

# SPRING 3

**ДЛЯ ПРОФЕССИОНАЛОВ**

# Pro SPRING 3

Clarence Ho  
Rob Harrop

Apress®

# SPRING 3

**ДЛЯ ПРОФЕССИОНАЛОВ**

Кларенс Хо  
Роб Харроп



Москва • Санкт-Петербург • Киев  
2013

ББК 32.973.26-018.2.75  
Х68  
УДК 681.3.07

Издательский дом “Вильямс”  
Зав. редакцией *С.Н. Тригуб*  
Перевод с английского *Ю.Н. Артеменко*  
Под редакцией *Ю.Н. Артеменко*

По общим вопросам обращайтесь в Издательский дом “Вильямс” по адресу:  
[info@williamspublishing.com](mailto:info@williamspublishing.com), <http://www.williamspublishing.com>

**Хо**, Кларенс, **Харроп**, Роб.

X68 Spring 3 для профессионалов. : Пер. с англ. — М. : ООО “И.Д. Вильямс”, 2013. — 880 с. : ил. — Парал. тит. англ.

ISBN 978-5-8459-1803-1 (рус.)

**ББК 32.973.26-018.2.75**

Все названия программных продуктов являются зарегистрированными торговыми марками соответствующих фирм.

Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотокопирование и запись на магнитный носитель, если на это нет письменного разрешения издательства APress, Berkeley, CA.

Authorized translation from the English language edition published by APress, Inc., Copyright © 2012 by Clarence Ho and Rob Harrop.

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

Russian language edition is published by Williams Publishing House according to the Agreement with R&I Enterprises International, Copyright © 2013.

*Научно-популярное издание*

**Кларенс Хо, Роб Харроп**

## Spring 3 для профессионалов

Верстка *Т.Н. Артеменко*  
Художественный редактор *В.Г. Павлютин*

Подписано в печать 28.08.2012. Формат 70×100/16.  
Гарнитура Times. Печать офсетная.  
Усл. печ. л. 70,95. Уч.-изд. л. 57,3.  
Тираж 1500 экз. Заказ № 0000.

Первая Академическая типография “Наука”  
199034, Санкт-Петербург, 9-я линия, 12/28

ООО “И. Д. Вильямс”, 127055, г. Москва, ул. Лесная, д. 43, стр. 1

ISBN 978-5-8459-1803-1 (рус.)

ISBN 978-1-43-024107-2 (англ.)

© Издательский дом “Вильямс”, 2013

© by Clarence Ho and Rob Harrop, 2012

# Оглавление

Об авторах	17
Глава 1. Введение в Spring	19
Глава 2. Начало работы	33
Глава 3. Пример приложения	57
Глава 4. Введение в IoC и DI в Spring	73
Глава 5. Детальные сведения о конфигурации Spring	135
Глава 6. Введение в аспектно-ориентированное программирование с использованием Spring	203
Глава 7. Дополнительные сведения об аспектно-ориентированном программировании в Spring и аннотациях	253
Глава 8. Поддержка JDBC в Spring	293
Глава 9. Использование Hibernate в Spring	339
Глава 10. Доступ к данным в Spring с использованием JPA 2	365
Глава 11. Использование MyBatis в Spring	415
Глава 12. Проектирование и реализация Spring-приложений	453
Глава 13. Управление транзакциями	475
Глава 14. Проверка достоверности с преобразованием типов и форматированием	509
Глава 15. Планирование задач в Spring	537
Глава 16. Использование удаленной обработки в Spring	553
Глава 17. Разработка веб-приложений в Spring	595
Глава 18. Проект Spring Web Flow и JSF	669
Глава 19. Тестирование в Spring	707
Глава 20. Проекты Spring: Spring Batch, Spring Integration и Spring Roo	735
Глава 21. Дополнительные сведения о примере приложения	769
Глава 22. Поддержка написания сценариев в Spring	809
Глава 23. Мониторинг приложений Spring	829
Приложение A. SpringSource Tool Suite (STS)	853
Предметный указатель	871

