

Оглавление

Об авторе	12
О рецензенте.....	13
Предисловие.....	14
Темы, рассматриваемые в книге	14
Что нужно для чтения книги.....	16
Принятые обозначения.....	16
Скачивание кода примеров.....	17
Об ошибках	17
Глава 1. Знакомство с Spring Framework 5.0 и паттернами проектирования.....	18
Знакомство с фреймворком Spring	18
Упрощение разработки приложений благодаря применению Spring и паттернов	20
Использование широчайших возможностей паттерна POJO	21
Внедрение зависимостей между POJO	22
Использование паттерна внедрения зависимостей для зависимых компонентов	25
Применение объектов для сквозных задач	29
Применение шаблонов для устранения стереотипного кода	33
Использование контейнеров Spring для управления компонентами с помощью паттерна «Фабрика».....	36
Фабрики компонентов	36
Контексты приложений	37
Создание контейнера с контекстом приложения.....	37
Жизнь компонента в контейнере	38
Модули фреймворка Spring.....	40
Core Container Spring	41
Модуль AOP.....	42
Spring DAO — доступ к данным и интеграция	42
ORM	42
Web MVC	42
Новые возможности Spring Framework 5.0	43
Резюме	44
Глава 2. Обзор паттернов проектирования GoF: базовые паттерны проектирования ...	45
Возможности паттернов проектирования.....	46
Обзор часто используемых паттернов проектирования GoF	47

Порождающие паттерны проектирования	48
Паттерн проектирования «Фабрика»	49
Паттерн проектирования «Абстрактная фабрика»	52
Паттерн проектирования «Одиночка»	58
Паттерн проектирования «Прототип»	60
Паттерн проектирования «Строитель»	63
Резюме	66
Глава 3. Соображения по поводу структурных и поведенческих паттернов	68
Базовые паттерны проектирования	68
Структурные паттерны проектирования	69
Поведенческие паттерны проектирования	94
Паттерны проектирования J2EE	103
Резюме	104
Глава 4. Связывание компонентов с помощью паттерна внедрения зависимостей...	105
Паттерн внедрения зависимостей	106
Решение проблем с помощью паттерна внедрения зависимостей	106
Виды внедрения зависимостей	111
Внедрение зависимостей через конструктор	111
Внедрение зависимости через сеттер	113
Сравнение внедрений через конструктор и сеттер, а также рекомендуемые практики	115
Описание конфигурации паттерна внедрения зависимостей с помощью Spring	115
Использование паттерна внедрения зависимостей с Java-конфигурацией	117
Создание класса Java-конфигурации: AppConfig.java	117
Объявления компонентов Spring в классе конфигурации	117
Внедрение компонентов Spring	118
Оптимальный подход к настройке паттерна внедрения зависимостей с помощью Java	119
Использование паттерна внедрения зависимостей с XML-конфигурацией	120
Создание файла XML-конфигурации	120
Объявление компонентов Spring в XML-файле	121
Внедрение компонентов Spring	121
Использование паттерна внедрения зависимостей с конфигурацией на основе аннотаций	124
Что такое стереотипные аннотации	124
Автосвязывание зависимостей и неоднозначности	131
Рекомендуемые практики для конфигураций паттерна DI	135
Резюме	137
Глава 5. Жизненный цикл компонентов и используемые паттерны	138
Жизненный цикл компонента Spring и его фазы	139
Фаза инициализации	140
Фаза использования компонентов	150
Фаза уничтожения компонента	151

Области видимости компонентов	153
Одиночная область видимости	154
Прототипная область видимости компонента	155
Сеансовая область видимости компонента	155
Запросная область видимости компонента	155
Другие области видимости в Spring	156
Резюме	158
Глава 6. Аспектно-ориентированное программирование в Spring с помощью паттернов «Заместитель» и «Декоратор»	159
Паттерн «Заместитель» в Spring	160
Что такое сквозная функциональность	162
Что такое аспектно-ориентированное программирование	162
Проблемы, решаемые с помощью AOP	163
Как AOP решает проблемы	166
Основные понятия и терминология AOP	167
Совет	167
Точка соединения	168
Срез	169
Аспект	169
Вплетение	169
Задание срезов	170
Создание аспектов	172
Реализация советов	174
Тип совета: до	174
Тип совета: после возврата	175
Тип совета: после исключения	176
Тип совета: после	177
Тип совета: везде	178
Описание аспектов с помощью XML-конфигурации	180
AOP-прокси	181
Резюме	183
Глава 7. Доступ к базе данных с помощью фреймворка Spring и JDBC-реализаций паттерна «Шаблонный метод»	184
Оптимальный подход к проектированию доступа к данным	185
Задача управления ресурсами	187
Реализация паттерна проектирования «Шаблонный метод»	188
Настройка источника данных и паттерн «Пул объектов»	192
Задание настроек источника данных с помощью JDBC-драйвера	193
Конфигурирование источника данных с помощью пула соединений	194
Реализация паттерна «Строитель» для создания встроенного источника данных ..	196
Абстрагирование доступа к базе данных с помощью паттерна DAO	196
Реализация паттерна DAO с помощью фреймворка Spring	197
Работа с JdbcTemplate	198
Когда использовать JdbcTemplate	199

