Александру, за то, что научил меня, как учить его.
– Гэри Грегори

Содержание

Предисловие к первому изданию ... 16
Введение ... 18
Благодарности ... 19
Об этой книге ... 21
Об изображении на обложке ... 24

Часть I. Начинаем работать с ORM ... 25

Глава 1. Основы объектно-реляционного отображения ... 26
1.1. Что такое долговременное хранение? ... 27
1.1.1. Реляционные базы данных ... 28
1.1.2. Разбираемся с SQL ... 29
1.1.3. Использование SQL в Java ... 30
1.2. Несоответствие парадигм ... 32
1.2.1. Проблема детализации ... 33
1.2.2. Проблема подтипов ... 35
1.2.3. Проблема идентичности ... 37
1.2.4. Проблемы, связанные с ассоциациями ... 38
1.2.5. Проблемы навигации по данным ... 39
1.3. ORM и JPA ... 41
1.4. Резюме ... 43

Глава 2. Создаем проект ... 44
2.1. Представляем Hibernate ... 44
2.2. «HELLO WORLD» и JPA ... 45
2.2.1. Настройка единицы хранения ... 46
2.2.2. Хранимый класс ... 47
2.2.3. Сохранение и загрузка сообщений ... 49
2.3. Оригинальная конфигурация Hibernate ... 51
2.4. Резюме ... 54

Глава 3. Модели предметной области и метаданные ... 55
3.1. Учебное приложение CaveatEmptor ... 56
3.1.1. Многоуровневая архитектура ... 56
3.1.2. Анализ предметной области ... 58
3.1.3. Предметная модель приложения CaveatEmptor ... 59

3.2. Реализация предметной модели ... 61
3.2.1. Предотвращение утечек функциональности ... 61
3.2.2. Прозрачность сохранения и его автоматизация ... 62
3.2.3. Создание классов с возможностью сохранения ... 64
3.2.4. Реализация ассоциаций в POJO ... 67
3.3. Метаданные предметной модели ... 72
3.3.1. Определение метаданных с помощью аннотаций ... 73
3.3.2. Применение правил валидации компонентов ... 75
3.3.3. Метаданные во внешних XML-файлах ... 78
3.3.4. Доступ к метаданным во время выполнения ... 82
3.4. Резюме ... 86

Часть II. Стратегии отображения ... 87

Глава 4. Отображение хранимых классов ... 88
4.1. Понятие сущностей и типов-значений ... 88
4.1.1. Хорошо детализированные модели предметной области ... 89
4.1.2. Определение сущностей приложения ... 89
4.1.3. Разделение сущностей и типов-значений ... 91
4.2. Отображение сущностей с идентичностью ... 93
4.2.1. Идентичность и равенство в Java ... 93
4.2.2. Первый класс сущности и его отображение ... 94
4.2.3. Выбор первичного ключа ... 95
4.2.4. Настройка генераторов ключей ... 97
4.2.5. Стратегии генерации идентификаторов ... 99
4.3. Способы отображений сущностей ... 103
4.3.1. Управление именами ... 103
4.3.2. Динамическое формирование SQL ... 106
4.3.3. Неизменяемые сущности ... 107
4.3.4. Отображение сущности в подзапрос ... 108
4.4. Резюме ... 110

Глава 5. Отображение типов-значений ... 111
5.1. Отображение полей основных типов ... 112
5.1.1. Переопределение настроек по умолчанию для свойств основных типов ... 112
5.1.2. Настройка доступа к свойствам ... 114
5.1.3. Работа с вычисляемыми полями ... 116
5.1.4. Преобразование значений столбцов ... 117
5.1.5. Значения свойств, генерируемые по умолчанию ... 118
5.1.6. Свойства для представления времени ... 119
5.1.7. Отображение перечислений ... 120
5.2. Отображение встраиваемых компонентов ... 121
5.2.1. Схема базы данных ... 121

5.2.2. Встраиваемые классы 122
5.2.3. Переопределение встроенных атрибутов 125
5.2.4. Отображение вложенных встраиваемых компонентов 126
5.3. Отображение типов Java и SQL с применением конвертеров 128
5.3.1. Встроенные типы 128
5.3.2. Создание собственных конвертеров JPA 135
5.3.3. Расширение Hibernate с помощью пользовательских типов 141
5.4. Резюме 148