# Содержание

<b>Об авторах</b>	17
О техническом рецензенте	18
От издательства	18
<b>Глава 1. Введение в Spring</b>	19
Что такое Spring?	19
Инверсия управления или внедрение зависимостей?	20
Эволюция внедрения зависимостей	21
По ту сторону внедрения зависимостей	23
Проект Spring	29
Происхождение Spring	29
Сообщество Spring	29
Spring для Microsoft .NET	30
Комплект SpringSource Tool Suite / IDE-среда Spring	30
Проект Spring Security	30
Проекты Spring Batch и Spring Integration	31
Другие проекты	31
Альтернативы Spring	31
JBoss Seam Framework	31
Google Guice	31
PicoContainer	32
Контейнер JEE 6	32
Резюме	32
<b>Глава 2. Начало работы</b>	33
Получение Spring Framework	34
Загрузка стандартного дистрибутива	34
Извлечение Spring из GitHub	34
Упаковка Spring	35
Модули Spring	35
Выбор модулей для разрабатываемого приложения	37
Модули Spring в репозитории Maven	38
Анализ зависимостей Spring	39
Примеры приложений	43
Получение исходного кода примеров Spring	43
Приложение Petclinic	44
Приложение Petclinic Groovy	44
Приложение jPetStore	44
Приложение Spring Configuration Basic	45
Приложение Spring Task and Scheduling	45
Приложение Spring MVC Showcase	45
Приложения Spring MVC Basic и Spring MVC Ajax	45
Приложение Spring Petcare	46
Примеры приложений для Spring Web Flow	46
Документация Spring	46
Помещение Spring в приложение “Hello World!”	47
Построение примера приложения “Hello World!”	47
Создание проекта Spring в STS	51
Резюме	55

<b>Глава 3. Пример приложения</b>	57
Требования для приложения SpringBlog	58
Безопасность и аутентификация	58
Просмотр записей блога	59
Помещение записей в блог	60
Комментирование записи блога	60
Фильтрация нежелательного контента	62
Присоединение файлов к записи блога или к ее комментарию	63
Аудит действий в блоге	63
RSS-лента	63
Загрузка блога из XML-файла	64
Реализация приложения SpringBlog	64
Инструменты разработки и управление зависимостями	64
Проектное решение для приложения	65
Управление конфигурацией приложения	65
Многоуровневая архитектура приложения SpringBlog	66
Реализация уровня постоянства	66
Реализация уровня обслуживания	68
Использование АОП для фильтрации нежелательного контента	68
Резюме	71
<b>Глава 4. Введение в IoC и DI в Spring</b>	73
Инверсия управления и внедрение зависимостей	74
Типы инверсии управления	74
Тип Dependency Pull	74
Тип Contextualized Dependency Lookup	75
Тип Constructor Dependency Injection	76
Тип Setter Dependency Injection	77
Выбор между внедрением и поиском	77
Выбор между Setter Injection и Constructor Injection	78
Инверсия управления в Spring	81
Внедрение зависимостей с помощью Spring	82
Бины и фабрики бинов	82
Реализации BeanFactory	83
Интерфейс ApplicationContext	85
Конфигурирование ApplicationContext	85
Варианты конфигурации Spring (XML и Java-аннотации)	85
Объявление компонентов Spring	87
Использование внедрения через метод установки	91
Использование внедрения через конструктор	92
Параметры внедрения	97
Использование внедрения через метод	112
Именованые бинов	121
Режим создания экземпляров бинов	123
Разрешение зависимостей	127
Автосвязывание бина	128
Различные режимы автосвязывания	128
Когда используется автосвязывание	130
Наследование бинов	130
Резюме	132

<b>Глава 5. Детальные сведения о конфигурации Spring</b>	<b>135</b>
Влияние Spring на переносимость приложений	136
Управление жизненным циклом бинов	137
Привязка к созданию бина	138
Привязка к уничтожению бина	146
Преобразование бинов в “осведомленные о платформе Spring”	153
Использование интерфейса <code>BeanNameAware</code>	154
Использование интерфейса <code>ApplicationContextAware</code>	155
Использование фабрик бинов	157
Пример специальной фабрики бинов: класс <code>MessageDigestFactoryBean</code>	158
Доступ к фабрике бинов напрямую	161
Использование атрибутов <code>factory-bean</code> и <code>factory-method</code>	162
Редакторы свойств для компонентов <code>JavaBean</code>	163
Встроенные редакторы свойств	164
Создание специального редактора свойств	169
Дополнительные сведения о конфигурации <code>ApplicationContext</code>	171
Интернационализация с помощью интерфейса <code>MessageSource</code>	172
Использование <code>MessageSource</code> в автономных приложениях	176
Интерфейс <code>MessageSourceResolvable</code>	176
События приложений	176
Доступ к ресурсам	179
Конфигурация, использующая Java-классы	181
Конфигурирование <code>ApplicationContext</code> в Java	181
Выбор между конфигурациями Java и XML	187
Профили	187
Пример использования средства профилей Spring	187
Соображения по поводу использования профилей	193
Абстракции <code>Environment</code> и <code>PropertySource</code>	193
Конфигурация, использующая аннотации JSR-330	197
Резюме	201
<b>Глава 6. Введение в аспектно-ориентированное программирование с использованием Spring</b>	<b>203</b>
Концепции АОП	205
Типы АОП	206
Статическое АОП	206
Динамическое АОП	206
Выбор типа АОП	206
АОП в Spring	207
Альянс АОП	207
Пример “Hello World!” в АОП	208
Архитектура АОП в Spring	210
Класс <code>ProxyFactory</code>	211
Создание совета в Spring	212
Советы и срезы в Spring	228
Интерфейс <code>Pointcut</code>	229
Что собой представляют прокси	245
Использование динамических прокси JDK	245
Использование прокси CGLIB	246
Сравнение производительности прокси	246
Какой прокси использовать?	250
Резюме	251



