

Посвящается Тине и Майе

Содержание

Об авторе	11
О техническом рецензенте	12
Благодарности	13
Введение	14
От издательства	15
Глава 1. Предметно-ориентированное проектирование	16
Концепции предметно-ориентированного проектирования	17
Предметная область/бизнес-домен.....	17
Поддомены/ограниченные контексты	19
Модель предметной области.....	22
Агрегаты/объекты-сущности/объекты-значения	23
Правила предметной области.....	24
Команды/запросы.....	25
События	26
Саги.....	26
Резюме	27
Глава 2. Проект Cargo Tracker	29
Основная предметная область (домен).....	29
Проект Cargo Tracker: поддомены/ограниченные контексты	30
Проект Cargo Tracker: модель предметной области (домена)	33
Агрегаты	34
Идентификаторы агрегатов.....	34
Сущности.....	35
Объекты-значения	36
Проект Cargo Tracker: операции модели предметной области (домена)	39
Саги.....	40
Сервисы модели предметной области	41
Проектирование сервисов модели предметной области (домена).....	43
Проект Cargo Tracker: реализации с использованием предметно-ориентированного проектирования	45
Резюме	46
Глава 3. Проект Cargo Tracker: Jakarta EE	47
Платформа Java EE.....	48
Смена торговой марки на Jakarta EE и дальнейшее развитие.....	48

Спецификации платформы Jakarta EE	49
Технологии веб-приложений.....	50
Сервлет Java	50
JavaServer Faces	51
JavaServer Pages.....	51
Expression Language.....	51
JSP Standard Tag Library (JSTL)	52
Java API для WebSocket	52
Java API для связывания с форматом JSON	52
Java API для обработки формата JSON	52
Технологии корпоративных приложений	52
Enterprise Java Beans (3.2).....	52
Contexts and Dependency Injection для Java (2.0)	53
Валидация компонентов Bean.....	53
Java Persistence API (JPA)	53
Java Transaction API (JTA)	54
Общие аннотации (Common Annotations)	54
Перехватчики (Interceptors).....	54
Веб-сервисы в Jakarta EE	54
Java API for RESTful Web Services (JAX-RS)	54
Технологии обеспечения безопасности	54
Java EE Security API (1.0)	55
Итоговый обзор спецификаций Jakarta EE	55
Cargo Tracker как модульное монолитное приложение.....	55
Ограниченные контексты с использованием платформы Jakarta EE	56
Пакет interfaces	58
Пакет application.....	59
Пакет domain.....	60
Пакет infrastructure.....	61
Совместно используемые ядра.....	61
Реализация модели предметной области (домена) с использованием Jakarta EE	62
Агрегаты.....	62
Сущности.....	70
Объекты-значения	72
Правила предметной области (домена).....	76
Команды.....	77
Запросы.....	78
Реализация сервисов предметной области с использованием Jakarta EE	79
Входящие сервисы.....	79
RESTful API.....	79
Собственные веб-API.....	80
Сервисы приложения	81
Сервисы приложения: события	83
Исходящие сервисы.....	86
Общая схема реализации.....	87
Резюме.....	88

Глава 4. Проект Cargo Tracker: Eclipse MicroProfile	89
Платформа Eclipse MicroProfile	89
Платформа Eclipse MicroProfile: функциональные возможности	91
Платформа MicroProfile: основные спецификации	93
Конфигурация Eclipse MicroProfile	93
Проверка работоспособности Eclipse MicroProfile	94
Аутентификация Eclipse MicroProfile JWT Authentication	94
Метрики Eclipse MicroProfile	94
Eclipse MicroProfile OpenAPI	94
Eclipse MicroProfile OpenTracing	94
Eclipse MicroProfile Type Safe Rest Client	95
Eclipse MicroProfile: спецификации поддержки	95
Context and Dependency Injection (CDI) for Java (2.0)	95
Общие аннотации	96
Java API for RESTful Web Services (JAX-RS)	96
Java API for JSON Binding	96
Java API for JSON Processing	96
Итоговый обзор спецификаций Eclipse MicroProfile	97
Реализация Cargo Tracker: Eclipse MicroProfile	97
Выбор реализации: проект Helidon MP	98
Реализация Cargo Tracker: ограниченные контексты	99
Ограниченные контексты: создание пакетов	101
Ограниченные контексты: структура пакета	103
Интерфейсы	104
Приложение	105
Предметная область (домен)	106
Инфраструктура	106
Реализация приложения Cargo Tracker	108
Модель предметной области (домена): реализация	110
Модель основного домена: реализация	111
Агрегаты, сущности и объекты-значения	111
Операции модели предметной области (домена)	122
Команды	122
Запросы	125
События	125
Сервисы модели предметной области (домена)	128
Входящие сервисы	129
Сервисы приложения	136
Исходящие сервисы	144
Итоговый обзор реализации	159
Резюме	160
Глава 5. Проект Cargo Tracker: платформа Spring	161
Платформа Spring	162
Spring Boot: функциональные возможности	164
Spring Cloud	165