Глава 6. Отображение наследования 150
6.1. Одна таблица для каждого конкретного класса и неявный полиморфизм 151
6.2. Одна таблица для каждого конкретного класса с объединениями 153
6.3. Единая таблица для целой иерархии классов 156
6.4. Одна таблица для каждого подкласса с использованием соединений 159
6.5. Смешение стратегий отображения наследования 163
6.6. Наследование и встраиваемые классы 165
6.7. Выбор стратегии 168
6.8. Полиморфные ассоциации 170
6.8.1. Полиморфная ассоциация *многие к одному* (many-to-one) 170
6.8.2. Полиморфные коллекции 173
6.9. Резюме 174

Глава 7. Отображение коллекций и связей между сущностями 175
7.1. Множества, контейнеры, списки и словари с типами-значениями 176
7.1.1. Схема базы данных 176
7.1.2. Создание и отображение поля коллекции 176
7.1.3. Выбор интерфейса коллекции 178
7.1.4. Отображение множества 180
7.1.5. Отображение контейнера идентификаторов 181
7.1.6. Отображение списка 182
7.1.7. Отображение словаря 184
7.1.8. Отсортированные и упорядоченные коллекции 185
7.2. Коллекции компонентов 188
7.2.1. Равенство экземпляров компонентов 189
7.2.2. Множество компонентов 191
7.2.3. Контейнер компонентов 193
7.2.4. Словарь с компонентами в качестве значений 194
7.2.5. Компоненты в роли ключей словаря 195
7.2.6. Коллекции во встраиваемых компонентах 197
7.3. Отображение связей между сущностями 198
7.3.1. Самая простая связь 199
7.3.2. Определение двунаправленной связи 200

7.3.3. Каскадная передача состояния....202
7.4. Резюме....209

Глава 8. Продвинутые приемы отображения связей между сущностями....211
8.1. Связи *один к одному*....212
8.1.1. Общий первичный ключ....212
8.1.2. Генератор внешнего первичного ключа....215
8.1.3. Соединение с помощью столбца внешнего ключа....218
8.1.4. Использование таблицы соединения....220
8.2. Связь *один ко многим*....222
8.2.1. Применение контейнеров в связях *один ко многим*....223
8.2.2. Однонаправленное и двунаправленное отображения списка....224
8.2.3. Необязательная связь *один ко многим* с таблицей соединения....227
8.2.4. Связь *один ко многим* во встраиваемых классах....229
8.3. Тройные связи и связи *многие ко многим*....231
8.3.1. Однонаправленные и двунаправленные связи *многие ко многим*....232
8.3.2. Связь *многие ко многим* с промежуточной сущностью....234
8.3.3. Тройные связи с компонентами....239
8.4. Связи между сущностями с использованием словарей....242
8.4.1. Связь *один ко многим* со свойством для ключа....242
8.4.2. Тройное отношение вида ключ/значение....243
8.5. Резюме....245

Глава 9. Сложные и унаследованные схемы....246
9.1. Улучшаем схему базы данных....247
9.1.1. Добавление вспомогательных объектов базы данных....248
9.1.2. Ограничения SQL....251
9.1.3. Создание индексов....258
9.2. Унаследованные первичные ключи....259
9.2.1. Отображение естественных первичных ключей....259
9.2.2. Отображение составных первичных ключей....260
9.2.3. Внешние ключи внутри составных первичных ключей....262
9.2.4. Внешний ключ, ссылающийся на составной первичный ключ....266
9.2.5. Внешние ключи, ссылающиеся на непервичные ключи....267
9.3. Отображение свойств во вторичные таблицы....268
9.4. Резюме....270

Часть III. Транзакционная обработка данных....271

Глава 10. Управление данными....272
10.1. Жизненный цикл хранения....273
10.1.1. Состояния экземпляров сущностей....273

10.1.2. Контекст хранения ... 275
10.2. Интерфейс EntityManager ... 277
10.2.1. Каноническая форма единицы работы ... 277
10.2.2. Сохранение данных ... 279
10.2.3. Извлечение и модификация хранимых данных ... 280
10.2.4. Получение ссылки на объект ... 282
10.2.5. Переход данных во временное состояние ... 283
10.2.6. Изменение данных в памяти ... 285
10.2.7. Репликация данных ... 285
10.2.8. Кеширование в контексте хранения ... 286
10.2.9. Выталкивание контекста хранения ... 288
10.3. Работа с отсоединенным состоянием ... 289
10.3.1. Идентичность отсоединенных экземпляров ... 289
10.3.2. Реализация метода проверки равенства ... 292
10.3.3. Отсоединение экземпляров сущностей ... 295
10.3.4. Слияние экземпляров сущностей ... 296
10.4. Резюме ... 298