<b>Глава 7. Дополнительные сведения об аспектно-ориентированном программировании в Spring и аннотациях</b>	<b>253</b>
Расширенное использование срезов	254
Использование срезов потока управления	254
Использование компоуемых срезов	257
Компоновка и интерфейс <code>Pointcut</code>	260
Резюме по созданию срезов	260
Работа с введениями	261
Основы введения	261
Обнаружение модификации объекта с помощью введений	264
Резюме по введениям	269
Службы платформы, предназначенные для АОП	269
Декларативное конфигурирование АОП	269
Использование <code>ProxyFactoryBean</code>	270
Использование пространства имен <code>aop</code>	275
Использование аннотаций в стиле <code>@AspectJ</code>	280
Соображения по поводу декларативного конфигурирования АОП в Spring	284
Интеграция с <code>AspectJ</code>	285
Что собой представляет <code>AspectJ</code>	285
Использование одиночных экземпляров аспектов	286
АОП в примере приложения	290
Фильтрация нежелательного контента в <code>SpringBlog</code>	290
Резюме	292
<b>Глава 8. Поддержка JDBC в Spring</b>	<b>293</b>
Модель данных для кода примеров	294
Исследование инфраструктуры JDBC	297
Инфраструктура JDBC в Spring	302
Обзор пакетов JDBC в Spring	303
Подключения к базе данных и источники данных	303
Поддержка встроенной базы данных	307
Использование источников данных в классах DAO	308
Обработка исключений	309
Класс <code>JdbcTemplate</code>	311
Инициализация <code>JdbcTemplate</code> в классе DAO	311
Извлечение одиночного значения с использованием класса <code>JdbcTemplate</code>	312
Использование именованных параметров с помощью <code>NamedParameterJdbcTemplate</code>	313
Извлечение объектов предметной области с помощью <code>RowMapper&lt;T&gt;</code>	314
Извлечение вложенных объектов предметной области с помощью <code>ResultSetExtractor</code>	316
Классы Spring, моделирующие операции JDBC	318
Настройка DAO-классов JDBC с использованием аннотаций	318
Запрашивание данных с использованием <code>MappingSqlQuery&lt;T&gt;</code>	321
Обновление данных с использованием <code>SqlUpdate</code>	325
Вставка данных и извлечение сгенерированного ключа	327
Объединение операций в пакеты с помощью <code>BatchSqlUpdate</code>	329
Вызов хранимых функций с использованием <code>SqlFunction</code>	333
Использование Java-конфигурации	336
Проект Spring Data: расширения JDBC	337
Соображения по поводу использования JDBC	337
Резюме	338

## 10 Содержание

<b>Глава 9. Использование Hibernate в Spring</b>	<b>339</b>
Создание служебного проекта Hibernate в STS	340
Модель данных для кода примера	342
Конфигурирование фабрики сеансов Hibernate	344
Объектно-реляционное отображение с использованием аннотаций Hibernate	346
Простое отображение	347
Отображение “один ко многим”	351
Отображение “многие ко многим”	352
Интерфейс <i>Session</i> в Hibernate	354
Выполнение операций базы данных с помощью Hibernate	354
Запрашивание данных с использованием языка запросов Hibernate	355
Вставка данных	360
Обновление данных	362
Удаление данных	362
Соображения по поводу использования Hibernate	363
Резюме	364
<b>Глава 10. Доступ к данным в Spring с использованием JPA 2</b>	<b>365</b>
Введение в JPA 2	366
Создание простого служебного проекта Spring JPA в STS	367
Модель данных для кода примеров	369
Конфигурирование <i>EntityManagerFactory</i> в JPA	369
Отображение ORM с использованием аннотаций JPA	371
Устранение уровня DAO	371
Внедрение диспетчера сущностей в классы уровня обслуживания	372
Операции базы данных в JPA	373
Запрашивание данных с использованием языка запросов постоянства Java	374
Вставка данных	381
Обновление данных	383
Удаление данных	384
Собственный запрос	385
Запрос с критерием, использующий API-интерфейс критериев JPA 2	387
Введение в проект Spring Data JPA	392
Добавление библиотечных зависимостей Spring Data JPA	392
Операции базы данных, использующие абстракцию <i>Repository</i> из проекта Spring Data JPA	394
Отслеживание изменений в сущностном классе	398
Отслеживание версий сущностей с использованием Hibernate Envers	405
Добавление зависимостей Hibernate Envers	406
Добавление таблиц для отслеживания версий сущностей	406
Конфигурирование <i>EntityManagerFactory</i> для отслеживания версий сущностей	407
Кодирование изменений для отслеживания версий сущностей и извлечения хронологии	409
Тестирование отслеживания версий сущностей	411
Соображения по поводу того, когда использовать JPA	412
Использование JPA в примере приложения	412
База данных серверной части	412
Использование JPA для реализации уровня постоянства	413
Аудит и отслеживание версий сущностей	413
Резюме	413

