



ОГЛАВЛЕНИЕ

| | |
|---|-----------|
| Предисловие | 14 |
| Благодарности | 15 |
| О книге | 18 |
| Структура книги | 19 |
| Загружаемый исходный код | 20 |
| Соглашения по оформлению исходного кода | 20 |
| Автор в сети | 20 |
| О названии | 21 |
| Об авторах | 21 |
| Об иллюстрации на обложке | 22 |
| | |
| ЧАСТЬ I | |
| Обзор ландшафта EJB | 23 |
| | |
| Глава 1. Что такое EJB 3 | 24 |
| 1.1. Обзор EJB | 25 |
| 1.1.1. EJB как модель компонентов | 26 |
| 1.1.2. Службы компонентов EJB | 26 |
| 1.1.3. Многоуровневые архитектуры и EJB | 28 |
| 1.1.4. Почему стоит выбрать EJB 3? | 32 |
| 1.2. Основы типов EJB | 34 |
| 1.2.1. Сеансовые компоненты | 34 |
| 1.2.2. Компоненты, управляемые сообщениями | 35 |
| 1.3. Связанные спецификации | 35 |
| 1.3.1. Сущности и Java Persistence API | 35 |
| 1.3.2. Контексты и внедрение зависимостей для Java EE | 37 |
| 1.4. Реализации EJB | 37 |
| 1.4.1. Серверы приложений | 38 |
| 1.4.2. EJB Lite | 39 |
| 1.4.3. Встраиваемые контейнеры | 40 |
| 1.4.4. Использование EJB 3 в Tomcat | 40 |
| 1.5. Превосходные инновации | 41 |
| 1.5.1. Пример «Hello User» | 41 |
| 1.5.2. Аннотации и XML | 42 |
| 1.5.3. Значения по умолчанию и явные настройки | 43 |
| 1.5.4. Внедрение зависимостей и поиск в JNDI | 44 |
| 1.5.5. CDI и механизм внедрения в EJB | 45 |
| 1.5.6. Тестируемость компонентов POJO | 45 |

| | |
|--|-----------|
| 1.6. Новшества в EJB 3.2 | 46 |
| 1.6.1. Поддержка EJB 2 теперь является необязательной..... | 46 |
| 1.6.2. Усовершенствования в компонентах, управляемых сообщениями | 46 |
| 1.6.3. Усовершенствования в сеансовых компонентах с сохранением состояния | 47 |
| 1.6.4. Упрощение локальных интерфейсов компонентов без сохранения состояния | 48 |
| 1.6.5. Усовершенствования в TimerService API..... | 49 |
| 1.6.6. Усовершенствования в EJBContainer API..... | 49 |
| 1.6.7. Группы EJB API..... | 49 |
| 1.7. В заключение..... | 50 |
| Глава 2. Первая проба EJB..... | 51 |
| 2.1. Введение в приложение ActionBazaar | 52 |
| 2.1.1. Архитектура | 52 |
| 2.1.2. Решение на основе EJB 3 | 54 |
| 2.2. Реализация прикладной логики с применением EJB 3 | 55 |
| 2.2.1. Использование сеансовых компонентов без сохранения состояния ... | 56 |
| 2.2.2. Использование сеансовых компонентов с сохранением состояния | 58 |
| 2.2.3. Модульное тестирование компонентов EJB 3 | 63 |
| 2.3. Использование CDI с компонентами EJB 3 | 64 |
| 2.3.1. Использование CDI с JSF 2 и EJB 3..... | 65 |
| 2.3.2. Использование CDI с EJB 3 и JPA 2..... | 68 |
| 2.4. Использование JPA 2 с EJB 3 | 70 |
| 2.4.1. Отображение сущностей JPA 2 в базу данных | 71 |
| 2.4.2. Использование EntityManager | 72 |
| 2.5. В заключение..... | 74 |
| ЧАСТЬ II | |
| Компоненты EJB..... | 75 |
| Глава 3. Реализация прикладной логики с помощью сеансовых компонентов..... | 76 |
| 3.1. Знакомство с сеансовыми компонентами..... | 77 |
| 3.1.1. Когда следует использовать сеансовые компоненты | 78 |
| 3.1.2. Состояние компонента и типы сеансовых компонентов | 80 |
| 3.2. Сеансовые компоненты без сохранения состояния | 83 |
| 3.2.1. Когда следует использовать сеансовые компоненты без сохранения состояния | 83 |
| 3.2.2. Организация компонентов в пулы | 84 |
| 3.2.3. Пример BidService | 86 |
| 3.2.4. Применение аннотации @Stateless | 89 |
| 3.2.5. Прикладные интерфейсы компонентов..... | 90 |
| 3.2.6. События жизненного цикла..... | 93 |
| 3.2.7. Эффективное использование сеансовых компонентов без сохранения состояния | 96 |
| 3.3. Сеансовые компоненты с сохранением состояния..... | 97 |
| 3.3.1. Когда следует использовать сеансовые компоненты с сохранением состояния..... | 98 |

| | |
|--|-----|
| 3.3.2. Пассивация компонентов..... | 99 |
| 3.3.3. Сеансовые компоненты с сохранением состояния в кластере | 100 |
| 3.3.4. Пример реализации создания учетной записи..... | 100 |
| 3.3.5. Применение аннотации @Stateful | 104 |
| 3.3.6. Прикладные интерфейсы компонентов..... | 105 |
| 3.3.7. События жизненного цикла..... | 105 |
| 3.3.8. Эффективное использование сеансовых компонентов с сохранением состояния..... | 107 |
| 3.4. Сеансовые компоненты-одиночки | 109 |
| 3.4.1. Когда следует использовать сеансовые компоненты-одиночки | 110 |
| 3.4.2. Пример реализации «товара дня» в ActionBazaar | 111 |
| 3.4.3. Применение аннотации @Singleton | 113 |
| 3.4.4. Управление конкуренцией в компоненте-одиночке..... | 114 |
| 3.4.5. Прикладной интерфейс компонента | 117 |
| 3.4.6. События жизненного цикла..... | 118 |
| 3.4.7. Аннотация @Startup | 119 |
| 3.4.8. Эффективное использование сеансовых компонентов-одиночек | 120 |
| 3.5. Асинхронные сеансовые компоненты | 122 |
| 3.5.1. Основы асинхронного вызова | 122 |
| 3.5.2. Когда следует использовать асинхронные сеансовые компоненты..... | 123 |
| 3.5.3. Пример компонента ProcessOrder..... | 124 |
| 3.5.4. Применение аннотации @Asynchronous | 126 |
| 3.5.5. Применение интерфейса Future..... | 127 |
| 3.5.6. Эффективное использование асинхронных сеансовых компонентов | 127 |
| 3.6. В заключение..... | 128 |