Рекомендуемые практики JDBC и настройки JdbcTemplate	204
Резюме	205
Глава 8. Доступ к базе данных с помощью паттернов ORM и транзакций	206
Фреймворки ORM и используемые в них паттерны.....	207
Управление ресурсами и транзакциями	209
Единообразная обработка и трансляция исключений	209
Паттерн «Объект доступа к данным»	210
Создание объектов DAO в Spring с помощью паттерна проектирования «Фабрика»	211
Паттерн «Отображение данных»	212
Паттерн «Модель предметной области»	213
Прокси для паттерна «Отложенная загрузка»	214
Паттерн «Шаблонный метод» для поддержки Hibernate в Spring	214
Интеграция Hibernate со Spring	214
Задание настроек объекта SessionFactory фреймворка Hibernate в контейнере Spring	214
Реализация объектов DAO на основе простого API Hibernate	216
Стратегии управления транзакциями в Spring.....	217
Декларативное задание границ и реализация транзакций.....	219
Развертывание диспетчера транзакций	220
Программное задание границ и реализация транзакций.....	223
Рекомендуемые практики для ORM Spring и модуля транзакций приложения	225
Резюме	226
Глава 9. Улучшение производительности приложения с помощью паттернов кэширования	227
Что такое кэш.....	228
Абстракция кэша	228
Включение возможности кэширования посредством паттерна «Заместитель»	229
Включение прокси для кэширования с помощью аннотаций	230
Включение прокси для кэширования с помощью пространства имен XML.....	231
Декларативное кэширование с помощью аннотаций.....	232
Аннотация @Cacheable	232
Аннотация @CachePut	233
Аннотация @CacheEvict	235
Аннотация @Caching	236
Аннотация @CacheConfig.....	236
Декларативное кэширование с помощью XML.....	237
Настройка хранилища кэша	240
Сторонние диспетчеры кэша.....	240
EhCache.....	240
XML-конфигурация	241
Создание пользовательских аннотаций кэширования	242
Лучшие рекомендуемые практики для веб-приложений.....	242
Резюме	244

Глава 10. Реализация паттерна MVC в веб-приложениях с помощью фреймворка Spring	245
Реализация паттерна MVC в веб-приложении	246
Архитектура «Модель 2» паттерна MVC в Spring	247
Паттерн проектирования «Единая точка входа»	248
Включение возможностей MVC Spring	257
Реализация контроллеров	259
Отображение запросов с помощью аннотации @RequestMapping	260
Передача данных модели представлению	264
Принятие параметров запроса	265
Обработка форм веб-страницы	268
Реализация контроллера обработки форм	270
Привязка данных с помощью паттерна проектирования «Команда»	272
Проверка корректности входных параметров форм	275
Реализация компонента «Представление» в паттерне MVC	277
Описание арбитра представлений в MVC Spring	278
Паттерн «Вспомогательный компонент представления»	281
Паттерн «Составное представление» и использование арбитра представлений фреймворка Apache Tiles	283
Рекомендуемые практики проектирования веб-приложений	285
Резюме	286
Глава 11. Реализация реактивных паттернов проектирования	288
Изменение требований к приложениям с течением времени	288
Паттерн «Реактивность»	290
Отличительные признаки паттерна «Реактивность»	290
Блокирующие вызовы	296
Неблокирующие вызовы	296
Контроль обратного потока данных	297
Реализация реактивности с помощью фреймворка Spring 5.0	298
Реактивный веб-модуль Spring	299
Реализация реактивного веб-приложения на стороне сервера	300
Модель программирования на основе аннотаций	301
Функциональная модель программирования	303
Реализация реактивного приложения на стороне клиента	308
Преобразование типов тела запроса и ответа	310
Резюме	311
Глава 12. Реализация конкурентных паттернов	312
Паттерн «Активный объект»	313
Паттерн проектирования «Монитор»	314
Паттерны «Полусинхронность» и «Полусинхронность»	315
Паттерн «Ведущий/ведомые»	316
Паттерн «Реактор»	317
Паттерн «Локальная память потока выполнения»	319
Резюме	320