<b>Глава 11. Использование MyBatis в Spring</b>	415
Начало работы с MyBatis в Spring	416
Введение в MyBatis	416
Создание простого служебного проекта с поддержкой MyBatis в STS	416
Модель данных для кода примеров	418
Конфигурирование <code>SqlSessionFactory</code> и <code>MapperScannerConfigurer</code> в MyBatis	419
SQL-отображение в MyBatis	421
Интерфейсы отображателей и файлы SQL-отображения	422
XML-конфигурация SQL-отображения	424
Операции базы данных в MyBatis	424
Запрашивание данных	424
Вставка данных	440
Обновление данных	443
Удаление данных	448
Соображения по поводу того, когда использовать MyBatis	450
Использование MyBatis в примере приложения	450
База данных серверной части	450
Использование MyBatis для реализации уровня постоянства	451
Аудит и отслеживание версий сущностей	451
Резюме	451
<b>Глава 12. Проектирование и реализация Spring-приложений</b>	453
Проектирование интерфейсов	454
Причины выбора подхода с проектированием интерфейсов	454
Шаблон “Фабрика”	455
Влияние Spring на проектирование, основанное на интерфейсах	458
Построение объектной модели предметной области	459
Spring и объектная модель предметной области	459
DOM — это не то же самое, что объект значения	459
Причины создания объектной модели предметной области	460
Моделирование объектов предметной области	461
Инкапсулировать поведение или нет?	462
Объектная модель предметной области для примера приложения SpringBlog	463
Итоги по объектной модели предметной области	468
Проектирование и построение уровня доступа к данным	469
Практические соображения по поводу проектирования	470
Итоги по уровню доступа к данным	471
Проектирование уровня обслуживания	472
Для чего нужен уровень обслуживания?	472
Проектирование бизнес-интерфейсов	473
Итоги по уровню обслуживания	474
Резюме	474
<b>Глава 13. Управление транзакциями</b>	475
Исследование уровня абстракции транзакций Spring	476
Типы транзакций	476
Реализации интерфейса <code>PlatformTransactionManager</code>	478
Анализ свойств транзакций	479
Интерфейс <code>TransactionDefinition</code>	479
Интерфейс <code>TransactionStatus</code>	481
Модель данных и инфраструктура для кода примеров	481
Создание простого служебного проекта Spring JPA с зависимостями	481

## 12 Содержание

Модель данных и общие классы	483
Декларативные и программные транзакции в Spring	485
Использование аннотаций для управления транзакциями	486
Использование XML-конфигурации для управления транзакциями	494
Использование программных транзакций	497
Соображения по поводу управления транзакциями	500
Глобальные транзакции в Spring	500
Инфраструктура для реализации примера применения JTA	500
Реализация глобальных транзакций с помощью JTA	501
Соображения по поводу использования диспетчера транзакций JTA	508
Резюме	508
<b>Глава 14. Проверка достоверности с преобразованием типов и форматированием</b>	<b>509</b>
Создание проекта для примеров в STS	510
Система преобразования типов Spring	510
Преобразование из String с использованием редакторов свойств	510
Появление в Spring 3 системы преобразования типов	514
Форматирование полей в Spring 3	519
Реализация специального форматировщика	519
Конфигурирование ConversionServiceFactoryBean	520
Проверка достоверности в Spring	522
Использование интерфейса Validator в Spring	523
Использование JSR-303: Bean Validation API	525
Какой API-интерфейс проверки достоверности использовать?	532
Проверка достоверности с преобразованиями типов и форматированием в примере приложения	533
Резюме	535
<b>Глава 15. Планирование задач в Spring</b>	<b>537</b>
Создание проекта для примеров в STS	537
Реализация планирования задач в Spring	538
Введение в абстракцию TaskScheduler	538
Пример задачи	540
Планирование задач с использованием пространства имен task	544
Планирование задач с использованием аннотаций	546
Асинхронное выполнение задач в Spring	547
Планирование задач в примере приложения	550
Резюме	551
<b>Глава 16. Использование удаленной обработки в Spring</b>	<b>553</b>
Создание проекта для примеров в STS	554
Реализация уровня обслуживания для примеров	554
Добавление обязательных зависимостей для серверной части JPA	555
Проверка работоспособности проекта	556
Модель данных для примеров	557
Реализация и конфигурирование интерфейса ContactService	559
Использование HTTP-активатора Spring	564
Отображение службы	564
Вызов службы	565
Использование JMS в Spring	566
Установка сервера ActiveMQ	567