Глава 4. Обмен сообщениями и разработка компонентов MDB 130

| | |
|--|-----|
| 4.1. Концепции обмена сообщениями | 131 |
| 4.1.1. Промежуточное ПО передачи сообщений | 131 |
| 4.1.2. Обмен сообщениями в ActionBazaar | 132 |
| 4.1.3. Модели обмена сообщениями | 134 |
| 4.2. Введение в JMS | 136 |
| 4.2.1. Интерфейс JMS Message | 138 |
| 4.3. Использование компонентов MDB | 140 |
| 4.3.1. Когда следует использовать обмен сообщениями и компоненты MDB..... | 141 |
| 4.3.2. Почему следует использовать MDB?..... | 141 |
| 4.3.3. Разработка потребителя сообщений с применением MDB..... | 143 |
| 4.3.4. Применение аннотации @MessageDriven | 144 |
| 4.3.5. Реализация интерфейса MessageListener | 145 |
| 4.3.6. Использование параметра ActivationConfigProperty | 146 |
| 4.3.7. События жизненного цикла..... | 149 |
| 4.3.8. Отправка сообщений JMS из компонентов MDB | 151 |
| 4.3.9. Управление транзакциями MDB | 152 |
| 4.4. Приемы использования компонентов MDB | 153 |
| 4.5. В заключение..... | 155 |

| | |
|---|------------|
| Глава 5. Контекст EJB времени выполнения, внедрение зависимостей и сквозная логика | 157 |
| 5.1. Контекст EJB | 157 |
| 5.1.1. Основы контекста EJB..... | 158 |
| 5.1.2. Интерфейсы контекста EJB..... | 159 |
| 5.1.3. Доступ к контейнеру через контекст EJB | 160 |
| 5.2. Использование EJB DI и JNDI | 161 |
| 5.2.1. Пример использования JNDI в EJB..... | 162 |
| 5.2.2. Как присваиваются имена компонентам EJB..... | 166 |
| 5.2.3. Внедрение зависимостей с применением @EJB | 169 |
| 5.2.4. Когда следует использовать внедрение зависимостей EJB..... | 170 |
| 5.2.5. Аннотация @EJB в действии..... | 171 |
| 5.2.6. Внедрение ресурсов с помощью аннотации @Resource | 173 |
| 5.2.7. Когда следует использовать внедрение ресурсов..... | 175 |
| 5.2.8. Аннотация @Resource в действии | 175 |
| 5.2.9. Поиск ресурсов и компонентов EJB в JNDI | 178 |
| 5.2.10. Когда следует использовать поиск в JNDI | 180 |
| 5.2.11. Контейнеры клиентских приложений | 180 |
| 5.2.12. Встраиваемые контейнеры | 181 |
| 5.2.13. Эффективный поиск и внедрение компонентов EJB | 183 |
| 5.2.14. Механизмы внедрения EJB и CDI | 184 |
| 5.3. AOP в мире EJB: интерцепторы..... | 185 |
| 5.3.1. Что такое AOP? | 185 |
| 5.3.2. Основы интерцепторов..... | 186 |
| 5.3.3. Когда следует использовать интерцепторы | 187 |
| 5.3.4. Порядок реализации интерцепторов | 187 |
| 5.3.5. Определение интерцепторов..... | 188 |
| 5.3.6. Интерцепторы в действии..... | 192 |
| 5.3.7. Эффективное использование интерцепторов | 198 |
| 5.3.8. Интерцепторы CDI и EJB | 199 |
| 5.4. В заключение..... | 205 |
| Глава 6. Транзакции и безопасность | 206 |
| 6.1. Знакомство с транзакциями | 207 |
| 6.1.1. Основы транзакций..... | 208 |
| 6.1.2. Транзакции в Java | 210 |
| 6.1.3. Транзакции в EJB | 212 |
| 6.1.4. Когда следует использовать транзакции | 214 |
| 6.1.5. Как реализованы транзакции EJB | 215 |
| 6.1.6. Двухфазное подтверждение | 217 |
| 6.1.7. Производительность JTA | 218 |
| 6.2. Транзакции, управляемые контейнером | 219 |
| 6.2.1. Досрочное оформление заказов с применением модели CMT..... | 219 |
| 6.2.2. Аннотация @TransactionManagement..... | 220 |
| 6.2.3. Аннотация @TransactionAttribute..... | 221 |
| 6.2.4. Откат транзакций в модели CMT | 224 |
| 6.2.5. Транзакции и обработка исключений | 226 |
| 6.2.6. Синхронизация с сеансом | 228 |
| 6.2.7. Эффективное использование модели CMT | 228 |