Глава 11. Транзакции и многопоточность ... 299
11.1. Основы транзакций ... 300
11.1.1. Атрибуты ACID ... 300
11.1.2. Транзакции в базе данных и системные транзакции ... 300
11.1.3. Программные транзакции с JTA ... 301
11.1.4. Обработка исключений ... 303
11.1.5. Декларативное определение границ транзакции ... 306
11.2. Управление параллельным доступом ... 307
11.2.1. Многопоточность на уровне базы данных ... 307
11.2.2. Оптимистическое управление параллельным доступом ... 313
11.2.3. Явные пессимистические блокировки ... 322
11.2.4. Как избежать взаимоблокировок ... 325
11.3. Доступ к данным вне транзакции ... 327
11.3.1. Чтение данных в режиме автоматического подтверждения ... 328
11.3.2. Создание очереди изменений ... 330
11.4. Резюме ... 332

Глава 12. Планы извлечения, стратегии и профили ... 333
12.1. Отложенная и немедленная загрузка ... 334
12.1.1. Прокси-объекты ... 335
12.1.2. Отложенная загрузка хранимых коллекций ... 339
12.1.3. Реализация отложенной загрузки путем перехвата вызовов ... 342
12.1.4. Немедленная загрузка коллекций и ассоциаций ... 345
12.2. Выбор стратегии извлечения ... 347
12.2.1. Проблема n + 1 выражений SELECT ... 347

12.2.2. Проблема декартова произведения 348
12.2.3. Массовая предварительная выборка данных 351
12.2.4. Предварительное извлечение коллекций с помощью подзапросов 354
12.2.5. Отложенное извлечение с несколькими выражениями SELECT 355
12.2.6. Динамическое немедленное извлечение 356
12.3. Профили извлечения 358
12.3.1. Определение профилей извлечения Hibernate 359
12.3.2. Графы сущностей 360
12.4. Резюме 364

Глава 13. Фильтрация данных 365
13.1. Каскадная передача изменений состояния 366
13.1.1. Доступные способы каскадирования 367
13.1.2. Транзитивное отсоединение и слияние 367
13.1.3. Каскадное обновление 370
13.1.4. Каскадная репликация 372
13.1.5. Глобальное каскадное сохранение 373
13.2. Прием и обработка событий 374
13.2.1. Приемники событий JPA и обратные вызовы 374
13.2.2. Реализация перехватчиков Hibernate 378
13.2.3. Базовый механизм событий 383
13.3. Аудит и версионирование с помощью Hibernate Envers 384
13.3.1. Включение ведения журнала аудита 384
13.3.2. Ведение аудита 386
13.3.3. Поиск версий 387
13.3.4. Получение архивных данных 388
13.4. Динамическая фильтрация данных 391
13.4.1. Создание динамических фильтров 392
13.4.2. Применение фильтра 392
13.4.3. Активация фильтра 393
13.4.4. Фильтрация коллекций 394
13.5. Резюме 395

Часть IV. Создание запросов 397

Глава 14. Создание и выполнение запросов 398
14.1. Создание запросов 399
14.1.1. Интерфейсы запросов JPA 399
14.1.2. Результаты типизированных запросов 402
14.1.3. Интерфейсы Hibernate для работы с запросами 402
14.2. Подготовка запросов 404
14.2.1. Защита от атак на основе внедрения SQL-кода 404
14.2.2. Связывание именованных параметров 405

14.2.3. Связывание позиционных параметров 406
14.2.4. Постраничная выборка больших наборов с результатами 407
14.3. Выполнение запросов 409
14.3.1. Извлечение полного списка результатов 409
14.3.2. Получение единичных результатов 409
14.3.3. Прокрутка с помощью курсоров базы данных 411
14.3.4. Обход результатов с применением итератора 412
14.4. Обращение к запросам по именам и их удаление из программного кода 413
14.4.1. Вызов именованных запросов 414
14.4.2. Хранение запросов в метаданных XML 414
14.4.3. Хранение запросов в аннотациях 416
14.4.4. Программное создание именованных запросов 416
14.5. Подсказки для запросов 417
14.5.1. Установка предельного времени выполнения 418
14.5.2. Установка режима выталкивания контекста хранения 419
14.5.3. Установка режима только для чтения 419
14.5.4. Определение количества одновременно извлекаемых записей 420
14.5.5. Управление комментариями SQL 420
14.5.6. Подсказки для именованных запросов 421
14.6. Резюме 422

