

Введение

Я начал работать с технологиями Java еще начиная с 1995 года, вскоре после выпуска *комплекта разработки Java* (Java Development Kit – JDK) версии 1.0. До этого я много лет программировал на языках C и C++. Я был на самом деле восхищен такими возможностями Java, как межплатформенная переносимость, простой синтаксис (более простой, чем у языка C++, например), ориентированность на объекты, безопасность, богатство API и т.д.

На протяжении 20-летней карьеры мне пришлось изучить многое. Поэтому мне всегда нравилась простота; когда я встречаю сложность, начинаю сомневаться, является ли решение правильным. Именно это как раз я и почувствовал в 2000 году, когда язык Java 2 Enterprise Edition (J2EE) начал становиться господствующим. Обратите внимание, с этого момента я буду называть версию J2EE как JEE, поскольку цифра “2” была недавно ликвидирована из имени компанией Sun Microsystems.

Мой интерес к Java снизился в результате того, что я заметил ненужную сложность в JEE, обусловленную слоями абстракции. Я начал полагать, что компания Sun Microsystems (разработчик Java) предназначила Java и JEE для создания наиболее сложных корпоративных приложений, несколько проигнорировав относительно менее сложные приложения малого и среднего размера. Кроме того, я с разочарованием наблюдал потерю у людей здравого смысла, поскольку не раз натыкался на проекты, в которых среда Enterprise JavaBeans (EJB) использовалось вовсе не для распределенной обработки, например для локальной регистрации. По этому поводу в 2000 году я написал короткую статью для журнала JavaWorld.com (<http://www.javaworld.com/javaworld/jw-10-2000/jw-1006-soapbox.html>) под названием “Do You Really Need Enterprise JavaBeans?” (Действительно ли вам нужна Enterprise JavaBeans?). (Примерно пять лет спустя мы заметили переработку спецификации EJB 3.0, она была сделана проще, чтобы облегчить разработку.) Это подводит нас к содержанию этой книги и причине, по которой я ее написал.

Я был недавно приглашен компанией Fortune 100 в качестве консультанта при создании корпоративного Web-приложения, работающего в кластеризуемой среде. Рассматривая альтернативу стандартной модели JEE/EJB, я исследовал сетевые ресурсы и беседовал с компетентными людьми. В результате я остановился на решении, которое подразумевало использование Web-среды Spring MVC, отказоустойчивой объектно-реляционной (Object-Relational – OR) среды Hibernate, среды IDE Eclipse, среды проверки JUnit, утилиты Ant, нескольких библиотек дескрипторов и других продуктов. (Более подробная информация об этих продуктах приведена в книге далее, наряду с моими объяснениями по поводу выбора этих технологий.)

Я люблю работать с системами Spring и Hibernate, главным образом, потому, что они позволяют работать со *старыми простыми объектами Java* (Plain-Old Java Object – POJO) и избегать некоторых из проблем работы с EJB. Кроме того, работая с IDE Eclipse, я получил хороший опыт. Мне бы хотелось еще поговорить о том, как хорош этот продукт, именно поэтому я посвящаю данной теме целую главу. По моему мнению, такие продукты, как упомянутые выше, вдыхают новую жизнь в язык Java, когда он подвергается опасности потери популярности в связи с появлением таких продуктов, как .NET от Microsoft, LAMP (Linux, Apache, MySQL и PHP или Python/PERL), а также Ruby on Rails.

Хотя в этой книге и описаны технологии Spring, Hibernate и Eclipse, основная моя задача заключается в том, чтобы снабдить вас полным решением с технической точки зрения и с точки зрения процесса. С технической точки зрения я предоставляю сквозное решение (с использованием ряда инструментальных средств) для полной реализации типового Web-приложения с фоновым управлением транзакциями и подходящего для кластеризуемой среды. С точки зрения процесса, я недавно перешел с использования процесса Rational Unified Process (RUP) на процесс, состоящий из критериев моделей Agile Model Driven Development (AMDD; agilemodeling.com) и Extreme Programming (XP; extremeprogramming.org). В этой книге вы найдете описание концепций, а также такие артефакты, как статьи пользователей, планы выпуска, карточки CRC и многое другое. Идея заключается в том, чтобы предоставить вам исчерпывающее решение для быстрой разработки и развертывания приложений Java.