| | |
|--|------------|
| 6.3. Транзакции, управляемые компонентами | 229 |
| 6.3.1. Досрочное оформление заказов с применением модели BMT | 230 |
| 6.3.2. Получение экземпляра UserTransaction | 231 |
| 6.3.3. Использование интерфейса UserTransaction | 232 |
| 6.3.4. Эффективное использование модели BMT | 234 |
| 6.4. Безопасность EJB | 234 |
| 6.4.1. Аутентификация и авторизация | 235 |
| 6.4.2. Пользователи, группы и роли | 236 |
| 6.4.3. Как реализована поддержка безопасности в EJB | 237 |
| 6.4.4. Декларативное управление безопасностью в EJB | 241 |
| 6.4.5. Программное управление безопасностью в EJB | 243 |
| 6.4.6. Эффективное использование поддержки безопасности в EJB | 246 |
| 6.5. В заключение | 247 |
| Глава 7. Планирование и таймеры | 249 |
| 7.1. Основы планирования | 250 |
| 7.1.1. Возможности Timer Service | 250 |
| 7.1.2. Таймауты | 253 |
| 7.1.3. Cron | 253 |
| 7.1.4. Интерфейс Timer | 254 |
| 7.1.5. Типы таймеров | 256 |
| 7.2. Декларативные таймеры | 257 |
| 7.2.1. Аннотация @Schedule | 257 |
| 7.2.2. Аннотация @Schedules | 258 |
| 7.2.3. Параметры аннотации @Schedule | 258 |
| 7.2.4. Пример использования декларативных таймеров | 259 |
| 7.2.5. Синтаксис правил в стиле cron | 260 |
| 7.3. Программные таймеры | 263 |
| 7.3.1. Знакомство с программными таймерами | 263 |
| 7.3.2. Пример использования программных таймеров | 265 |
| 7.3.3. Эффективное использование программных таймеров EJB | 267 |
| 7.4. В заключение | 268 |
| Глава 8. Компоненты EJB как веб-службы | 270 |
| 8.1. Что такое «веб-служба»? | 271 |
| 8.1.1. Свойства веб-служб | 271 |
| 8.1.2. Транспорты | 272 |
| 8.1.3. Типы веб-служб | 272 |
| 8.1.4. Java EE API для веб-служб | 273 |
| 8.1.5. Веб-службы и JSF | 274 |
| 8.2. Экспортирование компонентов EJB с использованием SOAP (JAX-WS) | 274 |
| 8.2.1. Основы SOAP | 274 |
| 8.2.2. Когда следует использовать службы SOAP | 279 |
| 8.2.3. Когда следует экспортировать компоненты EJB в виде веб-служб SOAP | 280 |
| 8.2.4. Веб-служба SOAP для ActionBazaar | 281 |
| 8.2.5. Аннотации JAX-WS | 286 |
| 8.2.6. Эффективное использование веб-служб SOAP в EJB | 290 |
| 8.3. Экспортирование компонентов EJB с использованием REST (JAX-RS) | 292 |

| | |
|--|-----|
| 8.3.1. Основы REST | 293 |
| 8.3.2. Когда следует использовать REST/JAX-RS | 296 |
| 8.3.3. Когда следует экспортировать компоненты EJB в виде веб-служб REST | 297 |
| 8.3.4. Веб-служба REST для ActionBazaar | 298 |
| 8.3.5. Аннотации JAX-RS..... | 302 |
| 8.3.6. Эффективное использование веб-служб REST в EJB | 307 |
| 8.4. Выбор между SOAP и REST | 308 |
| 8.5. В заключение..... | 310 |

ЧАСТЬ III

Использование EJB совместно с JPA и CDI 311

Глава 9. Сущности JPA..... 312

| | |
|---|-----|
| 9.1. Введение в JPA | 313 |
| 9.1.1. Несоответствие интерфейсов..... | 313 |
| 9.1.2. Взаимосвязь между EJB 3 и JPA..... | 314 |
| 9.2. Предметное моделирование | 315 |
| 9.2.1. Введение в предметное моделирование..... | 315 |
| 9.2.2. Предметная модель приложения ActionBazaar | 315 |
| 9.3. Реализация объектов предметной области с помощью JPA | 320 |
| 9.3.1. Аннотация @Entity | 320 |
| 9.3.2. Определение таблиц..... | 322 |
| 9.3.3. Отображение свойств в столбцы..... | 325 |
| 9.3.4. Типы представления времени | 330 |
| 9.3.5. Перечисления | 331 |
| 9.3.6. Коллекции | 332 |
| 9.3.7. Определение идентичности сущностей | 334 |
| 9.3.8. Генерирование значений первичных ключей | 339 |
| 9.4. Отношения между сущностями..... | 343 |
| 9.4.1. Отношение «один к одному» | 344 |
| 9.4.2. Отношения «один ко многим» и «многие к одному» | 346 |
| 9.4.3. Отношение «многие ко многим»..... | 349 |
| 9.5. Отображение наследования | 350 |
| 9.5.1. Стратегия единой таблицы | 351 |
| 9.5.2. Стратегия соединения таблиц | 353 |
| 9.5.3. Стратегия отдельных таблиц для каждого класса | 354 |
| 9.6. В заключение..... | 357 |

Глава 10. Управление сущностями358

| | |
|---|-----|
| 10.1. Введение в использование EntityManager | 358 |
| 10.1.1. Интерфейс EntityManager..... | 359 |
| 10.1.2. Жизненный цикл сущностей..... | 361 |
| 10.1.3. Контекст сохранения, области видимости и EntityManager..... | 364 |
| 10.1.4. Использование EntityManager в ActionBazaar | 366 |
| 10.1.5. Внедрение EntityManager..... | 367 |
| 10.1.6. Внедрение EntityManagerFactory | 369 |
| 10.2. Операции с хранилищем..... | 371 |
| 10.2.1. Сохранение сущностей | 372 |