Глава 15. Языки запросов 424
15.1. Выборка 425
15.1.1. Назначение псевдонимов и определение корневых источников запроса 426
15.1.2. Полиморфные запросы 427
15.2. Ограничения 428
15.2.1. Выражения сравнения 430
15.2.2. Выражения с коллекциями 434
15.2.3. Вызовы функций 435
15.2.4. Упорядочение результатов запроса 438
15.3. Проекции 439
15.3.1. Проекция сущностей и скалярных значений 439
15.3.2. Динамическое создание экземпляров 441
15.3.3. Извлечение уникальных результатов 443
15.3.4. Вызов функций в проекциях 443
15.3.5. Агрегирующие функции 446
15.3.6. Группировка данных 447
15.4. Соединения 449
15.4.1. Соединения в SQL 449
15.4.2. Соединение таблиц в JPA 452
15.4.3. Неявные соединения по связи 452
15.4.4. Явные соединения 454

15.4.5. Динамическое извлечение с помощью соединений 456
15.4.6. Тета-соединения 460
15.4.7. Сравнение идентификаторов 461
15.5. Подзапросы 463
15.5.1. Коррелированные и некоррелированные подзапросы 463
15.5.2. Кванторы 464
15.6. Резюме 466

Глава 16. Дополнительные возможности запросов 467
16.1. Преобразование результатов запросов 467
16.1.1. Получение списка списков 469
16.1.2. Получение списка словарей 469
16.1.3. Отображение атрибутов в свойства компонента JavaBean 470
16.1.4. Создание преобразователя ResultTransformer 471
16.2. Фильтрация коллекций 472
16.3. Интерфейс запросов на основе критериев в Hibernate 475
16.3.1. Выборка и упорядочение 475
16.3.2. Ограничения 476
16.3.3. Проекция и агрегирование 478
16.3.4. Соединения 479
16.3.5. Подзапросы 481
16.3.6. Запросы по образцу 482
16.4. Резюме 484

Глава 17. Настройка SQL-запросов 485
17.1. Назад к JDBC 486
17.2. Отображение результатов SQL-запросов 488
17.2.1. Проекции в SQL-запросах 489
17.2.2. Отображение в классы сущностей 490
17.2.3. Настройка отображения запросов 492
17.2.4. Размещение обычных запросов в отдельных файлах 504
17.3. Настройка операций CRUD 509
17.3.1. Подключение собственных загрузчиков 509
17.3.2. Настройка операций создания, изменения, удаления 510
17.3.3. Настройка операций над коллекциями 512
17.3.4. Немедленное извлечение в собственном загрузчике 514
17.4. Вызов хранимых процедур 517
17.4.1. Возврат результата запроса 518
17.4.2. Возврат нескольких результатов и количества изменений 519
17.4.3. Передача входных и выходных аргументов 521
17.4.4. Возвращение курсора 524
17.5. Применение хранимых процедур для операций CRUD 526
17.5.1. Загрузчик, вызывающий процедуру 526

17.5.2. Использование процедур в операциях CUD 527
17.6. Резюме 529

Часть V. Создание приложений 531

Глава 18. Проектирование клиент-серверных приложений 532
18.1. Разработка уровня хранения 533
18.1.1. Обобщенный шаблон «объект доступа к данным» 535
18.1.2. Реализация обобщенных интерфейсов 537
18.1.3. Реализация интерфейсов DAO 539
18.1.4. Тестирование уровня хранения 541
18.2. Создание сервера без состояния 543
18.2.1. Редактирование информации о товаре 543
18.2.2. Размещение ставки 546
18.2.3. Анализ приложения без состояния 550
18.3. Разработка сервера с сохранением состояния 552
18.3.1. Редактирование информации о товаре 553
18.3.2. Анализ приложений с сохранением состояния 558
18.4. Резюме 561

Глава 19. Создание веб-приложений 562
19.1. Интеграция JPA и CDI 563
19.1.1. Создание экземпляра EntityManager 563
19.1.2. Присоединение экземпляра EntityManager к транзакциям 565
19.1.3. Внедрение экземпляра EntityManager 565
19.2. Сортировка и постраничная выборка данных 567
19.2.1. Реализация постраничной выборки с помощью смещения или поиска 567
19.2.2. Реализация постраничной выборки в уровне хранения 570
19.2.3. Постраничная выборка 576
19.3. Создание JSF-приложений 577
19.3.1. Службы с областью видимости запроса 578
19.3.2. Службы с областью видимости диалога 581
19.4. Сериализация данных предметной модели 590
19.4.1. Создание JAX-RS-службы 591
19.4.2. Применение JAXB-отображений 592
19.4.3. Сериализация прокси-объектов Hibernate 595
19.5. Резюме 598