Итог краткого обзора рабочей среды Spring.....	166
Ограниченные контексты и платформа Spring Boot.....	166
Ограниченные контексты: формирование пакетов	168
Ограниченные контексты: структура пакета	169
Пакет interfaces	171
Пакет application.....	171
Пакет domain.....	172
Пакет infrastructure.....	173
Реализация приложения Cargo Tracker	176
Модель предметной области (домена): реализация	177
Модель основного домена: реализация	178
Агрегаты, сущности и объекты-значения	178
Операции модели предметной области (домена).....	188
Команды	188
Запросы	191
События.....	192
Регистрация событий	194
Сервисы модели предметной области (домена)	197
Входящие сервисы.....	198
REST API	198
Сервисы приложения	206
Сервисы приложения: делегирование команд и запросов	207
Исходящие сервисы.....	211
Исходящие сервисы: классы репозитория.....	212
Исходящие сервисы: REST API	213
Итоговый обзор реализации.....	225
Резюме.....	225
Глава 6. Проект Cargo Tracker: рабочая среда Axon.....	226
Шаблон Event Sourcing	227
Методика CQRS	230
Рабочая среда Axon.....	233
Компоненты рабочей среды Axon	233
Компоненты предметной области (домена) Axon Framework.....	234
Агрегаты.....	234
Команды и обработчики команд.....	235
События и обработчики событий.....	235
Обработчики запросов	235
Саги	235
Компоненты модели регулирования и координации Axon Framework	236
Шина команд	236
Шина запросов.....	237
Шина событий.....	238
Саги	239
Компоненты инфраструктуры Axon: Axon Server	240
Приложение Cargo Tracker и рабочая среда Axon.....	244
Ограниченные контексты в Axon	244

Ограниченные контексты: создание артефакта	247
Ограниченные контексты: структура пакета	248
Пакет interfaces	249
Пакет application.....	250
Пакет domain.....	251
Пакет infrastructure.....	251
Реализация модели предметной области с использованием Axon.....	254
Агрегаты.....	254
Состояние	257
Обработка команд	261
Публикация событий.....	265
Сопровождение состояния	267
Проекции агрегатов	278
Обработчики запросов	282
Саги	286
Подведение итогов реализации	291
Реализация сервисов модели предметной области (домена) с использованием Axon.....	292
Входящие сервисы.....	292
Сервисы приложения	296
Резюме.....	298
Предметный указатель	299

Об авторе

Виджей Наир (Vijay Nair) в настоящее время является руководителем подразделения проектирования и разработки платформ для банковских приложений типа «программное обеспечение как услуга» (SaaS) компании Oracle. Как энтузиаст предметно-ориентированного проектирования и распределенных систем, он обладает 18-летним практическим опытом в области проектирования архитектуры, создания и реализации особо ответственных приложений непрерывного действия для сферы финансовых услуг по всему миру. Виджей Наир доступен для общения на собственном веб-сайте www.practicalddd.com или через «Твиттер» @FusionVJ. Виджей Наир живет в Маунтин Вью (Калифорния) вместе с женой и дочерью.

О техническом рецензенте



Мануэль Хордан (Джордан) Элера (Manuel Jordan Elera) – разработчик и исследователь-самоучка (автодидакт), которому очень нравится изучать и осваивать новые технологии на собственном практическом опыте (экспериментах) и создавать новые технологические комбинации. Мануэль является обладателем наград Springy Award – Community Champion и Spring Champion 2013. Несмотря на дефицит свободного времени, он читает Библию и играет на гитаре, сочиняет музыку. Мануэль известен под псевдонимом `dr_pompeii`. Он является тех-

ническим рецензентом многочисленных книг издательства Apress, в том числе Pro Spring, Fourth Edition (2014); Practical Spring LDAP (2013); Pro JPA 2, Second Edition (2013) и Pro Spring Security (2013). Можно прочесть его 13 подробных руководств по многим технологиям Spring, а также связаться с ним лично через его блог на сайте www.manueljordanelera.blogspot.com и наблюдать за его деятельностью в аккаунте «Твиттера» `@dr_pompeii`.

Благодарности

В первую очередь моя самая глубокая сердечная благодарность за то, что создание этой книги стало возможным, адресована гурзу Jakarta EE (Enterprise Edition) Реза Рахману (Reza Rahman). Работая в компании Oracle, он создал инициативное направление Cargo Tracker как эскизный проект для Java EE Patterns на основе предметно-ориентированного проектирования (DDD). Я буду вечно благодарен ему за предоставленную мне возможность участвовать в этом проекте.