| | |
|---|------------|
| 10.2.2. Извлечение сущностей по ключу | 373 |
| 10.2.3. Изменение сущностей | 379 |
| 10.2.4. Удаление сущностей | 382 |
| 10.3. Запросы сущностей | 384 |
| 10.3.1. Динамические запросы | 385 |
| 10.3.2. Именованные запросы | 385 |
| 10.4. В заключение | 386 |
| Глава 11. JPQL | 387 |
| 11.1. Введение в JPQL | 387 |
| 11.1.1. Типы инструкций | 388 |
| 11.1.2. Предложение FROM | 390 |
| 11.1.3. Инструкция SELECT | 401 |
| 11.1.4. Управление результатами | 404 |
| 11.1.5. Соединение сущностей | 405 |
| 11.1.6. Операции массового удаления и изменения | 408 |
| 11.2. Запросы Criteria | 409 |
| 11.2.1. Метамоделю | 410 |
| 11.2.2. CriteriaBuilder | 413 |
| 11.2.3. CriteriaQuery | 414 |
| 11.2.4. Корень запроса | 415 |
| 11.2.5. Предложение FROM | 419 |
| 11.2.6. Предложение SELECT | 419 |
| 11.3. Низкоуровневые запросы | 422 |
| 11.3.1. Динамические SQL-запросы | 423 |
| 11.3.2. Именованные SQL-запросы | 424 |
| 11.3.3. Хранимые процедуры | 425 |
| 11.4. В заключение | 429 |
| Глава 12. Использование CDI в EJB 3 | 430 |
| 12.1. Введение в CDI | 431 |
| 12.1.1. Службы CDI | 433 |
| 12.1.2. Отношения между CDI и EJB 3 | 436 |
| 12.1.3. Отношения между CDI и JSF 2 | 437 |
| 12.2. Компоненты CDI | 437 |
| 12.2.1. Как пользоваться компонентами CDI | 438 |
| 12.2.2. Именованное компонентов и их разрешение в выражениях EL | 439 |
| 12.2.3. Области видимости компонентов | 440 |
| 12.3. Следующее поколение механизмов внедрения зависимостей | 443 |
| 12.3.1. Внедрение с помощью @Inject | 443 |
| 12.3.2. Фабричные методы | 445 |
| 12.3.3. Квалификаторы | 448 |
| 12.3.4. Методы уничтожения | 449 |
| 12.3.5. Определение альтернатив | 450 |
| 12.4. Интерцепторы и декораторы | 453 |
| 12.4.1. Привязка интерцепторов | 453 |
| 12.4.2. Декораторы | 456 |
| 12.5. Стереотипы | 457 |
| 12.6. Внедрение событий | 459 |

| | |
|---|-----|
| 12.7. Диалоги | 461 |
| 12.8. Эффективное использование CDI в EJB 3 | 467 |
| 12.9. В заключение | 469 |

ЧАСТЬ IV

Ввод EJB в действие471

Глава 13. Упаковка приложений EJB 3472

| | |
|---|-----|
| 13.1. Упаковка приложений | 472 |
| 13.1.1. Строение системы модулей Java EE | 475 |
| 13.1.2. Загрузка модулей Java EE | 476 |
| 13.2. Загрузка классов | 478 |
| 13.2.1. Основы загрузки классов | 478 |
| 13.2.2. Загрузка классов в приложениях Java EE | 478 |
| 13.2.3. Зависимости между модулями Java EE | 481 |
| 13.3. Упаковка сеансовых компонентов и компонентов, управляемых сообщениями | 483 |
| 13.3.1. Упаковка EJB-JAR | 483 |
| 13.3.2. Упаковка компонентов EJB в модуль WAR | 485 |
| 13.3.3. XML против аннотаций | 488 |
| 13.3.4. Переопределение настроек, указанных в аннотациях | 492 |
| 13.3.5. Определение интерцепторов по умолчанию | 493 |
| 13.4. Упаковка сущностей JPA | 494 |
| 13.4.1. Модуль доступа к хранимым данным | 494 |
| 13.4.2. Описание модуля доступа к хранимым данным в persistence.xml | 496 |
| 13.5. Упаковка компонентов CDI | 498 |
| 13.5.1. Модули CDI | 498 |
| 13.5.2. Дескриптор развертывания beans.xml | 499 |
| 13.5.3. Атрибут bean-discovery-mode | 500 |
| 13.6. Эффективные приемы и типичные проблемы развертывания | 501 |
| 13.6.1. Эффективные приемы упаковки и развертывания | 501 |
| 13.6.2. Решение типичных проблем развертывания | 503 |
| 13.7. В заключение | 504 |

Глава 14. Использование веб-сокетов с EJB 3505

| | |
|---|-----|
| 14.1. Ограничения схемы взаимодействий «запрос/ответ» | 505 |
| 14.2. Введение в веб-сокеты | 507 |
| 14.2.1. Основы веб-сокетов | 507 |
| 14.2.2. Веб-сокеты и AJAX | 511 |
| 14.2.3. Веб-сокеты и Comet | 513 |
| 14.3. Веб-сокеты и Java EE | 515 |
| 14.3.1. Конечные точки веб-сокетов | 516 |
| 14.3.2. Интерфейс Session | 517 |
| 14.3.3. Кодеры и декодеры | 520 |
| 14.4. Веб-сокеты в приложении ActionBazaar | 523 |
| 14.4.1. Использование программных конечных точек | 526 |
| 14.4.2. Использование аннотированных конечных точек | 530 |
| 14.5. Эффективное использование веб-сокетов | 537 |
| 14.6. В заключение | 539 |

| | |
|--|------------|
| Глава 15. Тестирование компонентов EJB | 541 |
| 15.1. Введение в тестирование | 541 |
| 15.1.1. Стратегии тестирования | 542 |
| 15.2. Модульное тестирование компонентов EJB | 544 |
| 15.3. Интеграционное тестирование с использованием EJBContainer | 548 |
| 15.3.1. Настройка проекта | 549 |
| 15.3.2. Интеграционный тест | 552 |
| 15.4. Интеграционное тестирование с применением Arquillian | 555 |
| 15.4.1. Настройка проекта | 556 |
| 15.4.2. Интеграционный тест | 560 |
| 15.5. Приемы эффективного тестирования | 563 |
| 15.6. В заключение | 565 |
| Приложение А. Дескриптор развертывания, справочник | 566 |
| A.1. ejb-jar.xml | 566 |
| A.1.1. <module-name> | 567 |
| A.1.2. <enterprise-beans> | 567 |
| A.1.3. Интерцепторы | 571 |
| A.1.4. <assembly-descriptor> | 571 |
| Приложение В. Введение в Java EE 7 SDK | 576 |
| В.1. Установка Java EE 7 SDK | 576 |
| В.2. GlassFish Administration Console | 581 |
| В.3. Запуск и остановка GlassFish | 584 |
| В.4. Запуск приложения «Hello World» | 586 |
| Приложение С. Сертификационные экзамены разработчика для EJB 3..... | 590 |
| С.1. Начало процесса сертификации | 591 |
| С.2. Порядок прохождения сертификационных испытаний для разработчиков EJB 3 | 593 |
| С.3. Знания, необходимые для прохождения испытаний | 595 |
| С.4. Подготовка к испытаниям | 597 |
| С.5. Сертификационные испытания | 598 |
| Предметный указатель | 600 |



ПРЕДИСЛОВИЕ

Первоначально технология EJB появилась под влиянием идей, заложенных в технологии распределенных вычислений, таких как CORBA, и предназначалась для поддержки масштабирования серверных приложений. Во время бума доткомов вокруг EJB и J2EE была поднята большая рекламная шумиха.