Глава 20. Масштабирование Hibernate 599
20.1. Массовые и пакетные операции обработки данных 600
20.1.1. Массовые операции в запросах на основе критериев и JPQL 600
20.1.2. Массовые операции в SQL 605
20.1.3. Пакетная обработка данных 606

20.1.4. Интерфейс StatelessSession 610
20.2. Кеширование данных 612
20.2.1. Архтектура общего кеша в Hibernate 613
20.2.2. Настройка общего кеша 618
20.2.3. Кеширование коллекций и сущностей 619
20.2.4. Проверка работы разделяемого кеша 623
20.2.5. Установка режимов кеширования 625
20.2.6. Управление разделяемым кешем 627
20.2.7. Кеш результатов запросов 627
20.3. Резюме 630

Библиография 631

Предметный указатель 632

Предисловие к первому изданию

Реляционные базы данных, бесспорно, составляют основу современного предприятия. В то время как современные языки программирования, включая Java, обеспечивают интуитивное, объектно-ориентированное представление бизнес-сущностей уровня приложения, данные, лежащие в основе этих сущностей, имеют выраженную реляционную природу. Кроме того, главное преимущество реляционной модели перед более ранней навигационной моделью, а также поздними моделями объектно-ориентированных баз данных – в том, что она изначально внутренне независима от программных взаимодействий и представления данных на уровне приложения. Было предпринято немало попыток для объединения реляционной и объектно-ориентированной технологий или замещения одной на другую, но пропасть между ними остается сегодня одним из непреложных фактов. Именно эту задачу – обеспечить связь между реляционными данными и Java-объектами – решает Hibernate при помощи своего подхода к реализации объектно-реляционного отображения (Object/Relational Mapping, ORM). Hibernate решает данную задачу очень прагматичным, ясным и реалистичным способом.

Как показывают Кристиан Бауэр (Christian Bauer) и Гэвин Кинг (Gavin King) в этой книге, эффективное использование технологии ORM в любом бизнес-окружении, кроме простейшего, требует понимания особенностей работы механизма, осуществляющего посредничество между реляционными данными и объектами. То есть разработчик должен понимать требования к своему приложению и к данным, владеть языком SQL, знать структуры реляционных хранилищ, а также иметь представление о возможных способах оптимизации, предоставляемых реляционными технологиями. Hibernate не только предоставляет полностью рабочее решение, непосредственно удовлетворяющее этим требованиям, но и гибкую и настраиваемую архитектуру. Разработчики Hibernate изначально сделали фреймворк модульным, расширяемым и легко настраиваемым под нужды пользователя. В результате через несколько лет после выхода первой версии фреймворк Hibernate быстро стал – и заслуженно – одной из ведущих реализаций ORM-технологий для разработчиков корпоративных приложений.

В этой книге представлен подробный обзор фреймворка Hibernate. Описывается, как использовать его возможности отображения типов и средства моделирования ассоциаций и наследования; как эффективно извлекать объекты, используя

язык запросов Hibernate; как настраивать Hibernate для работы в управляемом и неуправляемом окружениях; как использовать его инструментарий. Кроме того, на протяжении всей книги авторы приоткрывают проблемы ORM и проектные решения, легшие в основу Hibernate. Это дает читателю глубокое понимание эффективного использования ORM как корпоративной технологии. Данная книга является подробным руководством по Hibernate и объектно-реляционному отображению в корпоративных вычислениях.

Линда Демишель (*Linda DeMichiel*)
Ведущий архитектор, Enterprise Javabeans
Sun Microsystems
Ноябрь 2012

Введение

Это наша третья книга о Hibernate, проекте с открытым исходным кодом, которому почти 15 лет. Согласно недавнему опросу, Hibernate оказался в числе пяти самых популярных инструментов, которыми Java-разработчики пользуются каждый день. Это говорит о том, что базы данных SQL являются предпочтительной технологией для надежного хранения и управления данными, особенно в области разработки корпоративных приложений на Java. Также это является доказательством качества доступных спецификаций и инструментов, упрощающих запуск проектов, оценку и снижение рисков при создании крупных и сложных приложений.

Сейчас доступны пятая версия Hibernate и вторая версия спецификации Java Persistence API (JPA), которую реализует Hibernate. Ядро Hibernate или то, что сейчас называется объектно-реляционным отображением (Object/Relational Mapping, ORM), уже долгое время является развитой технологией, и на протяжении многих лет было сделано множество мелких улучшений. Другие связанные проекты, такие как Hibernate Search, Hibernate Bean Validation и недавнее объектно-сеточное отображение (Object/Grid Mapping, OGM), привносят новые инновационные решения, которые превращают Hibernate в полноценный набор инструментов для решения широкого спектра задач управления данными.