Благодарю приверженца предметно-ориентированного проектирования и энтузиаста использования программной среды Axon Framework Свапнилу Сурве (Swapnil Surve), инженера-архитектора ПО из Феникса (Phoenix) за техническое рецензирование содержимого и предложения по главам 5 и 6. Отдельная благодарность Элларду Бьюзи (Allard Buijze) (создателю программной среды Axon Framework) за техническое рецензирование главы 6.

Спасибо всем сотрудникам компании Oracle, которые помогли написать эту книгу, моим руководителям (Викраму, Тиксу, Чету) и моей команде подчиненных, которые ежедневно обучают меня (Сурабу, Шрипаду, Хари, Павану, Дашарату и Махендру).

Спасибо городу Маунтин Вью (Калифорния) за предоставление отдельной комнаты в потрясающей библиотеке, где я провел многие часы за написанием этой книги.

Если в семье появляется новорожденный, то человек в здравом уме вряд ли займется написанием книги. Поддержка моей семьи была колоссальной на протяжении всего периода работы над книгой, и я безмерно благодарен за это.

Спасибо моим братьям Гаутаму, Рохиту, Сумиту и Сачину, моей сестре Вините и ее мужу Мадху, моим детям Варуну и Арье, наконец моим родителям и родителям моей жены, за то, что они не пожалели своего времени и были с нами, помогали нам.

Последняя, но самая главная благодарность моей жене Тине. Она разрывалась между множеством дел, заботилась о новорожденном и обеспечивала мне возможность работать над книгой, прилагая нечеловеческие усилия и в прямом смысле жертвуя собой. Эта книга твоя в той же мере, что и моя. Спасибо тебе.

Введение

Предметно-ориентированное проектирование (Domain Driven Design) никогда не было настолько востребованным, как в современном мире разработки программного обеспечения. Концепции и шаблоны предметно-ориентированного проектирования помогают создавать правильно спроектированные приложения масштаба предприятия вне зависимости от того, являются ли эти приложения привычными монолитным программами или более современными приложениями на основе микросервисов.

Цель этой книги – снять покров мистики с концепций предметно-ориентированного проектирования, представив практические методы их реализации для более привычных устоявшихся монолитных приложений, а также для современных приложений, использующих микросервисы. С помощью реально существующего проекта приложения Cargo Tracker в книге подробно рассматриваются практические реализации разнообразных шаблонов предметно-ориентированного проектирования для обоих стилей приложений с применением различных инструментальных средств и рабочих программных сред из Enterprise Java Space (Jakarta EE, Eclipse MicroProfile, Spring Boot и Axon Framework). При таком подходе читатель получает полностью завершённую картину и возможность практического использования любого из этих инструментальных программных средств и сред в собственном процессе предметно-ориентированного проектирования.

Надеюсь, что чтение этой книги принесет вам пользу.

От издательства

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге, – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв прямо на нашем сайте www.dmkpress.com, зайдя на страницу книги, и оставить комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com, при этом напишите название книги в теме письма.

Если есть тема, в которой вы квалифицированы, и вы заинтересованы в написании новой книги, заполните форму на нашем сайте http://dmkpress.com/authors/publish_book/ или напишите в издательство: dmkpress@gmail.com.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы удостовериться в качестве наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг – возможно, ошибку в тексте или в коде, – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от расстройств и поможете нам улучшить последующие версии данной книги.

Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу dmkpress@gmail.com, и мы исправим это в следующих тиражах.

Скачивание исходного кода

Скачать файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте www.dmkpress.com на странице с описанием соответствующей книги.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Packt Publishing очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконно выполненной копией любой нашей книги, пожалуйста, сообщите нам адрес копии или веб-сайта, чтобы мы могли применить санкции.

Пожалуйста, свяжитесь с нами по адресу dmkpress@gmail.com со ссылкой на подозрительные материалы.

Мы высоко ценим любую помощь по защите наших авторов, помогающую нам предоставлять вам качественные материалы.

Глава 1

Предметно-ориентированное проектирование

Предметно-ориентированное проектирование (Domain Driven Design) предлагает основательную, систематическую и полнофункциональную методику проектирования и разработки программного обеспечения. Эта методика предоставляет набор инструментальных средств и конкретных методов, которые помогают уменьшить сложность бизнес-процесса при сохранении основной (центральной) бизнес-модели как базовой сущности методики.

Предметно-ориентированное проектирование являлось предпочтительной методикой для обычных (в подавляющем большинстве монолитных) проектов в течение длительного времени, но с появлением архитектуры микросервисов концепции предметно-ориентированного проектирования стали в постоянно возрастающей степени применяться даже к этой новой парадигме архитектуры.

