

Содержание

| | |
|--|----|
| Введение | 21 |
| Предисловие | 23 |
| Взлет-посадка | 23 |
| Посадка с помощью DDD | 24 |
| Изображение ландшафта и построение карты полета | 25 |
| Обзор глав | 26 |
| Глава 1. Знакомство с DDD | 26 |
| Глава 2. Предметная область, предметная подобласть И ограниченные контексты | 27 |
| Глава 3. Карты контекстов | 28 |
| Глава 4. Архитектура | 28 |
| Глава 5. Сущности | 28 |
| Глава 6. Объекты-значения | 29 |
| Глава 7. Службы | 29 |
| Глава 8. События предметной области | 29 |
| Глава 9. Модули | 30 |
| Глава 10. Агрегаты | 30 |
| Глава 11. Фабрики | 30 |
| Глава 12. Хранилища | 31 |
| Глава 13. Интеграция ограниченных контекстов | 31 |
| Глава 14. Приложение | 31 |
| Приложение. Агрегаты и источники событий: A+ES | 32 |
| Язык Java и инструменты разработки | 32 |
| Благодарности | 35 |
| Об авторе | 39 |
| Руководство по использованию книги | 40 |
| Общая картина DDD | 41 |
| Стратегическое моделирование | 42 |
| Архитектура | 43 |
| Тактическое моделирование | 44 |
| Глава 1. Знакомство с DDD | 47 |
| Могу ли я применить принципы DDD | 48 |
| Почему необходимо применять подход DDD | 53 |
| Обеспечить бизнес-ценность иногда трудно | 54 |

| | |
|--|-----------|
| Чем может помочь подход DDD | 56 |
| Столкновение со сложностью предметной области | 57 |
| Анемия и потеря памяти | 58 |
| Причины анемии | 62 |
| Как анемия влияет на вашу модель | 64 |
| Как применять DDD | 68 |
| ЕДИНЫЙ ЯЗЫК | 69 |
| Единый, но не универсальный | 73 |
| Бизнес-ценность DDD | 74 |
| 1. Организация получает полезную модель своей предметной области | 75 |
| 2. Вырабатываются точное определение и описание бизнеса | 75 |
| 3. В разработке программного обеспечения принимают участие эксперты в предметной области | 76 |
| 4. Пользователи системы повышают свою квалификацию | 76 |
| 5. Модели имеют четкие границы | 77 |
| 6. Улучшается архитектура предприятия | 77 |
| 7. Применяются гибкие, итеративные и непрерывные методы моделирования | 77 |
| 8. Развертываются новые стратегические и тактические инструменты | 77 |
| Проблемы применения DDD | 78 |
| Обоснование моделирования предметной области | 84 |
| Подход DDD не сложный | 87 |
| Правдоподобный вымысел | 88 |
| Резюме | 92 |
| Глава 2. Предметные области, подобласти и ограниченные контексты | 93 |
| Общая картина | 94 |
| ПРЕДМЕТНЫЕ ПОДОБЛАСТИ И ОГРАНИЧЕННЫЕ КОНТЕКСТЫ в действии | 95 |
| Внимание на СМЫСЛОВОЕ ЯДРО | 101 |
| Чем объяснить невероятную важность стратегического проектирования | 104 |
| Реальные предметные области и подобласти | 108 |
| Осмысление ограниченных контекстов | 114 |
| Пространство не только для модели | 119 |
| Размер ограниченных контекстов | 121 |
| Согласования с техническими компонентами | 124 |
| Примеры контекстов | 126 |
| Контекст сотрудничества | 127 |
| Контекст идентификации и доступа | 134 |
| Контекст управления гибким проектированием | 136 |
| Резюме | 139 |