Когда мы работали над предыдущим изданием этой книги, с Hibernate происходили важные изменения: в силу своего органичного развития, влияния со стороны сообщества разработчиков свободного ПО и повседневных требований Java-разработчиков Hibernate пришлось стать более формальным и реализовать первую версию спецификации JPA (Java Persistence API). Поэтому предыдущее издание получилось громоздким, так как многие примеры мы были вынуждены демонстрировать с использованием старого и нового, стандартизированного, подходов.

В настоящий момент этот расхождение практически исчезло, и мы можем в первую очередь опираться на стандартизированный прикладной программный интерфейс (API) и архитектуру Java Persistence. Также в этом издании мы обсудим множество выдающихся особенностей Hibernate. Хотя объем книги уменьшился, по сравнению с предыдущим изданием, мы использовали это место для многочисленных новых примеров. Мы также рассмотрим, как JPA вписывается в общую картину Java EE и как ваше приложение может интегрировать Bean Validation, EJB, CDI и JSF.

Пусть это новое издание станет путеводителем для вашего первого проекта Hibernate. Мы надеемся, что оно заменит предыдущее издание в качестве настольного справочного материала по Hibernate.

Благодарности

Мы не смогли бы написать эту книгу без помощи многих людей. Палак Матур (Palak Mathur) и Кристиан Альфано (Christian Alfano) проделали отличную работу, будучи техническими рецензентами нашей книги; спасибо вам за долгие часы, потраченные на редактирование наших «поломанных» примеров кода.

Мы также хотели бы поблагодарить наших рецензентов за потраченное время и неоценимую обратную связь в процессе разработки: Криса Бакара (Chris Bakar), Гаурава Бхардвая (Gaurav Bhardwaj), Якоба Босму (Jacob Bosma), Хосе Диаза (José Diaz), Марко Гамбини (Marco Gambini), Серхио Фернандеса Гонсалеса (Sergio Fernandez Gonzalez), Джерри Гуднафа (Jerry Goodnough), Джона Гриффина (John Griffin), Стефана Хеффнера (Stephan Heffner), Чеда Джонстона (Chad Johnston), Кристофа Мартини (Christophe Martini), Робби О'Коннора (Robby O'Connor), Антони Патрисио (Anthony Patricio) и Дениса Ванга (Denis Wang).

Издатель из Manning Марьян Бэйс (Marjan Bace) снова собрала отличную команду в Manning: Кристина Тэйлор (Christina Taylor) редактировала нашу сырую рукопись, превратив ее в настоящую книгу. Тиффани Тэйлор (Tiffany Taylor) нашла все опечатки, сделав книгу читаемой. Дотти Мариско (Dottie Marisco), ответственная за верстку, придала книге ее прекрасный вид. Мэри Пиргис (Mary Piergies) координировала и организовывала весь процесс. Мы благодарим всех вас за то, что работали с нами.

Наконец, отдельное спасибо Линде Демишель (Linda DeMichiel) за предисловие к первому изданию.

Гэри Грегори (Gary Gregory)

Я хотел бы поблагодарить моих родителей за то, что помогли мне начать это путешествие, дали мне прекрасное образование и свободу выбирать свой путь. Я бесконечно благодарен своей жене Лори и моему сыну Александру за то, что они предоставили мне время на завершение еще одного проекта – моей третьей книги.

В процессе работы я учился и сотрудничал с действительно выдающимися личностями, такими как Джордж Босворт (George Bosworth), Ли Брайзахер (Lee Breisacher), Кристофер Хэнсон (Christopher Hanson), Дебора Льюис (Deborah Lewis) и многими другими. Мой тесть, Бадди Мартин (Buddy Martin), заслуживает особого упоминания за то, что делился своей мудростью и проницательностью во время долгих бесед и рассказов историй, накопленных за те десятилетия, что он писал о спорте (вперед, «Аллигаторы»!). Я всегда нахожу вдохновение в му-

зыке, особенно в музыке следующих исполнителей: Wilco (Impossible Germany), Tom Waits (Blue Valentine), Donald Fagen (The Night-fly, A just machine to make big decisions/Programmed by fellows with compassion and vision), David Lindley и Баха. Наконец, я благодарю своего соавтора Кристиана Бауэра (Christian Bauer) за то, что делился со мной знаниями, и всех сотрудников издательства Manning за поддержку, профессионализм и доброжелательные отзывы.