Эта книга разделена на две основные части.

Моделирование концепций предметно-ориентированного проектирования

Реализация предметно-ориентированного проектирования начинается с моделирования процесса для идентификации артефактов (поддоменов, ограниченных контекстов, моделей доменов, правил доменов), которые отображаются в концепции предметно-ориентированного проектирования. В нескольких первых главах книги приводится обобщенный обзор концепций предметно-ориентированного проектирования, затем описывается полный процесс моделирования для идентификации и документирования связанных с ним артефактов с подробным рассмотрением вариантов использования на примере конкретного реально существующего приложения.

Реализация концепций предметно-ориентированного проектирования

Далее в книге подробнейшим образом рассматриваются реализации этих концепций. Используя Enterprise Java как основную платформу, мы исследуем три различные реализации:

- на основе монолитной архитектуры с использованием платформы Java EE/Jakarta EE;
- на основе архитектуры микросервисов на платформе MicroProfile;
- на основе архитектуры микросервисов на платформе Spring.

Эти реализации охватывают три основные платформы, которые наиболее широко распространены в среде Enterprise Java и предоставляют полную дета-

лизацию практической реализации шаблонов предметно-ориентированного проектирования.

КОНЦЕПЦИИ ПРЕДМЕТНО-ОРИЕНТИРОВАННОГО ПРОЕКТИРОВАНИЯ

После ознакомления с основной целью этой книги можно перейти к рассмотрению основных концепций предметно-ориентированного проектирования.

Предметная область/бизнес-домен

Самой первой основной концепцией, с которой необходимо познакомиться, является идентификация предметной области (problem space), или бизнес-домена (business domain). Предметная область/бизнес-домен – это отправной пункт для всего процесса предметно-ориентированного проектирования, где определяется главная бизнес-задача, которую вы намерены решить с помощью предметно-ориентированного проектирования.

Рассмотрим эту концепцию более подробно, используя несколько практических примеров.

Сначала рассматривается пример из области автофинансирования, показанный на рис. 1.1. Деловая сфера автофинансирования предполагает управление автоматическими кредитами и арендами, т. е. вы, как провайдер услуг автофинансирования, должны предоставлять кредиты/аренды клиентам, обслуживать их, наконец при возникновении каких-либо проблем устранять их или возвращать предоставленные активы. В этом случае предметную область можно определить как управление автокредитами/автоарендами. Эту предметную область также можно обозначить термином «основной бизнес-домен» (core business domain) и «бизнес-задача» (business problem), которую вы намерены решить с использованием предметно-ориентированного проектирования.

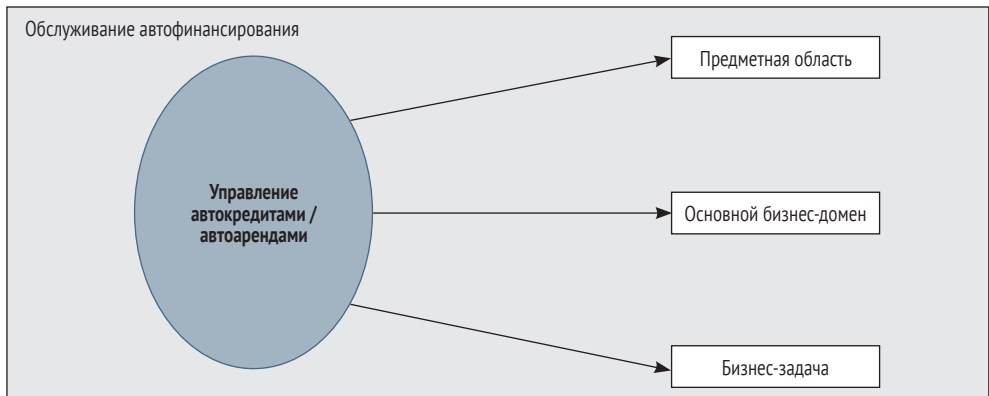


Рис. 1.1 ❖ Предметная область сферы обслуживания автофинансирования

Второй пример взят из банковской сферы. В отличие от первого примера здесь имеется не одна, а несколько предметных областей, в которых необходимо обеспечить решение с использованием предметно-ориентированного проектирования (см. рис. 1.2).