Первоначальной целью EJB было предоставить простую альтернативу технологии CORBA на основе стандартной инфраструктуры разработки и компонентов многократного пользования. К моменту выхода EJB 2 стало очевидно, что эта инфраструктура превратилась в новый стандарт разработки серверных приложений. Она предоставляет промышленным разработчикам все необходимое – поддержку удаленных взаимодействий, механизмы управления транзакциями, средства обеспечения безопасности, обработки и хранения информации, и веб-службы – но все это были тяжеловесные механизмы, требующие от разработчиков основное внимание уделять особенностям взаимодействий с самой инфраструктурой, чем реализации бизнес-логики приложений. Из-за включения в EJB все новых особенностей, основатели этой инфраструктуры оказались не в состоянии справиться со все возрастающей сложностью.

С появлением недовольств, вызванных ограничениями в EJB 2, стали появляться новые инструменты с открытым исходным кодом. Эти инструменты являются ярким признаком растущего недовольства сложностью Java EE. Однако, несмотря на благие намерения, эти инструменты еще больше усложнили разработку промышленных приложений, так как они отклонялись от стандартов, лежащих в основе сервера приложений, где эти инструменты должны использоваться. В этот период была запущен процесс Java Community Process (JCP) и сформирована экспертная группа для выполнения работ по упрощению разработки на основе Java EE. Это была единственная причина начала разработки Java EE 5 и EJB 3.

Для технологии с такой широкой областью применения, изменения в EJB 3 стали просто оглушительными. EJB 3 благополучно объединила в себе инновационные приемы, существенно упрощающие разработку компонентов. В число этих приемов входят: использование аннотаций, метапрограммирование, внедрение зависимостей, AspectJ-подобные интерцепторы и интеллектуальное использование значений по умолчанию. Произошел отказ от тяжеловесной модели программирования на основе наследования в пользу более легковесной модели на основе простых объектов Java (Plain Old Java Object, POJO), а подробные описания настроек на языке XML ушли с пути разработчика.

Еще более существенными оказались изменения в модели хранения данных. В EJB 3 произошел отказ от несовершенной модели Entity Beans, использовавшейся в EJB 2, в пользу легковесного программного интерфейса Java Persistence API (JPA). В отличие от Entity Beans, интерфейс JPA не основывается на контейнерах. Он больше похож на инструменты объектно-реляционного отображения (Object Relational Mapping, ORM), созданные сообществом в ответ на сложность Entity Beans. JPA может использоваться и внутри, и за пределами сервера Java Enterprise, и в настоящее время де-факто считается стандартом доступа к хранимым данным для Java. Его язык запросов Java Persistence Query Language (JPQL) стандартизует объектно-реляционные запросы, а также поддерживает запросы на языке SQL.

Изменения в EJB 3 были благосклонно восприняты сообществом Java. Упрощение спецификации привело к ее широкому распространению среди новых проектов. Все больше компаний повторно рассматривают возможность применения прежде «неудачной» технологии EJB и склоняются к положительному решению. С выходом версии EJB 3.2 число положительных решений выросло еще больше. Спецификация EJB 3.2 сделала поддержку EJB 2 необязательной, что обусловило скорый закат старой технологии и дальнейший рост числа инноваций в EJB 3. Спецификация EJB 3.2 также внесла существенные расширения в компоненты, управляемые событиями (Message-Driven Bean, MDB), значительно упростив реализацию обмена сообщениями. Версия EJB 3.2 улучшила поддержку сеансовых компонентов с сохранением состояния (stateful session bean) и их локальных интерфейсов, а также внесла важнейшие улучшения в службы таймеров. Все это и многое другое ждет вас в EJB 3.2.

Так как EJB опирается на POJO, любой разработчик на Java легко сможет превратиться в EJB-разработчика. Простые аннотации приносят в прикладной код надежные транзакции, механизмы поддержки безопасности и возможность выступать в качестве веб-служб, с целью упростить взаимодействия программных продуктов в пределах компании. Мы старались сделать нашу книгу отличной от других по EJB, наполнив ее практическими примерами, описанием эффективных приемов, а также рекомендациями по улучшению производительности. Мы особенно отметим новинки, появившиеся в спецификации EJB версии 3.2, дающие дополнительные инструменты для использования в процессе разработки. Мы надеемся, что это второе издание книги поможет вам быстро понять, как эффективнее использовать EJB 3 в вашем следующем промышленном приложении.

Благодарности

Работа над книгой требует усилий многих людей и порой бывает сложно перечислить всех, кто помогал в ее создании. Прежде всего мы хотели бы поблагодарить всех сотрудников издательства Manning за их одобрение и поддержку, и особенно владельца издательства Марджана Бейса (Marjan Vasce), совладельца Майкла Стефенса (Michael Stephens) и нашего редактора Нермину Миллер (Nermina Miller). Нам также хотелось бы выразить благодарность другим сотрудникам издательства Manning, участвовавшим в работе над проектом на разных этапах: редактору

рецензенту Оливии Бус (Olivia Booth); редактору проекта Джоди Аллен (Jodie Allen); менеджеру по развитию Маурину Спенсеру (Maureen Spencer); техническому корректору Дипаку Вохра (Deepak Vohra), выполнившему окончательную правку книги непосредственно перед выпуском в печать; Линде Ректенволд (Linda Recktenwald) и Мелоди Долаб (Melody Dolab), выполнившими литературное редактирование и корректуру; и верстальщику Денису Далиннику (Dennis Dalinnik), превратившему наши документы Word в настоящую книгу! Спасибо также всем, кто незримо участвовал в подготовке книги к публикации.