Реализация слушателя JMS в Spring	569
Отправка сообщений JMS в Spring	571
Использование веб-служб REST в Spring	573
Введение в веб-службы REST	574
Добавление обязательных зависимостей для примеров	575
Веб-службы REST для контактной информации	575
Использование Spring MVC для отображения веб-служб REST	576
Использование curl для тестирования веб-служб REST	583
Использование класса RestTemplate для доступа к веб-службам REST	584
Защита веб-служб REST с помощью Spring Security	588
Использование JSR-303 с веб-службами REST	592
Удаленная обработка в примере приложения	593
Резюме	594
<b>Глава 17. Разработка веб-приложений в Spring</b>	<b>595</b>
Создание проекта для примеров в STS	597
Реализация уровня обслуживания для примеров	597
Модель данных для примеров	597
Реализация и конфигурирование интерфейса ContactService	598
Введение в MVC и Spring MVC	603
Введение в MVC	603
Введение в Spring MVC	605
Создание первого представления в Spring MVC	610
Конфигурирование сервлета диспетчера	611
Реализация класса ContactController	611
Реализация представления списка контактов	612
Тестирование представления списка контактов	613
Обзор структуры проекта Spring MVC	614
Интернационализация	616
Конфигурирование интернационализации в сервлете диспетчера	616
Модификация представления списка контактов для поддержки интернационализации	618
Оформление темами и применение шаблонов	619
Поддержка оформления темами	620
Применение шаблонов представлений с помощью Apache Tiles	622
Реализация представлений для информации о контактах	628
Отображение URL на представления	628
Реализация представления просмотра контакта	629
Реализация представления редактирования контакта	632
Реализация представления добавления контакта	636
Включение проверки достоверности бинов JSR-303	637
Использование jQuery и jQuery UI	640
Введение в jQuery и jQuery UI	641
Активизация jQuery и jQuery UI в представлении	641
Редактирование форматированного текста с помощью SKEditor	643
Построение сетки данных, поддерживающей разбиение на страницы, с использованием jqGrid	645
Обработка загрузки файлов	651
Конфигурирование поддержки загрузки файлов	652
Изменение представлений для поддержки загрузки файлов	653
Изменение контроллера для поддержки загрузки файлов	654

## 14 Содержание

Защита веб-приложения с помощью Spring Security	656
Конфигурирование Spring Security	657
Добавление к приложению функций входа	659
Использование аннотаций для защиты методов контроллера	662
Поддержка конфигурации на основе кода для Servlet 3	663
Spring MVC в примере приложения	665
Реализация MVC для приложения SpringBlog	665
Насыщенный пользовательский интерфейс и Ajax	666
Поддержка безопасности	666
Поддержка Servlet 3.0	667
Резюме	667
<b>Глава 18. Проект Spring Web Flow и JSF</b>	<b>669</b>
Проект примера серверной части	669
Уровень обслуживания для примера серверной части	670
Импорт проекта примера серверной части в STS	670
Введение в Spring Web Flow	672
Модули Spring Web Flow	672
Функциональные возможности Spring Web Flow	673
Введение в JSF	674
Представление	675
Взаимодействие с моделью	675
Навигация	676
Жизненный цикл приложения	676
Пример приложения Spring Web Flow	677
Проектирование примера потока	677
Структура проекта	678
Конфигурация Spring Web Flow и JSF	679
Добавление обязательных зависимостей	679
Конфигурирование JSF	679
Конфигурирование дескриптора веб-развертывания	680
Конфигурирование Spring Web Flow и Spring MVC	682
Реализация примера потока	684
Создание определения потока	684
Реализация шаблонной страницы	687
Реализация специального преобразователя	688
Реализация контроллера и поддерживающего бина	689
Реализация представления просмотра контакта	695
Реализация потока добавления контакта	697
Шаг 1: ввод базовой информации	697
Шаг 2: выбор хобби	701
Шаг 3: просмотр введенной информации	703
Шаг 4: добавление контакта завершено	704
Резюме	706
<b>Глава 19. Тестирование в Spring</b>	<b>707</b>
Проект для примера веб-приложения	708
Импорт проекта примера серверной части в STS	708
Введение в корпоративную инфраструктуру тестирования	710
Реализация модульного тестирования логики	712
Добавление обязательных зависимостей	712
Модульное тестирование контроллера Spring MVC	713