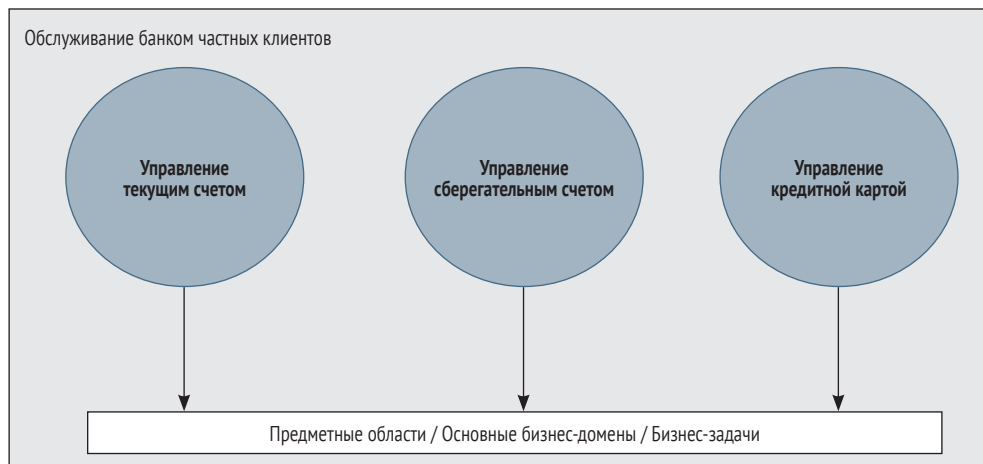


Рис. 1.2 ❖ Бизнес-домены (предметные области) в сфере обслуживания банком частных клиентов

От имени банка вы могли бы предложить обслуживание частных клиентов (рис. 1.2) в обобщенном случае или обслуживание корпоративных клиентов (рис. 1.3), если клиентом является предприятие или организация. В каждом рассматриваемом здесь варианте существует несколько предметных областей или основных бизнес-доменов.

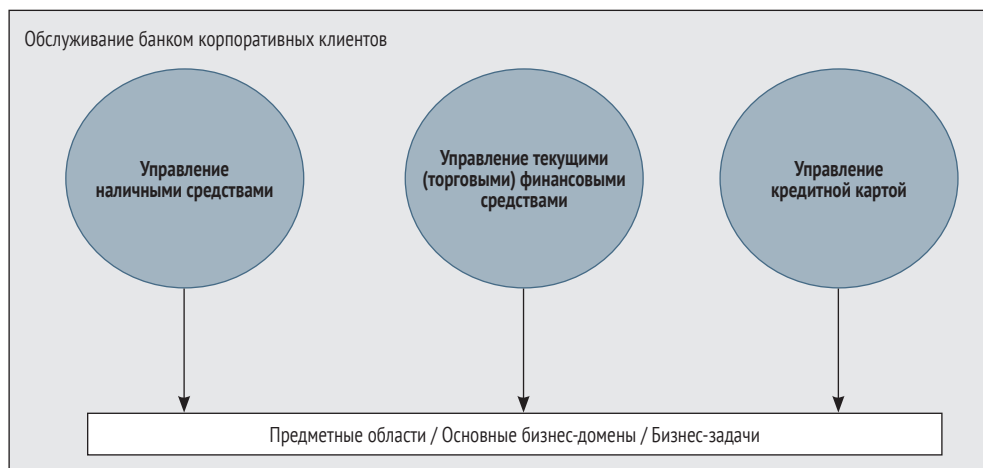


Рис. 1.3 ❖ Основные бизнес-домены в сфере обслуживания банком корпоративных клиентов

Предметные области/бизнес-домены всегда неизменно преобразовываются в основные проектные бизнес-утверждения, которые вы предлагаете как деловое предприятие.

Поддомены/ограниченные контексты

После определения основного бизнес-домена следующим шагом является разделение этого домена (предметной области) на поддомены (sub-domains). Определение поддоменов по существу означает разделение различных бизнес-возможностей вашего основного бизнес-домена на связанные друг с другом элементы функциональности бизнеса.

Снова обратимся к примеру из предметной области автофинансирования, которую можно разделить на три поддомена, как показано на рис. 1.4:

- поддомен «Первоначальные выплаты» – обеспечивает возможность выдачи новых автокредитов/предоставления автоматических аренд клиентам;
- поддомен «Обслуживание» – обеспечивает возможность обслуживания (например, ежемесячное выписывание счетов/сбор платежей) автокредитов/автоаренд;
- поддомен «Денежные сборы (инкассо)» – обеспечивает управление автокредитами/автоарендами, если возникают какие-либо проблемы (например, клиент не выполняет обязательства по платежам).