У нас было много рецензентов, потративших свое драгоценное время на чтение рукописей на разных этапах их подготовки, и их отзывы помогли значительно повысить качество книги. Мы очень признательны вам: Артур Новак (Artur Nowak), Азиз Рахман (Aziz Rahman), Боб Касацца (Bob Casazza), Кристоф Мартини (Christophe Martini), Дэвид Стронг (David Strong), Джит Марвах (Jeet Marwah), Джон Гриффин (John Griffin), Джонас Банди (Jonas Bandi), Джозеф Лехнер (Josef Lehner), Юрген де Коммер (Jürgen De Commer), Каран Малхи (Karan Malhi), Халид Муктар (Khalid Muktar), Корай Гюсю (Koray Güclü), Луис Пенья (Luis Peña), Матиас Агезли (Matthias Agethle), Палак Матхур (Palak Mathur), Павел Розенблюм (Pavel Rozenblioum), Рик Вагнер (Rick Wagner), Сумит Пал (Sumit Pal), Виктор Агилар (Victor Aguilar), Веллингтон Пинхейро (Wellington Pinheiro) и Зороджай Макая (Zorodzay Mukuya).

Наконец, спасибо редакторам из Manning Early Access Program (MEAP), вычитывавшим наши главы по мере их готовности и присылавшим комментарии и исправления на форум книги. Ваша помощь сделала эту книгу намного лучше.

Дебу Панда (Debu Panda)

Я хочу поблагодарить мою супругу Ренуку (Renuka), за ее одобрение и поддержку, за то, что терпела, когда я поздно ложился и рано вставал, и когда все выходные проводил в работе над первым изданием книги. Я также хочу сказать спасибо моим детям, Ништхе (Nistha) и Нишиту (Nisheet), которым приходилось делить их папу с компьютером.

Большое спасибо моим соавторам, Резу Рахману (Reza Rahman), Райану Купраку (Ryan Cuprak) и Майклу Ремижану (Michael Remijan), которые упорно трудились над вторым изданием этой книги.

Реза Рахман (Reza Rahman)

Даже путь в тысячу ли начинается с первого шага.

– Лао-Цзы

Не думаю, что приступая к работе над первым изданием этой книги, кто-то из авторов представлял себе, насколько успешной станет книга и как это отразится на нас самих. Сегодня, оглядываясь назад, я могу сказать, что книга добилась ошеломительного успеха, а ее создание стало первым шагом на долгом пути, длиной последние несколько лет. Должен признаться, что продолжаю получать удовольствие от каждой минуты на этом пути. После завершения работы над первым изданием

я стал все больше и больше участвовать в жизни сообщества Java. Я вошел в состав различных экспертных групп по Java EE, включая экспертную группу EJB, и получил уникальную возможность написать свободно распространяемый контейнер EJB практически с нуля, а теперь еще оказался в центре команды SunOracle, занимающейся популяризацией и продвижением Java EE.

В результате всего этого у меня практически не осталось свободного времени, которого было в достатке, когда я работал над первым изданием. Это основная причина, почему нам пришлось пропустить издание книги, посвященное Java EE 6 и EJB 3.1. И, как мне кажется, что к лучшему, потому что Java EE 7 оказалась еще более мощной и привлекательной платформой, что мы и попытались продемонстрировать в этом издании книги. Я очень благодарен Майклу и Райану, что приняли решение и сыграли важную роль в создании достойного второго издания. Я также благодарен многим членам сообщества Java EE, с которыми я имел честь работать. Наконец, я благодарен моей жене Николь (Nicole) и дочери Зехре (Zehra), что безоговорочно позволили мне поддаться моей страсти. Итак, мой путь продолжается.

Райан Купрак (Ryan Cuprak)

Работа над этой книгой стала бы для меня невозможной без поддержки семьи и друзей. Я особенно хотел бы поблагодарить Эльзу (Elsa), любовь всей моей жизни, которая поддерживала и подбадривала меня в течение длинного и трудного периода, когда я проводил ночи напролет, сгорбившись за компьютером. Наконец, я хочу сказать спасибо Резу, что предложил мне поработать над этим проектом и подвигнул меня обратить внимание на Java EE много лет тому назад.

Майкл Ремижан (Michael Remijan)

Моя красавица жена Келли (Kelly) и моя дочка София (Sophia) – первые, кому я хочу сказать спасибо. Они разделили со мной это приключение и без их поддержки ранними утрами, поздними вечерами и на выходных я не смог бы спокойно заниматься исследованиями и работой над этим проектом. Келли – мой лучший друг, она подбадривала меня во всех моих начинаниях, и это добавляет мне сил и уверенности. Я люблю ее очень нежно. И я счастлив, что у меня такая замечательная семья.

Мои соавторы – Дебу, Райан и Реза – следующие в моем списке: большое спасибо вам. Эта книга является плодом коллективных усилий. EJB – обширная технология, обладающая огромным множеством возможностей; отсюда и такой размер книги. Написать такую книгу в одиночку практически невозможно, поэтому наше сотрудничество сыграло решающую роль в ее создании. Для меня было большой честью работать с такими талантливыми коллегами.

Наконец, спасибо всем сотрудникам издательства Manning, потративших бесчисленные часы на обзоры этой книги и поддерживавшими процесс ее подготовки, особенно хотелось бы отметить Маурину Спенсера (Maureen Spencer) и Джоди Аллен (Jodie Allen). Спасибо также Кристине Рудлофф (Christina Rudloff), пригласившей меня в проект – без этого я не стал бы одним из соавторов книги.

О книге

EJB 3 позволяет взглянуть на разработку серверных Java-приложений с неожиданной стороны. Поэтому мы старались наполнить эту книгу о EJB неожиданными для вас фактами.

Большинство книг о Java-программировании на стороне сервера чрезвычайно тяжелы в чтении – они наполнены теорией, слегка одобренной нравоучениями, и предназначаются для опытных разработчиков. Хотя мы легко укладываемся в стереотип гиков и не обладаем талантами комедиантов или конференсье, мы все же попытались добавить живых красок в нашу книгу, чтобы сделать ее простой и практичной, насколько это возможно. Мы избрали дружественный и неформальный тон общения с читателем и постарались уделить основное внимание обсуждению примеров решения задач, с которыми многие из нас сталкиваются в своей повседневной работе. В большинстве случаев мы сначала рассказываем о задаче, которую требуется решить; показываем код, решающий ее с помощью EJB 3; а затем исследуем особенности этой технологии, использованные в коде.