Реализация модульного тестирования взаимодействия	718
Добавление обязательных зависимостей	718
Конфигурирование профиля для тестирования уровня обслуживания	719
Реализация классов инфраструктуры	720
Модульное тестирование уровня обслуживания	724
Реализация модульного тестирования интерфейсной части	728
Добавление обязательных зависимостей	728
Введение в Selenium	728
Реализация тестового сценария для пользовательского интерфейса	729
Контроль покрытия кода тестовыми сценариями	731
Резюме	733
<b>Глава 20. Проекты Spring: Spring Batch, Spring Integration и Spring Roo</b>	<b>735</b>
Проект для примеров этой главы	736
Импорт проекта примера серверной части в STS	736
Введение в Spring Batch	737
Поток и процесс пакетного задания	737
Компоненты инфраструктуры Spring Batch	738
Метаданные Spring Batch	738
Политики выполнения заданий	740
Реализация пакетного задания	741
Добавление обязательных зависимостей	741
Конфигурирование инфраструктуры Spring Batch	742
Реализация задания импорта контактов	743
Использование Spring Batch и Spring Integration	751
Введение в Spring Integration	752
Добавление обязательных зависимостей	753
Реализация механизма опроса файлов	753
Введение в Spring Roo	757
Конфигурирование Spring Roo в STS	758
Создание проекта Spring Roo	758
Настройка уровня постоянства и сущностного класса	758
Настройка уровня обслуживания	763
Настройка уровня презентаций	764
Дополнение Spring Roo	767
Соображения по поводу Spring Roo	768
Spring Batch и Spring Integration в примере приложения	768
Резюме	768
<b>Глава 21. Дополнительные сведения о примере приложения</b>	<b>769</b>
Настройка примера приложения	770
Настройка проекта	770
Переключение между базами данных MySQL и H2	773
Переключение между реализациями JPA и MyBatis	775
Проект приложения	776
Модель данных	776
Объектная модель предметной области	777
Модель UML	779
Подробности конфигурирования	785
Детали реализации	792
Реализация уровня обслуживания	792
Фильтрация нежелательного контента с использованием АОП	797

## 16 Содержание

Планирование задания для очистки данных аудита	800
Уровень презентаций	801
Резюме	807
<b>Глава 22. Поддержка написания сценариев в Spring</b>	<b>809</b>
Проект для примеров этой главы	810
Создание простого служебного проекта Spring	810
Установка подключаемого модуля Groovy для Eclipse	810
Поддержка написания сценариев в Java	813
Введение в Groovy	815
Динамическая типизация	815
Упрощенный синтаксис	816
Замыкание	817
Использование Groovy в Spring	818
Добавление обязательных зависимостей	819
Предметная область, связанная с контактами	819
Реализация процессора правил	820
Реализация фабрики правил как обновляемого бина Spring	823
Тестирование правила возрастной категории	825
Резюме	827
<b>Глава 23. Мониторинг приложений Spring</b>	<b>829</b>
Проект для примеров этой главы	830
Поддержка JMX в Spring	832
Экспорт бина Spring в JMX	832
Настройка VisualVM для мониторинга JMX	833
Мониторинг вошедших пользователей	835
Мониторинг статистики Hibernate	840
Мониторинг пакетных заданий Spring	842
Мониторинг приложения с помощью Spring Insight	845
Введение в Spring Insight	845
Конфигурирование Spring Insight	846
Использование Spring Insight	848
Резюме	851
<b>Приложение A. SpringSource Tool Suite (STS)</b>	<b>853</b>
Введение в STS	853
Установка STS	854
Установка автономной версии STS	854
Установка STS в существующей среде Eclipse	858
Настройка проекта и управление зависимостями	862
Создание простого служебного проекта Spring	862
Управление зависимостями для проекта	866
Использование STS	867
Установка расширений STS	867
Конфигурирование VMware tc Server в STS	869
Резюме	870
Предметный указатель	871
Содержание	872
<b>Предметный указатель</b>	<b>871</b>



*Всем моим друзьям и коллегам, с которыми я работал, за предоставленную неоценимую поддержку, помощь и великолепный опыт работы. Моей семье, особенно моей маме Энз, за настоящую любовь и горячую поддержку всех моих решений. Также команде авторов, включая Криса, Мануэля и Брента, за необыкновенную поддержку при написании этой книги.*

— Кларенс Хо

## Об авторах

**Кларенс Хо** — ведущий Java-архитектор в фирме SkywideSoft Technology Limited ([www.skywidesoft.com](http://www.skywidesoft.com)), занимающейся консультациями по программному обеспечению и расположенной в Гонконге. Работая в сфере информационных технологий более 20 лет, Кларенс выступал в качестве главы команды во многих проектах по разработке приложений на дому, а также предоставлял услуги консультаций по производственным решениям своим клиентам. Кларенс начал программировать на Java в 2001 г., но уже с 2005 г. был вовлечен в проектирование и разработку JEE-приложений с такими технологиями, как EJB, Spring Framework, Hibernate, JMS, WS и т.д. С тех пор он выступал в качестве Java-архитектора корпоративных приложений.