Отдельное спасибо Тиффани Тэйлор (Tiffany Taylor) из Manning за то, что придала книге отличный вид. Дон Вэннер (Don Wanner), спасибо, точка.

Об этой книге

Эта книга является и руководством, и справочником по Hibernate и Java Persistence. Если вы новичок в Hibernate, рекомендуем читать, начиная с главы 1, и опробовать все примеры с «Hello World» в главе 2. Если вы использовали старую версию Hibernate, прочитайте бегло две первые главы, чтобы получить общее представление, и переходите к середине главы 3. Там, где необходимо, мы сообщим, если конкретный раздел или тема являются дополнительными или справочными, которые можно пропустить при первом чтении.

Структура книги

Эта книга состоит из пяти больших частей.

В части I «Начинаем работать с ORM» мы обсудим основы объектно-реляционного отображения. Пройдемся по практическому руководству, чтобы помочь вам создать первый проект с Hibernate. Рассмотрим проектирование Java-приложений с точки зрения создания моделей предметной области и познакомимся с вариантами определения метаданных для объектно-реляционного отображения.

В части II «Стратегии отображения» рассказывается о классах Java и их свойствах, а также об их отображении в таблицы и столбцы SQL. Мы рассмотрим все основные и продвинутые способы отображения в Hibernate и Java Persistence. Покажем, как работать с наследованием, коллекциями и сложными ассоциациями классов. В заключение обсудим интеграцию с существующими схемами баз данных, а также особенно запутанные стратегии отображения.

Часть III «Транзакционная обработка данных» посвящена загрузке и сохранению данных с помощью Hibernate и Java Persistence. Мы представим программные интерфейсы, способы создания транзакционных приложений и то, как эффективнее загружать данные из базы данных в Hibernate.

В части IV «Создание запросов» мы познакомимся с механикой извлечения данных и детально рассмотрим язык запросов и прикладной программный интерфейс (API). Не все главы данного раздела написаны как руководство; мы рассчитываем, что вы будете часто заглядывать в эту часть книги во время разработки приложений для поиска решений проблем с конкретными запросами.

В части V «Разработка приложений» мы обсудим проектирование и разработку многоуровневых приложений баз данных на Java. Обсудим наиболее распространенные шаблоны проектирования, используемые вместе с Hibernate, такие как «Объект доступа к данным» (Data Access Object, DAO). Вы увидите, как можно легко протестировать приложение, использующее Hibernate, и познакомитесь с другими распространенными приемами, используемыми при работе с инструментами объектно-реляционного отображения в веб-приложениях или клиент-серверных приложениях в целом.

Кому адресована эта книга?

Читатели данной книги должны быть знакомы с основами объектно-ориентированного программирования и иметь практические навыки его применения. Чтобы понять примеры приложений, вы должны быть знакомы с языком программирования Java и унифицированным языком моделирования (Unified Modeling Language, UML).

В первую очередь книга адресована Java-программистам, работающим с базами данных SQL. Мы покажем, как повысить продуктивность при работе с ORM. Если вы – разработчик баз данных, эта книга может отчасти послужить вам введением в объектно-ориентированную разработку программного обеспечения.

Если вы администратор баз данных (DBA), вас наверняка заинтересует влияние ORM на производительность, а также способы настройки СУБД SQL и уровня хранения данных для достижения желаемой производительности. Доступ к данным является узким местом любого Java-приложения, поэтому проблемам производительности в данной книге уделяется повышенное внимание. Многие DBA, по понятным причинам, испытывают тревогу, не веря в высокую производительность автоматически сгенерированного кода SQL; мы постараемся развеять эти страхи, а также указать случаи, когда в приложениях не следует использовать автоматического доступа к данным. Вы почувствуете себя свободнее, когда поймете, что мы не считаем ORM лучшим решением для каждой проблемы.

Соглашения об оформлении программного кода

Эта книга изобилует примерами, включающими всевозможные артефакты Hibernate-приложений: Java-код, файлы конфигурации Hibernate и XML-файлы с метаданными отображений. Исходный код в листингах или тексте `выделен моноширинным шрифтом, как этот`, чтобы отделить его от остального текста. Кроме того, имена Java-методов, параметры компонентов, свойства объектов, а также XML-элементы и их атрибуты в тексте представлены с использованием `моноширинного шрифта`.