Мы будем рассказывать о теоретических предпосылках, только когда это действительно необходимо. Мы старались избегать теоретических рассуждений ради самих рассуждений, чтобы сделать наше с вами общение как можно более живым. Цель этой книги – помочь вам быстро и эффективно освоить EJB 3, а не служить исчерпывающим справочником. Мы не будем рассматривать редко используемые возможности. Зато мы глубоко будем проникать в наиболее полезные особенности EJB 3 и родственные технологии. Мы будем обсуждать разные варианты, чтобы вы могли делать осознанный выбор, и расскажем о наиболее распространенных ловушках, а также покажем испытанные приемы.

Коль скоро вы выбрали эту книгу, весьма маловероятно, что вы новичок в Java. Мы полагаем, что у вас уже есть опыт программирования на этом языке, и, возможно, вам доводилось заниматься разработкой веб-приложений с применением таких технологий, как JSF, Struts, JSP или сервлеты (servlets). Мы также предполагаем, что вы знакомы с технологиями баз данных, такими как JDBC, и имеете хотя бы минимальное представление о языке SQL. Для работы с этой книгой вам не понадобится опыт использования EJB 2.x; EJB 3 полностью отличается от нее. Мы не рассчитываем, что вы знакомы с какими-либо технологиями Java EE, на которых основывается технология EJB, такими как Java Naming and Directory Interface (JNDI), Java Remote Method Invocation (RMI) и Java Messaging Service (JMS). Более того, мы будем исходить из предположения, что вы не знакомы с понятиями промежуточного слоя, такими как удаленные взаимодействия (remoting), организация пулов (pooling), конкурентное (или параллельное) программирование, безопасность и распределенные транзакции. Эта книга идеально подходит для разработчиков на Java, имеющих двух-трех летний опыт работы, интересующихся технологией EJB 3.

Вы можете найти в этой книге одно важное отличие от других подобных ей книг. EJB – это серверная связующая (или промежуточная) технология. Это означает, что она существует не в вакууме и должна интегрироваться с другими технологи-

ями для выполнения своей миссии. На протяжении всей книги мы будем рассказывать о том, как EJB 3 интегрируется с такими технологиями, как JNDI, JMS, JSF, JSP, Servlets, AJAX и даже с клиентами Java SE на основе Swing.

Эта книга рассказывает о EJB 3, как о стандарте, а не как о конкретной технологии сервера приложений. По этой причине мы не будем пытаться обсуждать какую-то определенную реализацию – все примеры программного кода в этой книге построены так, что смогут выполняться в любом контейнере EJB 3 и взаимодействовать с любым провайдером доступа к хранимым данным. На веб-сайте поддержки книги, по адресу: www.manning.com/EJB3inActionSecondEdition, вы найдете инструкции, как опробовать примеры в GlassFish и Oracle Application Server 10g. Размещение инструкций по использованию конкретных серверов приложений на веб-сайте вместо книги позволит нам поддерживать их в актуальном состоянии и дополнять новыми сведениями о последних реализациях.

Структура книги

Эта книга делится на четыре части.

В первой части дается общий обзор EJB. В главе 1 вы познакомитесь с EJB 3 и типами EJB, и составите некоторое представление о EJB 3. Здесь также рассказывается об изменениях в версии EJB 3.2. В главе 2 вы впервые попробуете EJB на вкус, реализовав свое первое решение с применением технологии EJB.

Вторая часть охватывает особенности работы с компонентами EJB при реализации прикладной логики. Глава 3 описывает тонкости сеансовых компонентов (session beans) и знакомит с наиболее удачными приемами. Глава 4 представляет собой краткое введение в организацию обмена сообщениями, JMS и подробно рассказывает о MDB. Глава 5 охватывает дополнительные темы, такие как контекст EJB, JNDI, ресурсы и внедрение EJB, интерцепторы AOP и контейнер клиентского приложения. Глава 6 обсуждает транзакции и безопасность. Глава 7 знакомит с таймерами и новыми возможностями реализации отложенных вычислений. Глава 8 демонстрирует возможность экспорта прикладной логики EJB в виде веб-служб SOAP и RESTful.

В третьей части подробно рассказывается о взаимоотношениях EJB 3 с JPA и CDI. В главе 9 дается введение в предметное моделирование и особенности отображения сущностей JPA в предметную область. Глава 10 охватывает управление сущностями JPA через CRUD-операции. Глава 11 рассказывает о языке JPQL и более подробно рассказывает о приемах извлечения данных. Глава 12 служит введением в технологию CDI и рассказывает, как она дополняет EJB.

В четвертой части содержатся рекомендации по внедрению EJB 3 в работу на предприятии. Глава 13 обсуждает подготовку компонентов EJB к развертыванию на сервере. Глава 14 знакомит с веб-сокетами, их взаимосвязях с компонентами EJB, а также с возможностью асинхронного выполнения прикладной логики с применением утилит EJB. Глава 15 охватывает вопросы модульного и интеграционного тестирования без необходимости развертывания приложений на действующем сервере.

Книга имеет три приложения. Приложение А – это справочник по дескриптору развертывания *ejb-jar.xml*. Приложение В содержит пошаговые инструкции по загрузке и установке пакета Java EE 7 SDK, включающего Java SE 7, GlassFish 4 и NetBeans. Приложение С содержит информацию о процессе сертификации по технологии EJB в Oracle и сертификационных экзаменах по EJB.

Загружаемый исходный код

Приложение В содержит пошаговые инструкции по загрузке и установке пакета Java EE 7 SDK. Исходный код примеров для этой книги доступен для загрузки по адресу: <http://code.google.com/p/action-bazaar/>. Здесь вы сможете либо получить копию репозитория Git со всеми примерами, либо загрузить подготовленный ZIP-файл со всеми примерами внутри. Программный код разрабатывался в основном в среде NetBeans, но все примеры собирались с помощью Maven, поэтому они должны сохранить работоспособность и в других средах разработки.

ZIP-файл с исходным кодом также доступен для загрузки на сайте издательства, по адресу: www.manning.com/EJB3inActionSecondEdition.