В настоящее время Кларенс работает консультантом в международном финансовом учреждении, принимая участие в различных областях, в числе которых архитектурное проектирование Java EE, обучение, предоставление рекомендаций по технологическим решениям и рекомендуемые приемы разработки приложений.

В свободное от работы время Кларенс занимается спортом (бегом трусцой, плаванием, футболом, пешеходным туризмом), чтением, просмотром фильмов, общением с друзьями и т.п.

**Роб Харроп** является соучредителем SpringSource — компании, которая стоит за успешным проектом Spring Framework. В настоящее время он занимает должность руководителя технического отдела в First Banco. До SpringSource Роб был соучредителем и техническим директором в Cake Solutions (Манчестер, Соединенное Королевство). Он специализируется на крупных масштабируемых корпоративных системах. В настоящее время он занимает должность руководителя технического отдела в First Banco. До SpringSource Роб был соучредителем и техническим директором в Cake Solutions (Манчестер, Соединенное Королевство). Он специализируется на крупных масштабируемых корпоративных системах.

Роб выступал автором и соавтором в более чем 5 книгах. Он доступен в Твиттере как [@robertharrop](https://twitter.com/robertharrop).

## О техническом рецензенте

**Мануэль Джордан Элера** — разработчик-самоучка и исследователь, которому нравится изучать новые технологии для собственных экспериментов и создавать новые интеграции.

Мануэль выиграл звание 2010 Springy Award — Community Champion. В свободное от работы время (которого не особенно много) он читает Библию и сочиняет музыку на своей гитаре. Он является ведущим участником на форумах сообщества Spring (Spring Community Forums) и известен там как *dr\_pompeii*.

Мануэль выступал техническим рецензентом в следующих книгах, выпущенных издательством Apress:

- *Pro SpringSource dm Server* (2009 г.)
- *Spring Enterprise Recipes* (2009 г.)
- *Spring Recipes (Second Edition)* (2010 г.)
- *Pro Spring Integration* (2011 г.)
- *Pro Spring Batch* (2011 г.)
- *Pro Spring MVC: With Web Flow* (2012 г.)

Найти его можно в собственном блоге по адресу <http://manueljordan.wordpress.com/>, а также в Твиттере как @dr\_pompeii.

## От издательства

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

Мы ждем ваших комментариев и надеемся на них. Вы можете прислать нам бумажное или электронное письмо, либо просто посетить наш веб-сервер и оставить свои замечания там. Одним словом, любым удобным для вас способом дайте нам знать, нравится или нет вам эта книга, а также выскажите свое мнение о том, как сделать наши книги более интересными для вас.

Посылая письмо или сообщение, не забудьте указать название книги и ее авторов, а также ваш обратный адрес. Мы внимательно ознакомимся с вашим мнением и обязательно учтем его при отборе и подготовке к изданию последующих книг.

Наши координаты:

E-mail: [info@williamspublishing.com](mailto:info@williamspublishing.com)

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

Информация для писем из:

России: 127055, г. Москва, ул. Лесная, д. 43, стр. 1

Украины: 03150, Киев, а/я 152

# ГЛАВА 1

## Введение в Spring

**К**огда мы думаем о сообществе разработчиков на языке Java, мы вспоминаем полчища старателей, которые в конце 1840-х годов неистово прочесывали реки Северной Америки в поисках золотых самородков. Наши “реки” — реки Java-разработчиков — изобилуют проектами с открытым кодом, но, в отличие от истории со старателями, нахождение действительно полезного проекта может оказаться длительным по времени и трудным.

Многим проектам с открытым кодом Java характерна общая особенность — они были призваны просто заполнить пробел в реализации последней “модной” технологии или шаблона. С учетом сказанного, многие высококачественные и полезные проекты предназначены для решения реальных потребностей в реальных приложениях, и в этой книге вы встретите ряд таких проектов. С одним из них вы ознакомитесь достаточно подробно — и это, как не трудно догадаться, Spring.

На протяжении всей книги вы будете сталкиваться со многими применениями разнообразных технологий с открытым кодом, причем все они объединены в платформу Spring Framework. Благодаря Spring, разработчик приложений может пользоваться широким спектром инструментов с открытым кодом, не занимаясь написанием больших объемов кода и не привязывая создаваемое приложение слишком тесно к какому-то конкретному инструменту.