Еще одно, дополнительное, замечание по поводу моей квалификации. Я обладаю почти 20-летним опытом разработки с использованием таких фундаментальных технологий, как C/C++, Java, корпоративные реляционные базы данных, серверы приложений, Unix, Microsoft Windows и т.д. Однако в моей работе был приблизительно пятилетний перерыв, с 1996 по 1998 год, поскольку я основал собственную компанию и программировал в это время мало. Позже я продал компанию, чтобы вернуться к разработке. Даже при том, что в этой компании я был руководителем высшего ранга и имел несколько подчиненных, работавших на меня, я находил возможность разговаривать буквально с сотнями разработчиков и обсуждать с ними технологические новшества. Кроме этой компании, я также основал сетевое сообщество для разработчиков Java, которое разрослось до более чем 100 000 участников (я очень горжусь этим). Я надеюсь, что мой опыт этих предприятий добавит уникальную привлекательность этой книге. Надеюсь, что вы найдете эту книгу полезной и вам понравится читать ее!

Для кого написана эта книга

Эта книга подразумевает, что вы имеете некоторые практические навыки работы с Java и реляционными базами данных (включая SQL), а также опыт работы с командной строкой.

Эта книга предназначена также для следующих категорий читателей.

- Разработчики и архитекторы программного обеспечения. Они могут почерпнуть из этой книги знания о процессе разработки программного обеспечения высокого уровня, проектировании приложений, а также о глубокой и полной проверке кода Java и связанных с ними файлов на предмет полной функциональности на примере типичного корпоративного Web-приложения.

- Технические руководители и менеджеры. Руководители, обладающие опытом программирования, предпочтительно в Java или подобном языке, могут получить углубленное представление о создании приложений с использованием ряда технологий Java. Эти знания окажутся весьма полезны на этапе планирования проекта или при решении персоналом технических проблем (когда сроки поджимают; возможно, только для моральной поддержки). В качестве альтернативы технические менеджеры могут углубиться в определенную главу (например, главу 5, “Применение Hibernate для постоянных объектов”), чтобы понять, как та или иная технология работает и вписывается в общую картину.

Кроме того, из этой книги вы можете получить некоторое представление об альтернативах JEE, которые можно использовать для создания надежных приложений корпоративного класса. Если вы не знакомы с технологиями Agile Modeling и Extreme Programming или подбираете более легкий процесс разработки программного обеспечения, то этой книги могло бы оказаться достаточно, чтобы получить полное представление о разработке программных приложений.

Задачи этой книги

При написании этой книги решались следующие задачи.

- Гибкая разработка. Приоритетная задача этой книги заключается в том, чтобы продемонстрировать вам быстрые способы разработки приложений Java. Это достигается за счет комбинации нескольких аспектов: продуманного и минимизированного процесса разработки программного обеспечения, простого проекта (это снижает необходимость использования схем проектирования или слоев абстракции), применения вспомогательных технологий (типа Spring и Hibernate), работы с POJO вместо дистанционных объектов, а главное, манипуляций классическими технологиями реализации с открытым исходным кодом, когда это возможно. Короче говоря, идея заключается в том, чтобы сделать приложения Java по возможности более простыми и быстрыми в разработке.
- Полное решение. Вторая задача этой книги заключается в том, чтобы снабдить вас полным решением с технической точки зрения и с точки зрения процесса. По завершении изучения этой книги, вы должны быть способны создавать любые приложения, не только технически, но и с использованием процесса, описанного в этой книге. Кроме того, когда я не могу описать какую-либо технологию достаточно глубоко, предоставляю ссылки на ресурсы (Web-сайты) для дальнейшего ее исследования. Замечательной особенностью технологий, рассматриваемых в этой книге, является то, что вы можете получить полную систему, от пользовательского интерфейса до встроенной базы данных, наряду с возможностью планирования работ (благодаря среде Spring Framework), в одном единственном саморазвращающемся файле архива Web-приложения (.war)! Но вы можете в любой момент заменить упомянутые здесь технологии некоторыми другими, по своему выбору (например, использовать базу данных Oracle вместо HSQLDB). В результате, вы будете иметь полное решение, как с технической точки зрения, так и с точки зрения процесса!
- Использование решений исключительно за счет реализации с открытым исходным кодом не является задачей этой книги. Хотя я и старался использовать