Соглашения по оформлению исходного кода

Так как обсуждение разных тем в этой книге в значительной степени опирается на примеры, исходному коду было уделено самое пристальное внимание. Большие фрагменты кода представлены в виде отдельных листингов. Весь программный код набран моноширинным шрифтом, как этот фрагмент, для большей наглядности. Все, что находится внутри фрагментов программного кода, например: элементы XML; имена методов, типов данных, пакетов, переменных и прочее, набраны шрифтом Courier. Некоторые элементы программного кода набраны **жирным моноширинным шрифтом** – это сделано, чтобы выделить наиболее важные фрагменты. Также в коде могут содержаться комментарии, описывающие некоторые важные понятия. В некоторых случаях мы сократили код примеров, чтобы сделать его более простым и компактным. В любом случае, полный код примеров можно найти в загружаемых zip-файлах. Дополнительно мы рекомендуем настраивать среду разработки перед прочтением каждой главы.

Автор в сети

Приобретая книгу «EJB 3 в действии, второе издание», вы получаете бесплатный доступ на частный веб-форум издательства Manning Publications, где вы сможете оставлять свои отзывы о книге, задавать вопросы технического характера и получать помощь от авторов и других пользователей. Чтобы получить доступ к форуму и зарегистрироваться на нем, откройте в браузере страницу www.manning.com/EJB3inActionSecondEdition. 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.

О названии

Сочетая в себе введение, краткий обзор и примеры использования, книги из серии «в действии» предназначены для изучения и запоминания. Согласно исследованиям в когнитивистике, люди лучше запоминают вещи, которые они узнают в процессе самостоятельного изучения.

В издательстве Manning нет ученых-когнитивистов, тем не менее, мы уверены, что для надежного усваивания необходимо пройти через стадии исследования, игры и, что интересно, пересказа всего, что было изучено. Люди усваивают новые знания до уровня овладения ими, только после их активного исследования. Люди учатся в действии. Особенность учебников из серии «в действии» в том, что они основываются на примерах. Это побуждает читателя пробовать, играть с новым кодом, и исследовать новые идеи.

Есть и другая, более прозаическая причина выбора такого названия книги: наши читатели – занятые люди. Они используют книги для выполнения работы или для решения проблем. Им нужны книги, позволяющие легко перепрыгнуть с места на место и изучать только то, что они хотят, и только когда они этого хотят. Им нужны книги, помогающие в действии. Книги этой серии создаются для таких читателей.

Об авторах

Дебу Панда (Debu Panda) – опытный руководитель, инженер и лидер сообщества. Он написал более 50 статей о технологиях Enterprise Java, Cloud и SOA, и две книги о промежуточном ПО уровня предприятия (Enterprise middleware). Следуйте за Дебу в Твиттере [@debupanda](#).

Реза Рахман (Reza Rahman) – прежде долгое время работал независимым консультантом, а ныне является официальным пропагандистом Java EE/GlassFish в Oracle. Реза часто выступает в группах пользователей Java и на конференциях по всему миру. Он часто пишет статьи для промышленных журналов, таких как «JavaLobby/DZone» и «TheServerSide». Реза входил в состав экспертных групп, занимавшихся разработкой спецификаций Java EE, EJB и JMS. Реализовал контейнер EJB для Resin, открытого сервера приложений Java EE.

Райан Купрак (Ryan Cuprak) – аналитик в компании Dassault Systèmes (DS), автор руководства «NetBeans Certification Guide», выпущенного издательством McGraw-Hill, и президент группы пользователей «Connecticut Java Users Group» с 2003 года. Удостоен награды JavaOne 2011 Rockstar Presenter. В DS занимается

разработкой средств интеграции данных, обеспечивающих преобразование клиентских данных, а также созданием пользовательских интерфейсов. До поступления на работу в компанию DS он работал в начинающей компании, специализирующейся на распределенных вычислениях, в TurboWorx и в группе Eastman Kodak Molecular Imaging Systems, ныне являющейся частью компании Carestream Health. В TurboWorx служил инженером службы сбыта и занимался предпродажной поддержкой и разработкой программ на языке Java. Купрак получил степень бакалавра информатики и биологии в университете Лойола в Чикаго. Получил в компании Sun сертификат специалиста по NetBeans IDE.

Майкл Ремижан (Michael Remijan) – директор-распорядитель и технический руководитель в больнице VJC Hospital. Начал работать с Java EE в конце 1990-х. Занимался разработкой систем B2C и B2B для коммерции, производства, астрономии, сельского хозяйства, телекоммуникации, национальной обороны и здравоохранения. Получил степень бакалавра информатики и математики в университете штата Иллинойс в Урбана-Шампейн и степень магистра управления научно-техническим развитием в университете города Феникс. Обладает множеством сертификатов компании Sun Microsystems и является автором статей в журналах «Java Developer Journal» и «Javalobby/DZone». Его блог можно найти по адресу: mjremijan.blogspot.com.

Об иллюстрациях на обложке

На обложке книги «EJB 3 в действии» изображена «Русская девушка с мехом». Этот рисунок взят из Французского туристического путеводителя «Encyclopedie des Voyages» (автор J. G. St. Saverur), выпущенного в 1796 году. Путешествия ради удовольствия были сравнительно новым явлением в то время и туристические справочники, такие как этот, были популярны, они позволяли знакомиться с жителями других регионов Франции и других стран, не вставая с кресла.

Многообразие рисунков в путеводителе «Encyclopedie des Voyages» отчетливо демонстрирует уникальные и индивидуальные особенности городов и районов мира, существовавших 200 лет назад. Это было время, когда по одежде можно было отличить двух людей, проживающих в разных регионах, расположенных на расстоянии всего нескольких десятков километров. Туристический справочник позволяет почувствовать изолированность и отдаленность того периода от любого другого исторического периода отличного от нашей гиперподвижной современности.

С тех пор мода изменилась, а региональные различия, такие существенные в те времена, исчезли. Сейчас зачастую сложно отличить жителей разных континентов. Возможно, пытаясь рассматривать это с оптимистической точки зрения, мы обменяли культурное и визуальное разнообразие на более разнообразную личную жизнь. Или более разнообразную и интересную интеллектуальную жизнь и техническую вооруженность.

Мы в издательстве Manning славим изобретательность, предприимчивость и радость компьютерного бизнеса обложками книг, изображающими богатство региональных различий двухвековой давности.