| | |
|--|-----|
| Глава 3. Карты контекстов | 141 |
| Важность КАРТ КОНТЕКСТОВ | 142 |
| Рисование КАРТ КОНТЕКСТОВ | 144 |
| Проектные и организационные отношения | 146 |
| Карты трех контекстов | 149 |
| Контекст сотрудничества | 157 |
| Контекст управления гибким проектированием | 160 |
| Резюме | 168 |
| Глава 4. Архитектура | 169 |
| Интервью с успешным директором по информатизации | 171 |
| Уровни | 176 |
| Принцип инверсии зависимостей | 180 |
| Гексагональная архитектура, или Архитектура портов и адаптеров | 183 |
| Сервис-ориентированная архитектура | 188 |
| Передача репрезентативного состояния — REST | 192 |
| Автор: Стефан Тильков (Stefan Tilkov) | 192 |
| REST как архитектурный стиль | 192 |
| Ключевые аспекты HTTP-сервера RESTful | 193 |
| Ключевые аспекты HTTP-клиента RESTful | 195 |
| Принципы REST и DDD | 195 |
| Почему REST? | 197 |
| Разделение ответственности на команды и запросы, или Принцип CQRS | 197 |
| Исследование областей шаблона CQRS | 200 |
| Событийно-ориентированная архитектура | 208 |
| ДЛИТЕЛЬНЫЕ ПРОЦЕССЫ, или САГИ | 215 |
| ПОРОЖДЕНИЕ СОБЫТИЙ | 222 |
| ФАБРИКА ДАННЫХ и РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛЕНИЯ | 226 |
| Репликация данных | 228 |
| Событийно-ориентированные фабрики и события предметной области | 228 |
| Непрерывные запросы | 229 |
| Распределенная обработка | 230 |
| Резюме | 232 |

| | |
|--|-----|
| Глава 5. Сущности | 233 |
| Зачем нужны СУЩНОСТИ | 234 |
| Уникальный идентификатор | 235 |
| Идентификатор вводится пользователем | 237 |
| Идентификатор генерируется приложением | 238 |
| Идентификатор генерируется механизмом постоянного хранения | 242 |
| Идентификатор присваивается другим ограниченным контекстом | 246 |
| Если время генерирования идентификатора имеет значение | 248 |
| Суррогатный идентификатор | 250 |
| Постоянство идентификатора | 253 |
| Выявление сущностей и их внутренних характеристик | 255 |
| Выявление сущностей и свойств | 256 |
| Поиски важных функций | 261 |
| Роли и обязанности | 265 |
| Конструирование | 271 |
| Проверка корректности | 273 |
| Наблюдение за изменениями | 282 |
| Резюме | 283 |
| Глава 6. Объекты-значения | 285 |
| Характеристики значений | 287 |
| Измерение, количественная оценка или описание | 288 |
| Неизменяемость | 288 |
| Концептуальное целое | 289 |
| Заменяемость | 293 |
| Равенство значений | 294 |
| Функция без побочных эффектов | 295 |
| Интеграция в стиле минимализма | 299 |
| Стандартные типы, выраженные в виде значений | 301 |
| Тестирование ОБЪЕКТОВ-ЗНАЧЕНИЙ | 307 |
| Реализация | 311 |
| Хранение ОБЪЕКТОВ-ЗНАЧЕНИЙ | 317 |
| Предотвращение чрезмерного влияния утечки информации из модели данных | 318 |
| Механизм ORM и отдельные объекты-значения | 320 |
| Механизм ORM и многочисленные значения, сериализованные в одном столбце | 323 |
| Механизм ORM и многочисленные значения на основе сущности из базы данных | 324 |
| Механизм ORM и многочисленные значения на основе объединенной таблицы | 329 |
| ORM и ОБЪЕКТЫ, в которых состояние представлено перечислением | 331 |
| Резюме | 333 |

| | |
|---|-----|
| Глава 7. Службы | 335 |
| Чем является служба предметной области (но сначала о том, чем она не является) | 337 |
| Убедитесь, что вам необходима СЛУЖБА | 339 |
| Моделирование службы в предметной области | 343 |
| Необходим ли ВЫДЕЛЕННЫЙ ИНТЕРФЕЙС | 346 |
| Процесс вычисления | 348 |
| Службы преобразования | 351 |
| Использование микроуровней служб предметной области | 352 |
| Службы тестирования | 352 |
| Резюме | 355 |
| Глава 8. События предметной области | 357 |
| Когда и почему происходят СОБЫТИЯ ПРЕДМЕТНОЙ ОБЛАСТИ | 358 |
| Моделирование событий | 361 |
| Характеристики АГРЕГАТОВ | 366 |
| Идентичность | 367 |
| Публикация событий за пределами модели предметной области | 368 |
| Издатель | 369 |
| Подписчики | 373 |
| Распространение новостей в удаленных ОГРАНИЧЕННЫХ КОНТЕКСТАХ | 375 |
| Согласованность инфраструктуры обмена сообщениями | 376 |
| Автономные службы и системы | 377 |
| Терпимость к задержкам | 379 |
| Хранилище событий | 380 |
| Архитектурные стили для пересылки сохраняемых событий | 385 |
| Публикация уведомлений в виде ресурсов RESTful | 386 |
| Публикация уведомлений с помощью промежуточного программного обеспечения для обмена сообщениями | 391 |
| Реализация | 393 |
| Публикация экземпляров класса NotificationLog | 394 |
| Публикация уведомлений на основе сообщений | 398 |
| Резюме | 407 |
| Глава 9. Модули | 409 |
| Разработка МОДУЛЕЙ | 409 |
| Основные правила именования модулей | 413 |
| Соглашения о выборе имен для МОДУЛЕЙ | 414 |
| Модули контекста управления гибким проектированием | 416 |

| | |
|--|------------|
| МОДУЛИ на других уровнях | 420 |
| МОДУЛИ перед ОГРАНИЧЕННЫМ КОНТЕКСТОМ | 421 |
| Резюме | 422 |
| Глава 10. Агрегаты | 423 |
| Использование АГРЕГАТОВ в СМЫСЛОВОМ ЯДРЕ системы Scrum | 424 |
| Первая попытка: крупнокластерный агрегат | 425 |
| Вторая попытка: множество агрегатов | 427 |
| Правило: моделируйте истинные инварианты в границах согласованности | 430 |
| Правило: проектируйте небольшие агрегаты | 432 |
| Не доверяйте каждому сценарию использования | 436 |
| Правило: ссылайтесь на другие АГРЕГАТЫ по идентификаторам | 437 |
| Ссылки на АГРЕГАТЫ по их идентификаторам | 439 |
| Навигация по модели | 440 |
| Масштабируемость и распределение | 441 |
| Правило: используйте принцип итоговой согласованности за пределами границы | 442 |
| Спрашивайте, чья эта обязанность | 445 |
| Причины нарушения правил | 446 |
| Причина 1: удобство пользовательского интерфейса | 446 |
| Причина 2: отсутствие технических механизмов | 447 |
| Причина 3: глобальные транзакции | 448 |
| Причина 4: производительность запросов | 448 |
| Выполнение правил | 449 |
| Понимание через открытие | 449 |
| Пересмотр проекта | 449 |
| Оценка стоимости агрегата | 451 |
| Типичные сценарии использования | 453 |
| Затраты памяти | 454 |
| Исследование альтернативного проекта | 456 |
| Обеспечение итоговой согласованности | 457 |
| Должен ли член команды делать эту работу | 458 |
| Время принимать решения | 460 |
| Реализация | 461 |
| Создайте корневую сущность с уникальным идентификатором | 461 |
| Полезные части ОБЪЕКТОВ-ЗНАЧЕНИЙ | 462 |
| Использование закона Деметры и принципов совмещения данных и поведения | 463 |
| Оптимистический параллелизм | 466 |
| Как избежать внедрения зависимости | 468 |
| Резюме | 469 |

| | |
|--|-----|
| Глава 11. Фабрики | 471 |
| ФАБРИКИ в модели предметной области | 471 |
| ФАБРИЧНЫЙ МЕТОД в КОРНЕ АГРЕГАТА | 473 |
| Создание экземпляров класса <code>CalendarEntry</code> | 474 |
| Создание экземпляров класса <code>Discussion</code> | 478 |
| ФАБРИКА СЛУЖБ | 479 |
| Резюме | 482 |
| Глава 12. Хранилища | 483 |
| Хранилища, ориентированные на имитацию коллекции | 485 |
| Реализация с помощью системы <code>Hibernate</code> | 490 |
| Реализация с помощью системы <code>TopLink</code> | 499 |
| ХРАНИЛИЩА, ориентированные на механизм постоянного хранения | 501 |
| Реализация с помощью системы <code>Coherence</code> | 503 |
| Реализация с помощью системы <code>MongoDB</code> | 509 |
| Дополнительные поведенческие функции | 514 |
| Управление транзакциями | 517 |
| Предупреждение | 521 |
| Иерархии типов | 522 |
| Хранилище и объект доступа к данным | 525 |
| Тестирование хранилищ | 526 |
| Тестирование реализаций в оперативной памяти | 530 |
| Резюме | 533 |
| Глава 13. Интеграция ограниченных контекстов | 535 |
| Основы интеграции | 536 |
| Распределенные системы совершенно другие | 537 |
| Обмен информацией через границы систем | 538 |
| Интеграция с помощью ресурсов <code>RESTful</code> | 545 |
| Реализация ресурса <code>RESTful</code> | 547 |
| Реализация клиента в архитектуре <code>REST</code> с помощью ПРЕДОХРАНИТЕЛЬНОГО УРОВНЯ | 550 |
| Интеграция с помощью сообщений | 557 |
| Информирование о владельцах продукта и членах команды | 557 |
| Можете ли вы принять на себя ответственность | 564 |
| Длительные процессы, или Как избежать ответственности | 569 |
| Конечные автоматы и механизмы для отслеживания простоев | 581 |

| | |
|--|-----|
| Проектирование более сложных процессов | 592 |
| Если система не допускает обмен сообщениями | 595 |
| Резюме | 596 |
| Глава 14. Приложение | 599 |
| Пользовательский интерфейс | 602 |
| Визуализация объектов предметной области | 603 |
| Визуализация объектов передачи данных с помощью экземпляров АГРЕГАТА | 604 |
| Использование посредника для публикации внутреннего состояния агрегата | 605 |
| Визуализация экземпляров агрегата с помощью объекта полезных данных предметной области | 606 |
| Представления состояний экземпляров агрегата | 607 |
| Оптимальные запросы сценария использования | 608 |
| Работа с многочисленными и разнородными клиентами | 608 |
| Адаптеры визуализации и реакция на пользовательские операции редактирования | 609 |
| Прикладные службы | 613 |
| Пример прикладной службы | 613 |
| Не связанный вывод службы | 620 |
| Компоновка многочисленных ОГРАНИЧЕННЫХ КОНТЕКСТОВ | 623 |
| Инфраструктура | 625 |
| Контейнеры стандартных компонентов | 626 |
| Резюме | 630 |
| Приложение. Агрегаты и источники событий | 631 |
| Внутри прикладной службы | 633 |
| Обработчики команд | 642 |
| Синтаксис лямбда-выражений | 646 |
| Управление параллельной работой | 647 |
| Структурная свобода шаблона A+ES | 650 |
| Производительность | 651 |
| Реализация хранилища событий | 654 |
| Реляционный механизм постоянного хранения | 658 |
| Хранение объектов BLOB | 660 |
| Специализированные агрегаты | 662 |
| Проекция модели чтения | 663 |
| Комбинация с АГРЕГАТОМ | 666 |

| | |
|---|------------|
| Расширение событий | 666 |
| Вспомогательные средства и шаблоны | 669 |
| Механизмы сериализации событий | 669 |
| Неизменяемость события | 670 |
| Объекты-значения | 670 |
| Генерация контрактов | 673 |
| Модульное тестирование и спецификации | 675 |
| ИСТОЧНИКИ СОБЫТИЙ в функциональных языках | 676 |
| Библиография | 678 |
| Предметный указатель | 683 |