в этой книге среды, инструменты и продукты только с открытым исходным кодом, проповедь реализации решений исключительно с открытым исходным кодом также не является задачей этой книги. Например, вы вполне можете отойти от требований производителя Java по переносимости и заменить один из описанных здесь продуктов каким-либо коммерческим. Однако реализация с открытым исходным кодом прошла очень долгий путь, и я просто поражен тем, насколько надежными оказались эти технологии и насколько хорошо они себя зарекомендовали. Например, такие технологии, как SDK Eclipse и Hibernate, на мой взгляд, значительно лучше, чем некоторые из их коммерческих дубликатов. Точно так же для корпоративного решения вы можете использовать все упомянутые в этой книге технологии и надеяться, что все остальные сработают так, как указано в их рекламе. Фактически я недавно реализовал корпоративное решение для большой компании, используя такие инструменты, как Spring Framework, Hibernate, Eclipse, JUnit, Ant и другие, упомянутые в этой книге! Но мы также использовали и коммерческие продукты, такие как WebLogic Server от BEA и сервер баз данных Oracle. Упомянутая выше компания (и несколько других, известных мне) основывает свои корпоративные решения на реализации технологий с открытым исходным кодом.

- Быстрое чтение. Эта книга преднамеренно имеет объем меньше, чем у типичной 600-страничной книги по Java. Это было сделано для того, чтобы позволить вам прочитать книгу быстро и сразу начать использовать описанные в ней решения в реальном мире. В свете сказанного выше я постарался сделать содержание этой книги как можно более рациональным и близким к сути дела. Одним из недостатков написания весьма “нетолстой” книги является то, что я вынужден был зачастую принимать жесткие решения о предоставляемом материале; однако я постарался все же включить побольше материала, связанного с процессом и технологией, в котором вы будете нуждаться при быстрой разработке приложений Java (как уже упоминалось в предыдущем абзаце, для предоставления полного решения).
- Простота. Каждый раз, когда это возможно, я предпочитаю изложить более простой подход выполнения задачи. Например, примеры рассматриваемых в этой книге приложений используют минимальные слои абстракции для достижения цели. Под слоями абстракции я подразумеваю чрезмерное использование схем проектирования, интерфейсов и дробления приложений. Каждый из этих элементов имеет огромный смысл, но их уместное использование — это хорошая практика, и я ей, как правило, следую, когда разрабатываю программное обеспечение. Кроме того, я полагаю, что простота должна также распространяться на проектирование, в котором я предпочитаю использовать UML, когда это необходимо, но в более простых случаях я предпочитаю использовать для диаграмм такие инструментальные средства, как OpenOffice.org, PowerPoint или Visio, а не что-нибудь столь сложное, как Rational Rose.
- Советы и приемы. Как вы могли уже заметить, при работе с инструментальными средствами и технологиями советы и приемы позволяют не только эффективнее использовать программный продукт, но и проще решать некоторые задачи. Для некоторых технологий, описываемых в этой книге, я привожу советы и описания приемов. Но и приложения сами содержат некоторые вспомога-

тельные средства, такие как контрольные списки и списки особенно полезных инструментов.

- Альтернативы. В этой книге (хоть и без подробностей) я попытаюсь предоставить альтернативы стандартным решениям. Я понимаю, что не существует одного решения для всех случаев. Например, в качестве IDE вы могли бы использовать NetBeans от Sun Microsystems или IntelliJ от JetBrains, а не Eclipse. Это вполне ожидаемо и полностью понятно. Организация этой книги учитывает такой случай, и вы вполне можете извлечь пользу из остальной части книги, заменив технологию, описанную здесь, на технологию по вашему выбору (например, JDO вместо Hibernate).

