений о нескольких служащих посредством одного запроса (Java)	278
Отображение зависимых объектов (Dependent Mapping)	283
Принцип действия	283
Назначение	285
Пример: альбомы и композиции (Java)	285
Внедренное значение (Embedded Value)	288
Принцип действия	289
Назначение	289
Дополнительные источники информации	290
Пример: простой объект-значение (Java)	290
Сериализованный крупный объект (Serialized LOB)	292
Принцип действия	292
Назначение	294
Пример: сериализация иерархии отделов в формат XML (Java)	294
Наследование с одной таблицей (Single Table Inheritance)	297
Принцип действия	298
Назначение	298
Пример: общая таблица игроков (C#)	299
Загрузка объекта из базы данных	301
Обновление объекта	303
Вставка объекта	303
Удаление объекта	304
Наследование с таблицами для каждого класса (Class Table Inheritance)	305
Принцип действия	305
Назначение	306
Дополнительные источники информации	307
Пример: семейство игроков (C#)	307
Загрузка объекта	307
Обновление объекта	310
Вставка объекта	311
Удаление объекта	312
Наследование с таблицами для каждого конкретного класса (Concrete Table Inheritance)	313
Принцип действия	314
Назначение	315

Пример: конкретные классы игроков (C#)	316
Загрузка объекта из базы данных	318
Обновление объекта	320
Вставка объекта	320
Удаление объекта	321
Преобразователи наследования (Inheritance Mappers)	322
Принцип действия	323
Назначение	324
Глава 13. Типовые решения объектно-реляционного отображения с использованием метаданных	325
Отображение метаданных (Metadata Mapping)	325
Принцип действия	326
Назначение	327
Пример: использование метаданных и метода отражения (Java)	328
Хранение метаданных	328
Поиск по идентификатору	330
Запись в базу данных	332
Извлечение множества объектов	334
Объект запроса (Query Object)	335
Принцип действия	336
Назначение	337
Дополнительные источники информации	337
Пример: простой объект запроса (Java)	337
Хранилище (Repository)	341
Принцип действия	342
Назначение	343
Дополнительные источники информации	344
Пример: поиск подчиненных заданного сотрудника (Java)	344
Пример: выбор стратегий хранилища (Java)	345
Глава 14. Типовые решения, предназначенные для представления данных в Web	347
Модель–представление–контроллер (Model View Controller)	347
Принцип действия	348
Назначение	350
Контроллер страниц (Page Controller)	350
Принцип действия	351
Назначение	352
Пример: простое отображение с помощью контроллера-сервлета и представления JSP (Java)	352
Пример: использование страницы JSP в качестве обработчика запросов (Java)	355
Пример: обработка запросов страницей сервера с применением механизма разделения кода и представления (C#)	358
Контроллер запросов (Front Controller)	362
Принцип действия	362
Назначение	364

12 Содержание

Дополнительные источники информации	364
Пример: простое отображение (Java)	365
Представление по шаблону (Template View)	368
Принцип действия	369
Вставка маркеров	369
Вспомогательный объект	370
Условное отображение	370
Итерация	371
Обработка страницы	372
Использование сценариев	372
Назначение	372
Пример: использование страницы JSP в качестве представления с вынесением контроллера в отдельный объект (Java)	373
Пример: страница сервера ASP.NET (C#)	375
Представление с преобразованием (Transform View)	379
Принцип действия	379
Назначение	380
Пример: простое преобразование (Java)	381
Двухэтапное представление (Two Step View)	383
Принцип действия	383
Назначение	385
Пример: двухэтапное применение XSLT (XSLT)	390
Пример: страницы JSP и пользовательские дескрипторы (Java)	393
Контроллер приложения (Application Controller)	397
Принцип действия	398
Назначение	400
Дополнительные источники информации	400
Пример: модель состояний контроллера приложения (Java)	400
Глава 15. Типовые решения распределенной обработки данных	405
Интерфейс удаленного доступа (Remote Facade)	405
Принцип действия	406
Интерфейс удаленного доступа и типовое решение интерфейс сеанса (Session Facade)	409
Слой служб	409
Назначение	410
Пример: использование компонента сеанса Java в качестве интерфейса удаленного доступа (Java)	410
Пример: Web-служба (C#)	414
Объект переноса данных (Data Transfer Object)	419
Принцип действия	419
Сериализация объекта переноса данных	421
Сборка объекта переноса данных из объектов домена	423
Назначение	424
Дополнительные источники информации	424
Пример: передача информации об альбомах (Java)	425
Пример: сериализация с использованием XML (Java)	429