Листинги на Java, XML и HTML могут быть объемными. Во многих случаях исходный код (доступный онлайн) был переформатирован; мы добавили разделители строк и переделали отступы, чтобы уместить их по ширине книжных страниц. В редких случаях, когда этого оказалось недостаточно, в листинги были добавлены символы продолжения строки (➥). Также из многих листингов, описываемых в тексте, мы убрали комментарии. Некоторые листинги сопровождают аннотации, выделяющие важные понятия. В других случаях используются пронумерованные маркеры, отсылающие к пояснениям в тексте, следующим за листингами.

Загрузка исходного кода

Hibernate – это проект с открытым исходным кодом, распространяемым на условиях лицензии Lesser GNU Public Lisence. Инструкции по загрузке модулей Hibernate в виде исходных или двоичных кодов доступны на сайте проекта: http://

hibernate.org/. Исходный код всех примеров в данной книге доступен на http://jpwh.org. Код примеров можно также загрузить с сайта издательства https://www.manning.com/books/java-persistence-with-hibernate-second-edition.

Автор онлайн

Приобретая книгу «Java Persistence API и Hibernate», вы получаете бесплатный доступ на частный веб-форум издательства Manning Publications, где вы сможете оставлять отзывы о книге, задавать технические вопросы и получать помощь от авторов и других пользователей. Чтобы получить доступ к форуму и зарегистрироваться на нем, откройте в браузере страницу https://www.manning.com/books/java-persistence-with-hibernate-second-edition. На этой странице «Author Online» (Автор в сети) описывается, как попасть на форум после регистрации, какие виды помощи доступны и правила поведения на форуме.

Издательство Manning обязуется предоставить своим читателям место встречи, где может состояться содержательный диалог между отдельными читателями и между читателями и автором. Но со стороны автора отсутствуют какие-либо обязательства уделять форуму какое-то определенное внимание – его присутствие на форуме остается добровольным (и неоплачиваемым). Мы предлагаем задавать автору стимулирующие вопросы, чтобы его интерес не угасал!

Форум и архивы предыдущих дискуссий будут оставаться доступными, пока книга продолжает издаваться.

Об авторах

Кристиан Бауэр (Christian Bauer) – член коллектива разработчиков Hibernate; он работает инструктором и консультантом.

Гэвин Кинг (Gavin King) – основатель проекта Hibernate и член первоначального состава экспертной группы по Java Persistence (JSR 220). Он также участвовал в работе по стандартизации CDI (JSR 299). В настоящее время Гэвин разрабатывает новый язык программирования Ceylon.

Гэри Грегори (Gary Gregory) является главным инженером в Rocket Software, где работает над серверами приложений и интеграцией с устаревшими системами. Еще он соавтор книг издательства Manning: «JUnit in Action» и «Spring Batch in Action», а также член комитетов по руководству проектом (Project Management Commitee) в различных проектах в Apache Software Foundation: Commons, HttpComponents, Logging Services и Xalan.

Об изображении на обложке

Иллюстрация на обложке книги «Java Persistence API и Hibernate, второе издание» взята из сборника костюмов Османской империи, изданного 1 января 1802 г. Вильямом Миллером (William Miller) в Old Bond Street, Лондон. Титульный лист сборника утерян, поэтому мы не смогли точно определить дату его выхода. В содержании книги изображения описаны на английском и французском языках, и под каждой иллюстрацией приводятся имена двух художников, которые трудились над ней. Не сомневаемся, что все они, несомненно, были бы удивлены, узнав, что их творчество украсит обложку книги по программированию... 200 лет спустя.

Рисунки из сборника костюмов Османской империи, так же как и другие иллюстрации, появляющиеся на наших обложках, возрождают богатство и разнообразие традиций в одежде двухсотлетней давности. Они напоминают о чувствах уединенности и удаленности этого периода – и любого другого исторического периода, кроме нашего гиперкинетического настоящего. Манеры одеваться сильно изменились с тех пор, а своеобразие каждого региона, такое яркое в то время, давно поблекло. Сейчас бывает трудно различить обитателей разных континентов. Возможно, если смотреть на это оптимистично, мы обменяли культурное и визуальное разнообразие на более разнообразную частную жизнь или более разнообразную интеллектуальную и техническую жизнь.

Мы в Manning высоко ценим изобретательность, инициативу и, конечно, радость от компьютерного бизнеса с книжными обложками, основанными на разнообразии жизни в разных регионах два века назад, которое оживает благодаря картинкам из этой коллекции.

Конец ознакомительного фрагмента.
Приобрести книгу можно
в интернет-магазине «Электронный универс»
(e-Univers.ru)