Что не описано

Эта книга подразумевает, что вы имеете практические знания в области Java и относительно хорошее понимание JEE. Здесь также в значительной степени подразумевается, что вы имеете приемлемое понимание процессов разработки программного обеспечения, реляционных баз данных, многоуровневых архитектур и т.д. С учетом этого, я сразу перехожу к вопросам, связанным с созданием нашего типового приложения. Кроме того, я считаю, что вам доступны соответствующие Web-сайты (и дополнительные средства) и не дублирую здесь эту информацию, рискуя выпустить устаревшую книгу.

С другой стороны, эта книга подразумевает, что вы не имеете никаких практических знаний об основных описанных здесь технологиях, таких как Spring Framework, Hibernate, Eclipse и т.д. С учетом сказанного выше, данная книга содержит изложение основ этих технологий и их совместной работы; эта книга делает также следующий шаг и обеспечивает вас кратким вводным курсом по некоторым из дополнительных возможностей, предоставляемых этими технологиями. Все, что не упомянуто здесь, выходит за рамки этой книги, поскольку существует множество других книг, посвященных большинству технологий, упомянутых в данной книге.

Что описано (технологии и процесс)

В этой книге основное внимание уделено разработке и в меньшей мере инфраструктуре. Другими словами, я уделю больше внимания технологиям разработки приложения, таким как Spring, Hibernate и Eclipse, а не программным продуктам, таким как серверы приложений (например, JBoss) или базы данных (например, MySQL). Кроме того, я предпочитаю сначала получить функциональные возможности реализуемого приложения, а затем осуществить его оптимизацию и рефакторинг. Все, что представлено в этой книге, опробовано в реальных приложениях, которые успешно работают (некоторые в кластеризуемой среде сервера приложений), поэтому мне бы не хотелось, чтобы у вас создалось впечатление, что мы игнорируем инфраструктуру в целом. Одна из задач этой книги заключается в краткости и конкретности, поэтому я решил практически полностью сосредоточиться на разработке хорошо масштабируемого приложения.

С учетом преимуществ переносимости на операционные системы (Operating System — OS), в теории, когда ваше приложение будет готово и развернуто, вы можете перевести его на более надежный сервер Web-приложения в комбинации с базой дан-

ных. Например, сначала вы могли бы использовать относительно маломощные продукты, описанные в этой книге (Tomcat от Apache и HSQLDB), а затем перейти на комбинацию сервера JBoss Application Server и базы данных MySQL или наращивать вычислительные возможности далее, до комбинации серверов BEA WebLogic Server и Oracle, например. В этом красота Java; т.е. не только переносимость на другую OS, но и возможность смены производителя.

Еще одно замечание об основных технологиях, описанных в этой книге, а именно Spring, Hibernate и Eclipse. Хотя именно эти технологии я использовал в данной книге, я также упомянул альтернативные и конкурирующие технологии, которые советую вам также рассмотреть. Например, если вы решили использовать JDO вместо Hibernate, вполне можете использовать содержимое всех остальных глав, за исключением, возможно, посвященных Hibernate.

Как организована эта книга

Эта книга организована так, чтобы каждая последующая глава основывалась на материале предыдущей. Кроме того, поскольку главы логически разделены, вы можете непосредственно перейти к определенной главе (например, главе 6, “Обзор среды Spring Framework”) и прочитать только ее содержимое. Кроме того, вы можете пропустить некоторую главу, если использование описанной в ней технологии вас не интересует (например, если, вместо Eclipse, вы используете NetBeans, главу 8, “Феномен Eclipse!”, можно пропустить). Глава 1, “Введение в разработку приложений на Java”, содержит краткий обзор и предварительное описание технологий и процессов, используемых в этой книге. Глава 2, “Простое приложение: сетевая система учета рабочего времени”, посвящена, в основном, определению бизнес-требований для нашего типового приложения; но она содержит также краткий обзор методологий AMDD и XP. В главе 3, “Архитектура и модель проекта на базе XP и AMDD”, рассматривается проект нашего типового приложения. В главе 4, “Установка среды: JDK, Ant и JUnit”, описана установка среды.