Глава 16. Типовые решения для обработки задач автономного параллелизма	433
Оптимистическая автономная блокировка (Optimistic Offline Lock)	434
Принцип действия	435
Назначение	439
Пример: слой домена с преобразователями данных (Java)	439
Пессимистическая автономная блокировка (Pessimistic Offline Lock)	445
Принцип действия	446
Назначение	450
Пример: простой диспетчер блокировки (Java)	450
Блокировка с низкой степенью детализации (Coarse-Grained Lock)	457
Принцип действия	457
Назначение	460
Пример: общая оптимистическая автономная блокировка (Java)	460
Пример: общая пессимистическая автономная блокировка (Java)	466
Пример: оптимистическая автономная блокировка корневого элемента (Java)	467
Неявная блокировка (Implicit Lock)	468
Принцип действия	469
Назначение	470
Пример: неявная пессимистическая автономная блокировка (Java)	470
Глава 17. Типовые решения для хранения состояния сеанса	473
Сохранение состояния сеанса на стороне клиента (Client Session State)	473
Принцип действия	473
Назначение	474
Сохранение состояния сеанса на стороне сервера (Server Session State)	475
Принцип действия	475
Назначение	478
Сохранение состояния сеанса в базе данных (Database Session State)	479
Принцип действия	479
Назначение	481
Глава 18. Базовые типовые решения	483
Шлюз (Gateway)	483
Принцип действия	484
Назначение	484
Пример: создание шлюза к службе отправки сообщений (Java)	485
Преобразователь (Mapper)	489
Принцип действия	490
Назначение	490
Супертип слоя (Layer Supertype)	491
Принцип действия	491
Назначение	491
Пример: объект домена (Java)	491
Отделенный интерфейс (Separated Interface)	492
Принцип действия	493
Назначение	494

Глава 5. Управление параллельными заданиями	87
Проблемы параллелизма	88
Контексты выполнения	89
Изолированность и устойчивость данных	91
Стратегии блокирования	91
Предотвращение возможности несогласованного чтения данных	93
Разрешение взаимоблокировок	94
Транзакции	95
ACID: свойства транзакций	96
Ресурсы транзакций	96
Уровни изоляции	97
Системные транзакции и бизнес-транзакции	99
Типовые решения задачи обеспечения автономного параллелизма	101
Параллельные операции и серверы приложений	102
Дополнительные источники информации	104
Глава 6. Сеансы и состояния	105
В чем преимущество отсутствия “состояния”	105
Состояние сеанса	107
Способы сохранения состояния сеанса	108
Глава 7. Стратегии распределенных вычислений	111
Соблазны модели распределенных объектов	111
Интерфейсы локального и удаленного вызова	112
Когда без распределения не обойтись	114
Сужение границ распределения	115
Интерфейсы распределения	116
Глава 8. Общая картина	119
Предметная область	120
Источник данных	121
Источник данных для сценария транзакции	121
Источник данных для модуля таблицы	122
Источник данных для модели предметной области	122
Слой представления	123
Платформы и инструменты	124
Java и J2EE	124
.NET	125
Хранимые процедуры	126
Web-службы	126
Другие модели слоев	127

14 Содержание

Реестр (Registry)	495
Принцип действия	495
Назначение	497
Пример: реестр с единственным экземпляром (Java)	498
Пример: реестр, уникальный в пределах потока (Java)	499
Объект-значение (Value Object)	500
Принцип действия	501
Назначение	502
Совпадение названий	502
Деньги (Money)	502
Принцип действия	503
Назначение	506
Пример: класс Money (Java)	506
Частный случай (Special Case)	511
Принцип действия	512
Назначение	512
Дополнительные источники информации	512
Пример: объект NullEmployee (C#)	513
Дополнительный модуль (Plugin)	514
Принцип действия	514
Назначение	515
Пример: генератор идентификаторов (Java)	516
Фиктивная служба (Service Stub)	519
Принцип действия	519
Назначение	520
Пример: служба определения величины налога (Java)	521
Множество записей (Record Set)	523
Принцип действия	524
Явный интерфейс	524
Назначение	526
Список основных источников информации	527
Предметный указатель	532