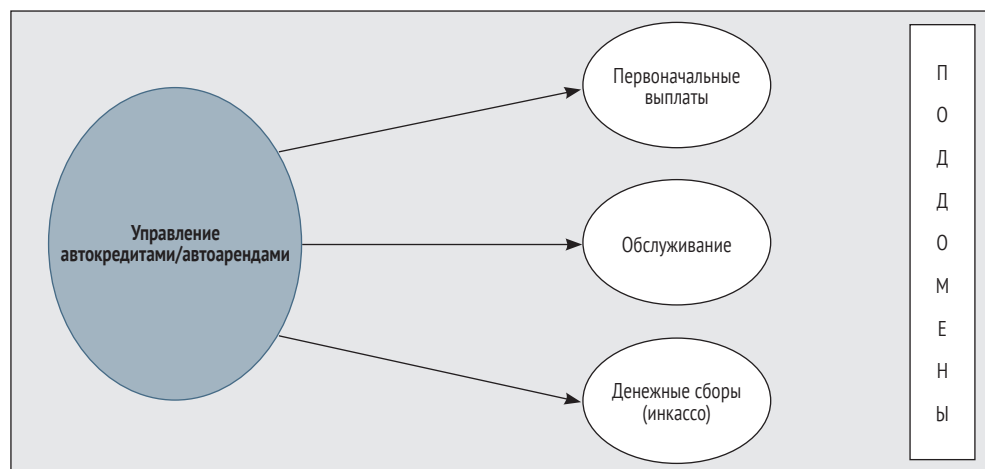


Рис. 1.4 ❖ Поддомены в предметной области «Управление автокредитами/автоарендами»

Очевидно, что поддомены определяются в терминах бизнес-возможностей (функций) вашей основной сферы деятельности, используемых повседневно.

На рис. 1.5 показан еще один пример определения поддоменов для одной из предметных областей сферы обслуживания банком частных клиентов – для бизнес-домена «Управление кредитной картой».

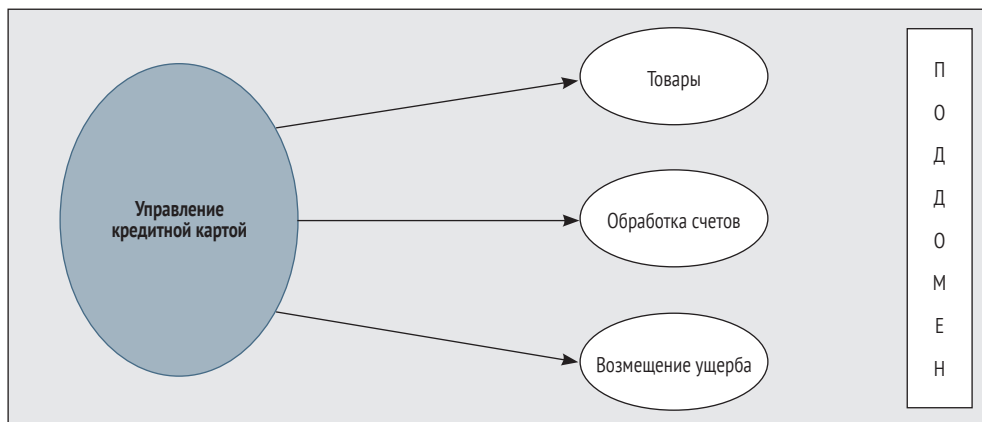


Рис. 1.5 ❖ Поддомены в предметной области «Управление кредитной картой»

Перечислим эти поддомены:

- «Товары» – определяет возможности по управлению всеми типами товаров, оплачиваемых кредитной картой;
- «Обработка счетов» – определяет возможности обработки счетов, оплачиваемых кредитной картой клиента;
- «Возмещение ущерба» – определяет возможности по управлению любыми типами исков, претензий и т. п. при возмещении ущерба по кредитной карте клиента.

И в этом случае действительные бизнес-возможности помогают точно определить поддомены.

Но что такое ограниченные контексты (bounded contexts)?

Напомним, что мы начали изучение процесса предметно-ориентированного проектирования с определения предметных областей (бизнес-доменов). Далее мы продолжили работу с бизнес-доменами, разделяя их на различные возможности (функции), чтобы определить соответствующие поддомены, отображающие эти функциональные возможности в конкретной деловой сфере.

Теперь необходимо приступить к созданию решений для соответствующих доменов/поддоменов, определенных ранее, т. е. необходимо переместиться из предметной области в область решений (solution space), где ограниченные контексты играют центральную роль.

Проще говоря, ограниченные контексты (bounded contexts) – это проектные решения для ранее определенных бизнес-доменов/поддоменов.

Процедура определения (идентификации) ограниченных контекстов руководствуется главным образом связностью, которая необходима внутри бизнес-домена и между установленными поддоменами.

Вернемся к первому примеру предметной области автофинансирования. Можно было бы выбрать единое решение для всей предметной области в целом, т. е. единый ограниченный контекст для всех поддоменов. Но можно определить ограниченные контексты, связанные с одним или с несколькими поддоменами.