Теперь мы быстро переходим в мир кода Java и рассматриваем в главе 5, “Применение Hibernate для постоянных объектов”, программирование с использованием среды Hibernate. Главы 6, “Обзор среды Spring Framework”, и 7, “Среда Spring Web MVC Framework”, посвящены среде Spring Framework. Главы 7, “Среда Spring Web MVC Framework”, и 8, “Феномен Eclipse!”, – это то, что я называю “ого!”, поскольку здесь все собирается вместе и вы понимаете, почему мы прошли предыдущие главы именно так, как это было сделано. Вы узнаете, что происходит внутри, а следовательно, получите представление об основе таких технологий, как Spring, Hibernate, Ant и JUnit. Далее мы рассмотрим некоторые дополнительные концепции и закончим книгу приложениями, содержащими полезные сведения.

Еще одно замечание по поводу использования при разработке командной строки, а не GUI (например, использования SDK Eclipse). В первых главах командная строка используется преднамеренно для того, чтобы вы могли получить некоторое представление об основных правилах работы этих инструментальных средств Java. Зато впоследствии, когда будете использовать эти инструменты (например, Ant и JUnit) в IDE типа Eclipse, вы будете точно знать, что происходит за кулисами. Это становится особенно важным, если IDE не удовлетворяет вашим потребностям.

О коде для этой книги

Эта книга о программировании в Java, поэтому она содержит большое количество кода, относящегося к примерам приложений. Данное полнофункциональное приложение со всем исходным кодом и связанными файлами может быть разгружено с Web-сайта издателя (http://www.samspublishing.com/content/images/0672328968/downloads/0672328968_AgileJavaDev_code.zip). Получив код в исходной электронной форме, вместо подвергнутого многократной правке и напечатанного в книге, можно избежать возможных опечаток.

Код в книге рассматривается в виде фрагментов полного кода. Кроме того, для этого кода используются два соглашения:

- В некоторых местах, когда строка кода слишком длинная (т.е. больше чем 82 символа), для удобочитаемости она разделяется на две строки символом ↵.
- Код представлен моноширинным шрифтом.

Примечание

Весь код книги доступен для загрузки на Web-сайте издателя. Для удобства доступа к файлам примеров этой книги и исходному коду, а также к любым возможным обновлениям или исправлениям, зарегистрируйтесь по адресу www.samspublishing.com/register.

Об иллюстрациях для этой книги

Как вполне справедливо подмечено, лучше один раз увидеть, чем сто раз услышать, поэтому в этой книге вы найдете множество схем и снимков экрана. Хотя я и привел в этой книге несколько диаграмм UML, разрабатывая приложение, я предпочитаю более простые схемы в свободной форме. Я не трачу много времени на формальные схемы, поскольку большинство из них становится не нужно, после того как они выполнили свою задачу, а кроме того, в большинстве проектов такие схемы даже на это не способны. По моему глубокому убеждению, код и база данных в окончательном виде значительно важнее других артефактов.

Таким образом, я полагаю, что схемы должны удовлетворять следующим требованиям:

- Схемы нужны только тогда, когда они эффективны.
- Схемы должны быть понятны (используйте только понятные слова, применяйте легенды).
- Схемы должны быть просты и действительно отражать суть дела.
- Придерживайтесь стандартной формы записи (например, UML), если таково требование вашей организации или если схему предстоит передать другому человеку или группе для создания объектного кода или просто потому, что вы предпочитаете использовать стандартную форму записи.

Еще одно замечание об иллюстрациях в этой книге. В некоторых главах я несколько раз использовал иллюстрации из предыдущих глав. Это было сделано по следующим причинам.

- Чтобы задать контекст главы или раздела, который предстоит обсудить.
- Мне не хотелось беспокоить вас требованиями вернуться к предыдущей главе, чтобы посмотреть приведенную там иллюстрацию.