В этой главе вы ознакомитесь с платформой Spring Framework без просмотра крупных примеров или изучения подробных объяснений. Если вы уже знакомы с проектом Spring, можете сразу переходить к чтению главы 2.

### Что такое Spring?

Возможно, самой трудной частью объяснения Spring как технологии является точная классификация того, что Spring собой представляет. Обычно Spring описывают как облегченную платформу для построения Java-приложений, но с этим утверждением связаны два интересных момента. Во-первых, Spring можно использовать для построения любого приложения на языке Java (т.е. автономных, веб-приложений, приложений JEE и т.д.), что отличает Spring от многих других платформ, таких как Apache Struts, которая ограничена только веб-приложениями. Во-вторых, характеристика “облегченная” в действительности не имеет никакого отношения к количеству классов или размеру дистрибутива; напротив, она определяет принцип всей философии Spring — минимальное воздействие. Платформа Spring является облегченной в том смысле, что для использования ядра Spring вы должны вносить минимальные (если вообще какие-либо) изменения в код своего приложения, а если в какой-то момент вы решите больше не пользоваться Spring, то и это сделать очень просто. Обратите внимание, что речь идет только о ядре Spring — многие дополнительные компоненты Spring, такие как доступ к данным, требуют более тесной привязки к Spring Framework. Однако польза от такой

привязки вполне очевидна, и практически повсеместно в книге мы будем представлять приемы минимизации влияния этой привязки на разрабатываемые приложения.

## Инверсия управления или внедрение зависимостей?

Ядро Spring Framework основано на принципе инверсии управления (Inversion of Control — IoC), при котором создание и управление зависимостями между компонентами становятся внешними. Рассмотрим пример, в котором класс Foo при выполнении обработки определенного вида зависит от экземпляра класса Bar. Традиционно Foo создает экземпляр Bar, используя операцию new или получая нужный экземпляр от некоторого фабричного класса. Согласно подходу IoC, экземпляр класса Bar (или его подкласса) предоставляется классу Foo во время выполнения некоторым внешним процессом. Такое поведение, т.е. внедрение зависимостей во время выполнения, было переименовано Мартином Фаулером на более описательное — внедрение зависимостей (Dependency Injection — DI). Точная природа зависимостей, управляемых DI, обсуждается в главе 4.

---

**На заметку!** Как будет показано в главе 4, использование термина внедрение зависимостей при ссылке на инверсию управления всегда корректно. В контексте Spring эти термины можно применять взаимозаменяемо, совершенно не теряя при этом смысла.

---

Реализация DI в Spring основана на двух ключевых концепциях Java — компонентах JavaBean и интерфейсах. При использовании Spring в качестве поставщика DI вы получаете гибкость определения конфигурации зависимостей внутри своих приложений разнообразными путями (т.е. внешне в XML-файлах, с помощью конфигурационных Java-классов Spring или посредством аннотаций Java в коде). Компоненты JavaBean (также называемые POJO (Plain Old Java Object — простой старый объект Java)) предоставляют стандартный механизм для создания ресурсов Java, которые являются конфигурируемыми множеством способов. В главе 4 вы увидите, как Spring использует спецификацию JavaBean для формирования ядра своей модели конфигурации DI; фактически на любой ресурс, управляемый Spring, ссылаются как на бин (bean). На тот случай, если вы еще не знакомы с технологией JavaBean, в начале главы 4 будет приведен краткий пример.

Интерфейсы и DI — это взаимовыгодные технологии. Мы уверены, что любой читатель этой книги согласится с утверждением о том, что проектирование и написание кода интерфейсов предназначено для гибких приложений, но сложность сборки вместе приложения, использующего интерфейсы, довольно высока и требует от разработчиков дополнительных усилий по кодированию. За счет применения DI объем кода, который необходим при проектировании приложения на основе интерфейсов, снижается почти до нуля. Кроме того, с помощью интерфейсов можно получить максимальную отдачу от DI, потому что бины могут использовать любую реализацию интерфейса для удовлетворения их зависимости.

В контексте внедрения зависимостей Spring действует больше подобно контейнеру, чем платформе, предоставляя экземпляры классов вашего приложения со всеми необходимыми зависимостями, но делает это гораздо менее навязчивым способом. Использование Spring для DI основано всего лишь на следовании внутри своих классов соглашениям об именовании, принятым для компонентов JavaBean (требование, которое, как будет показано в главе 5, можно обойти, воспользовавшись поддержкой внедрения через метод в Spring); ни специальных классов, предназначенных для наследования, ни собственных схем именования, которые должны соблюдаться, не предусмотрено. Во всяком случае, единственным изменением, которое делается в приложении, использующем DI, является открытие доступа к дополнительным свойствам ваших компонентов JavaBean, что позволяет внедрять больше зависимостей во время выполнения.