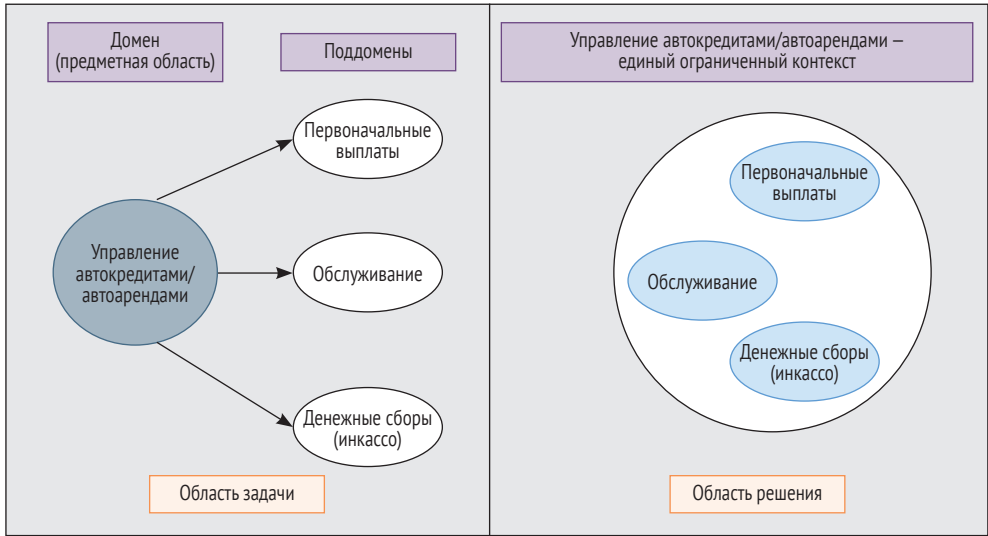


Рис. 1.6 ❖ Предметная область автофинансирования с решением в виде единого ограниченного контекста

Решением на рис. 1.6 для предметной области автофинансирования (автокредитов и аренд) является единый ограниченный контекст для всех поддоменов.

Другой подход – решения для различных поддоменов в предметной области автофинансирования как отдельные ограниченные контексты, как показано на рис. 1.7.

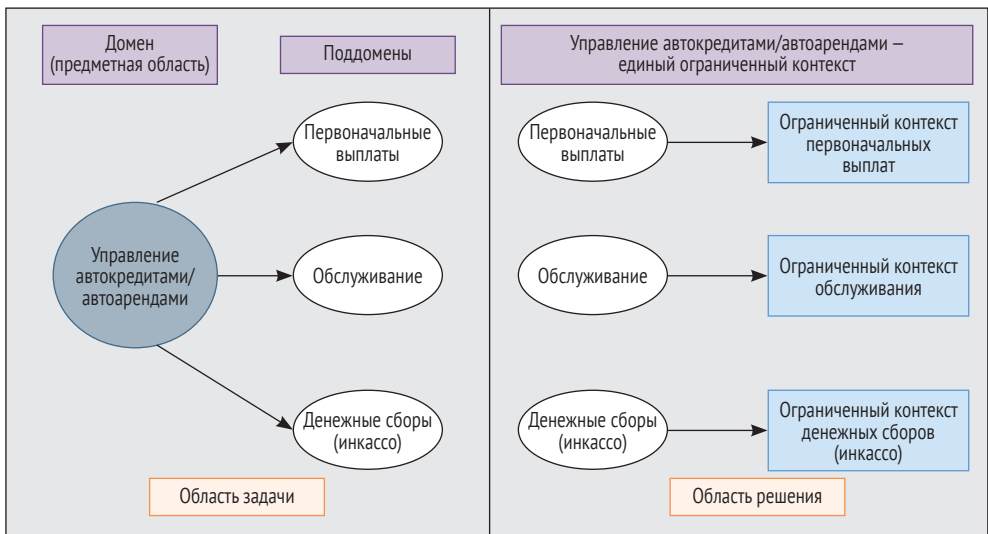


Рис. 1.7 ❖ Предметная область автофинансирования с решениями в виде отдельных ограниченных контекстов

При выборе конкретного подхода к развертыванию решения не существует никаких ограничений, поскольку ограниченный контекст интерпретируется как единый связный элемент. Можно использовать монолитное развертывание при подходе с несколькими ограниченными контекстами (единый веб-архив Web Archive [WAR] с несколькими JAR-файлами для каждого отдельного ограниченного контекста). Можно выбрать модель развертывания с использованием микросервисов, когда каждый ограниченный контекст представлен как отдельный контейнер. Также можно выбрать модель без сервера (serverless), в которой каждый ограниченный контекст развертывается как функция.

В последующих главах мы будем рассматривать каждый возможный тип модели развертывания как часть конкретных реализаций.