Вставки с личным мнением

Текст книги содержит многочисленные вставки, озаглавленные “Личное мнение”, — именно таково содержимое этих вставок. Я постарался отделить этот субъективный материал от объективного материала книги. Однако надеюсь, что вам моя точка зрения, выраженная в этих разделах, будет полезна, поскольку в них содержатся обобщения моего личного опыта как: инженера по программному обеспечению, консультанта, преподавателя, автора, создателя сетевого сообщества и даже руководителя компании, занимающейся технологиями IT.

Комикс

В начале каждой главы вы увидите иллюстрации, составляющие вымышленную историю о восьминедельном проекте с использованием AMDD. Четыре главных действующих лица этой истории, включая клиента (Сьюзен), руководителя проекта (Рон) и двух программистов (Стив и Радж), также вымышленные. Цель этих иллюстраций проста: придать книге немного юмора. Кроме того, свободный стиль этих иллюстраций — следствие моего опыта написания книг для детей (моих первых читателей). Предупреждаю, юмор немного банальный, но я надеюсь, что вы его оцените.

Если стиль этих иллюстраций вам понравился и вы хотите посмотреть их в цвете, посетите, пожалуйста, Web-сайт visualpatterns.com.

Рекомендуемые ресурсы

Здесь описано множество технологий, однако, с учетом характера этой книги, о каждой технологии будет предоставляться только та информация, которая достаточна для практической работы. Но каждая из описанных здесь технологий сама по себе заслуживает отдельной книги, и такие книги действительно имеются (во всяком случае по большинству технологий).

Поэтому ниже приведены Web-сайты, посвященные основным технологиям, описанным в этой книге. Каждый из них предоставляет дополнительную информацию (а в некоторых случаях и форумы для обсуждения) по соответствующей технологии:

- Agile Modeling <http://www.agilemodeling.com>
- Ant <http://ant.apache.org/>
- Eclipse SDK <http://eclipse.org>
- Extreme Programming <http://extremeprogramming.org>
- Hibernate Framework <http://hibernate.org>
- Механизм баз данных HSQLDB <http://hsqldb.org/>
- JUnit <http://junit.org>
- Spring Framework <http://springframework.org>
- Visual Patterns <http://visualpatterns.com>

В конце каждой главы я буду предоставлять ссылки на соответствующие ресурсы, так что к концу этой книги у вас будет хорошая коллекция ресурсов для дальнейшего изучения!

Соглашения, принятые в книге

При оформлении книги использованы соглашения, общепринятые в компьютерной литературе.

- Новые термины в тексте выделяются *курсивом*. Чтобы привлечь внимание читателя на отдельные фрагменты текста, также применяется *курсив*.
- Текст программ, функций, переменных, URL Web-страниц и другой код представлен моноширинным шрифтом.
- Все, что придется вводить с клавиатуры, выделено **полужирным моноширинным** шрифтом.
- Знакоместо в описаниях синтаксиса выделено *курсивом*. Это указывает на необходимость заменить знакоместо фактическим именем переменной, параметром или другим элементом, который должен находиться на этом месте $BINDSIZE = (\text{максимальная ширина колонки}) * (\text{номер колонки})$.
- Пункты меню и названия диалоговых окон представлены следующим образом: Menu Option (Пункт меню).

От издательства

Вы, читатель этой книги, и есть главный ее критик. Мы ценим ваше мнение и хотим знать, что было сделано нами правильно, что можно было сделать лучше и что бы еще вы хотели увидеть изданным нами. Нам интересно услышать и любые другие замечания, которые вам хотелось бы высказать авторам.

Мы ждем ваших комментариев. Вы можете прислать письмо по электронной почте или просто посетить наш Web-сервер, оставив на нем свои замечания, — одним словом, любым удобным для вас способом дайте нам знать, нравится ли вам эта книга, а также выскажите свое мнение о том, как сделать наши книги более подходящими для вас.

Посылая письмо или сообщение, не забудьте указать название книги и ее авторов, а также ваш e-mail. Мы внимательно ознакомимся с вашим мнением и обязательно учтем его при отборе и подготовке к изданию следующих книг. Наши координаты:

E-mail: info@williamspublishing.com

WWW: <http://www.williamspublishing.com>