МОДЕЛЬ ПРЕДМЕТНОЙ ОБЛАСТИ

Теперь перейдем к самой важной и ответственной части процесса создания решения для предметной области – определению модели домена для ограниченного контекста. Если говорить кратко, то модель домена (domain model) – это реализация основной бизнес-логики внутри конкретно определенного ограниченного контекста.

На языке бизнеса это означает определение следующих элементов:

- бизнес-сущностей (бизнес-объектов);
- бизнес-правил;
- бизнес-поток;
- бизнес-операций;
- бизнес-событий.

На техническом языке, принятом в мире предметно-ориентированного проектирования, элементы, подлежащие определению (идентификации), обозначаются следующим образом:

- агрегаты/сущности/объекты-значения;
- правила предметной области (домена);
- саги (sagas);
- команды/запросы;
- события.

Соответствие между приведенными выше двумя списками показано на рис. 1.8. В соответствии с показанной здесь схемой конструкции бизнес-языка отображаются в соответствующие конструкции технического языка предметно-ориентированного проектирования.

Поскольку мы будем подробно рассматривать все эти разнообразные концепции в последующих главах, необходимо кратко описать их здесь. Если эти концепции и элементы сейчас будут не вполне понятны, не беспокойтесь. В последующих главах вы получите полное представление об этих концепциях и элементах.

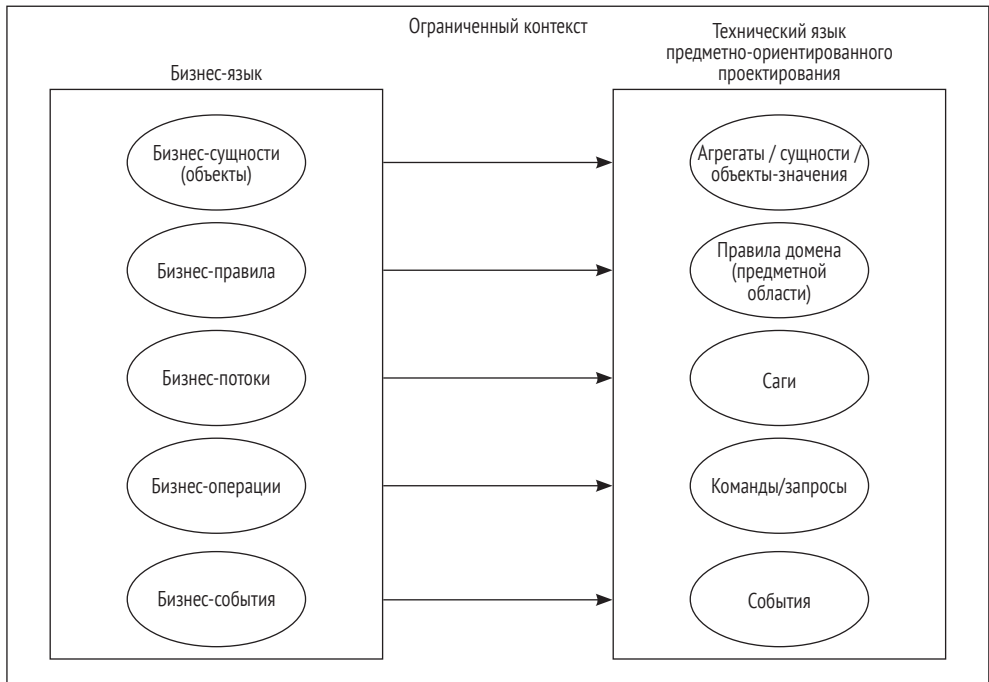


Рис. 1.8 ❖ Модель домена (предметной области) ограниченного контекста в терминах бизнес-логики и соответствующие им термины технического языка предметно-ориентированного проектирования

Агрегаты/объекты-сущности/объекты-значения

Агрегат (aggregate) (также называемый корневым агрегатом – root aggregate) – это центральный бизнес-объект в ограниченном контексте, который определяет область логической связности (целостности) внутри этого ограниченного контекста. Каждый аспект конкретного ограниченного контекста начинается и заканчивается внутри соответствующего корневого агрегата.

Агрегат = Самый главный идентификатор конкретного ограниченного контекста

Объекты-сущности (entity objects), или просто сущности (entity), обладают собственной идентичностью, но не могут существовать вне корневого агрегата, т. е. сущности создаются при создании корневого агрегата и уничтожаются при уничтожении корневого агрегата.

(Объекты)-сущности = Вторичные идентификаторы конкретного ограниченного контекста

Объекты-значения (value objects) не обладают собственной идентичностью и с легкостью заменяемы в любом экземпляре корневого агрегата или сущности.

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru