

# Содержание

Об авторе .....	xv
О техническом обозревателе .....	xvii
Благодарности .....	xix
Введение .....	xxi

## ЧАСТЬ 1 ■ ■ ■ Groovy в примерах

<b>ГЛАВА 1</b>	<b>Начало работы с Groovy</b> .....	3
	1-1. Что такое Groovy? .....	3
	1-2. Что не так с Java? .....	4
	1-3. Как Groovy устраняет недостатки Java? .....	5
	1-4. Как загрузить и установить Groovy? .....	9
	1-5. Какие инструменты поставляются вместе с Groovy? .....	10
	1-6. Как использовать Groovy-оболочку? .....	10
	1-7. Как использовать Groovy-консоль? .....	11
	1-8. Как использовать groovus и groovy? .....	12
	1-9. Есть ли поддержка Groovy в IDE? .....	13
	1-10. Как добавить поддержку Groovy в Eclipse? .....	13
	1-11. Как добавить поддержку Groovy в IntelliJ IDEA? .....	14
	Резюме .....	16
<b>ГЛАВА 2</b>	<b>От Java к Groovy</b> .....	17
	2-1. Чем схожи Java и Groovy? .....	17
	2-2. Чем отличаются Java и Groovy? .....	19
	Необязательные синтаксические элементы .....	19
	Новые и усовершенствованные синтаксические элементы, структуры и конструкции .....	22

	Новые хелперы, библиотеки и APIs	29
	Другие отличия	33
2-3.	Как интегрировать Groovy с Java?	38
	Компиляция в байткод	39
	Использование GroovyShell	39
	Использование GroovyScriptEngine	40
	Использование GroovyClassLoader	41
	Использование JSR 223	43
	Резюме	44
<b>ГЛАВА 3</b>	<b>Типы данных Groovy и управляющие структуры</b>	<b>45</b>
3-1.	Какие виды строк есть в Groovy и как их использовать?	45
3-2.	Как использовать регулярные выражения в Groovy?	51
3-3.	Чем отличаются числа в Groovy и Java?	54
3-4.	Как использовать списки в Groovy?	58
3-5.	Как реализовать сортировку слиянием в Groovy?	62
3-6.	Как использовать карты в Groovy?	63
3-7.	Что такое диапазоны и как их использовать в Groovy?	66
3-8.	Что является истинностным значением в Groovy?	67
3-9.	Чем отличается оператор switch в Groovy и Java?	68
3-10.	Как организовать циклы в Groovy?	69
	Резюме	70
<b>ГЛАВА 4</b>	<b>Объектно-ориентированный Groovy</b>	<b>73</b>
4-1.	Чем отличаются классы и скрипты?	73
	Один общедоступный класс на файл	74
	Несколько классов на файл	74
	Только скриптовый код	75
	Классы и скриптовый код в одном файле	75
	Выбор стратегии	76
4-2.	Как использовать пакеты?	76
4-3.	Что такое синонимия типов и как её использовать?	77
4-4.	Как использовать наследование в Groovy?	78
4-5.	Как использовать интерфейсы в Groovy?	79

4-6. Что такое мультиметоды и как их использовать? .....	82
4-7. Что такое категории и как их использовать? .....	85
4-8. Чем отличаются поля и локальные переменные в Groovy и Java? ..	85
4-9. Чем отличаются методы в Groovy и Java? .....	87
Использование позиционных параметров. ....	88
Использование списка в качестве единственного аргумента ....	89
Использование массива для необязательных параметров. ....	89
Использование параметров в виде карты. ....	90
4-10. Чем отличаются конструкторы в Groovy и Java? .....	90
4-11. Что такое GroovyBeans? .....	91
4-12. Что такое GPaths? .....	93
4-13. Как использовать класс Expando? .....	95
4-14. Что такое Metaclass и как его использовать? .....	96
4-15. Как перехватить все вызовы методов объекта? .....	97
4-16. Как перехватить методы, которые отсутствуют в классе? .....	98
4-17. Как привнести в класс дополнительное поведение с помощью ExpandoMetaClass? .....	98
Резюме. ....	99

## ■ ГЛАВА 5

<b>Замыкания</b> .....	101
5-1. Что такое замыкание? .....	101
5-2. Зачем нужны замыкания? .....	102
5-3. Можно ли сравнить замыкания с анонимными внутренними классами? .....	103
5-4. Как создать замыкание? .....	104
5-5. Как вызвать замыкание? .....	104
5-6. Как вернуть значение из замыкания? .....	105
5-7. Как повторно использовать метод в качестве замыкания? .....	105
5-8. Как передать замыкание в качестве аргумента метода? .....	106
5-9. Какова область видимости замыкания? .....	107
5-10. Что означают зарезервированные слова this, owner и delegate внутри замыкания? .....	109
5-11. Как из замыкания вернуть результат? .....	110

5-12. Что означает каррирование замыканий? .....	111
5-13. Как использовать замыкание внутри оператора switch? .....	113
5-14. Как получить дополнительную информацию о параметрах, переданных в замыкание? .....	113
5-15. Как использовать замыкания внутри карт? .....	114
5-16. Как использовать замыкания с файлами? .....	114
Резюме .....	115

## ■ ГЛАВА 6

<b>Билдеры</b> .....	117
6-1. Что такое билдеры? .....	117
6-2. Зачем нужны билдеры? .....	118
6-3. Как использовать MarkupBuilder для генерации XML? .....	121
6-4. Как использовать MarkupBuilder для генерации HTML? .....	125
6-5. Как с помощью NodeBuilder создать дерево объектов? .....	127
6-6. Как использовать ObjectGraphBuilder для генерации дерева объектов? .....	128
6-7. Как использовать AntBuilder для генерации Ant-заданий? .....	130
6-8. Как с помощью SwingBuilder создать Swing-виджеты? .....	131
6-9. Как использовать менеджеры разметки вместе со SwingBuilder? ..	135
6-10. Как добавить экшен к Swing-виджету? .....	137
6-11. Как совместно использовать экшены из виджетов? .....	139
6-12. Как использовать Swing-модели? .....	140
6-13. Как создать свой собственный билдер? .....	142
Резюме .....	145

## ■ ГЛАВА 7

<b>Работа с базами данных</b> .....	147
7-1. Как установить соединение с базой данных? .....	147
7-2. Как использовать пул соединений? .....	149
7-3. Как создать новую таблицу? .....	149
7-4. Как добавить, обновить и удалить данные? .....	151
7-5. Как извлечь данные из таблиц? .....	152
7-6. Как получить метаданные таблицы? .....	155
7-7. Как использовать DataSet? .....	156
7-8. Как использовать DataSet с объединением таблиц? .....	158
Резюме .....	161

<b>ГЛАВА 8</b>	<b>Тестирование с Groovy</b> .....	163
	8-1. Как в Groovy создать встраиваемый тест? .....	163
	8-2. Как в Groovy создать тестовый класс? .....	164
	8-3. Как использовать Groovy для тестирования Java-кода? .....	168
	8-4. Как объединить и выполнить тесты из IDE? .....	169
	8-5. Как использовать Ant для выполнения тестов? .....	172
	8-6. Как использовать Maven для выполнения тестов? .....	173
	8-7. Какие передовые методики тестирования предоставляет Groovy? .....	175
	8-8. Как использовать карты для тестирования кода? .....	175
	8-9. Как использовать объект Expando для тестирования кода? .....	177
	8-10. Как использовать заглушки и макеты в Groovy? .....	178
	8-11. Как использовать GroovyLogTestCase? .....	182
	8-12. Как измерить покрытие кода с помощью Cobertura? .....	184
	Резюме .....	189
<b>ГЛАВА 9</b>	<b>Разнообразные советы</b> .....	191
	9-1. Как использовать шаблоны Groovy для генерации динамического повторно используемого содержимого? .....	191
	9-2. Как использовать грувелеты для генерации динамического Web-содержимого? .....	195
	9-3. Как загрузить и обработать XML с помощью XmlParser? .....	197
	9-4. Как загрузить и обработать XML с помощью XmlSlurper? .....	201
	9-5. Как использовать XPath? .....	203
	9-6. Как загрузить XML-содержимое RSS-ленты? .....	204
	9-7. Как использовать Groovy из командной строки? .....	205
	9-8. Как использовать ConfigSlurper для создания конфигурационных файлов? .....	206
	9-9. Как с помощью Groovy запустить внешний процесс? .....	209
	9-10. Как в Groovy загрузить файл? .....	209
	9-11. Как обработать все файлы в каталоге? .....	210
	9-12. Как подсчитать все вхождения слова в строке? .....	211
	Резюме .....	212

## ЧАСТЬ 2 ■ ■ ■ Grails в примерах

<b>ГЛАВА 10</b>	<b>Начало работы с Grails</b> .....	215
	10-1. Что такое Grails? .....	215
	10-2. Зачем нужен ещё один фреймворк? .....	216
	10-3. Как загрузить и установить Grails? .....	217
	10-4. Как в Grails создать первое приложение? .....	218
	10-5. Как использовать Grails в Eclipse? .....	221
	10-6. Как использовать Grails в IntelliJ IDEA? .....	222
	10-7. Какие основные команды есть в Grails? .....	224
	Резюме .....	225
<b>ГЛАВА 11</b>	<b>Web-уровень</b> .....	227
	11-1. Как создать контроллер? .....	228
	11-2. Что такое серверные страницы Groovy? .....	229
	11-3. Как связаны контроллеры и GSPs? .....	231
	11-4. Как передать переменные из контроллера в GSP? .....	232
	11-5. Как использовать теги в качестве вызовов методов? .....	233
	11-6. Как в контроллере создать несколько экшенов? .....	234
	11-7. Какие неявные объекты доступны внутри контроллера и GSP? ..	235
	11-8. Как для пользователя создать несколько представлений? .....	242
	11-9. Как последовательно вызвать экшены? .....	244
	11-10. Как перехватить экшены контроллера? .....	245
	11-11. Как выполнить привязку входных параметров? .....	247
	11-12. Как вернуть в качестве результата JSON? .....	248
	11-13. Как сохранить доменные классы в виде XML или JSON (сериализовать)? .....	249
	11-14. Как выгрузить и загрузить файлы? .....	250
	11-15. Что такое шаблоны и как их использовать? .....	251
	11-16. Как изменить разметку и внешний вид приложения? .....	252
	11-17. Как создать собственные пользовательские теги? .....	257
	11-18. Как использовать фильтры? .....	258
	11-19. Как использовать Ajax? .....	260
	Резюме .....	262

<b>ГЛАВА 12</b>	<b>Уровень данных</b> .....	263
	12-1. Как настроить использование базы данных? .....	263
	12-2. Как создать доменный класс? .....	267
	12-3. Как установить отношения? .....	271
	12-4. Как использовать композицию? .....	278
	12-5. Как выполнять CRUD-операции доменных классов? .....	279
	12-6. Как выполнять запросы с использованием GORM? .....	282
	12-7. Как использовать динамические файндеры? .....	283
	12-8. Как использовать критерии? .....	285
	12-9. Как использовать HQL? .....	288
	12-10. Как использовать наследование? .....	289
	12-11. Что такое оптимистическое и пессимистическое блокирование? .....	291
	12-12. Как использовать события? .....	292
	12-13. Как использовать метки времени? .....	293
	12-14. Как использовать кэширование? .....	294
	12-15. Как использовать пользовательский идентификатор базы данных? .....	296
	12-16. Как использовать составной первичный ключ? .....	296
	12-17. Как добавить индекс к полю? .....	297
	Резюме .....	297
<b>ГЛАВА 13</b>	<b>Скаффолдинг</b> .....	299
	13-1. Как использовать динамический скаффолдинг? .....	300
	13-2. Как динамически выполнить скаффолдинг отношений? .....	303
	13-3. Как настроить созданные представления? .....	306
	13-4. Какие в Grails есть встроенные ограничения? .....	310
	13-5. Как переопределить в рамках скаффолдинга экшены и представления? .....	312
	13-6. Как использовать статический скаффолдинг? .....	317
	13-7. Как изменить шаблоны скаффолдинга? .....	321
	13-8. Как добавить собственный редактор свойств? .....	322
	13-9. Как использовать скаффолдинг с классами, отображёнными с помощью Hibernate? .....	326
	Резюме .....	328

# Введение

**П**латформа Java топчется на месте довольно долгое время. При этом язык программирования Java начинает устаревать. Сейчас для Java-разработчиков наступает время к переходу в сторону мышления в терминах динамических языков. Groovy является одним из самых лучших динамических языков доступных на платформе Java. После нескольких лет работы с Groovy я твёрдо убедился, что всем Java-разработчикам следует по крайней мере познакомиться с Groovy. Количество кода, которое можно сократить с помощью динамического языка подобного Groovy, действительно поражает, особенно, работая с коллекциями и файлами. Именно по этой причине я решил написать эту книгу. Я хочу поделиться с Java-разработчиками огромным выигрышем в эффективности, которая достигается с помощью Groovy.

Динамические языки подобные Groovy делают реальностью такие фреймворки как Grails. Grails – это глоток свежего воздуха для Java-разработчиков и это одна из основных причин моей заинтересованности динамическими языками. Я помню первые дни разработки на Java с использованием Struts и Tapestry. Но я по ним не скучаю. Лично мне эти фреймворки всегда казались излишне сложными. Я просто не мог определить объёмы конфигурационного и шаблонного кода, необходимые для того, чтобы хоть что-то сделать. Это не те фреймворки, о которых стоит думать. Фреймворки должны делать вещи проще и позволять сосредоточиться на логике. Это то, что делает Grails. Grails является осмысленной технологией, что для меня является первым, что я ищу в новой технологии. Grails является столь простым и одновременно мощным фреймворком, что нельзя не удивляться, почему о нём никто не подумал раньше.

Одна из сильных сторон Groovy и Grails – это то, что виртуальная машина Java является для них «родной». Учитывая вездесущность Java в наше время, было бы глупо предлагать Java-разработчикам отказаться от всей Java-инфраструктуры, APIs, библиотек и фреймворков и начать всё сначала. По этой причине Groovy и Grails обязаны быть успешными в мире корпоративных приложений, где платформа Java успешно обосновалась. Их идеальная интеграция с Java – огромное подспорье в распространении. В одной из организаций мы обсуждали, следует ли использовать Ruby on Rails или Groovy и Grails. К концу дня победу одержали Groovy и Grails. Их отличное взаимодействие с Java и лёгкость изучения для Java-разработчиков стали ключевыми факторами, повлиявшими на окончательное решение.

В этой книге у меня две цели. Во-первых, на практике научить Groovy и Grails с нуля. Во-вторых, предоставить практические решения частых проблем в Groovy и Grails. Я хотел бы, чтобы Вы могли взять книгу и, выбрав интересующую проблему, быстро найти

подходящее решение. Вы никогда не найдёте подробных теоретических выкладок о том, как же всё устроено изнутри, а непосредственно небольшие фрагменты кода для решения проблемы.

Я надеюсь, Вы получите такое же удовольствие от прочтения книги, как я при её написании. С Groovy и Grails действительно удобно работать. Я не припомню, чтобы получал от работы с технологией такое удовольствие, как от работы с Groovy и Grails.

## Для кого эта книга

Эта книга нацелена напрямую Java-разработчикам. Она не подразумевает предварительного знакомства с Groovy и Grails. И будет преподносить материал в виде вопросов и ответов. У разработчиков, не использовавших Java, но знакомых с динамическими языками (такими как Ruby, PHP или Python), также не должно возникнуть проблем при чтении книги, хотя Java-разработчики, вероятно, извлекут из неё больше пользы.

Grails – это больше, чем Web-фреймворк. Это множество приложений, которые собирают вместе другие технологии, а именно: Hibernate, Spring и SiteMesh. Хотя по этим технологиям и не требуется предварительных знаний, читателям, которые уже знакомы с ними, будет легче понять соответствующие советы, чем тем, которые их никогда не использовали.

## Как организована эта книга

Эта книга разделена на 16 глав с помощью подхода в виде вопросов и ответов.

Я всегда был поклонник книг в виде советов. Они раскрывают суть без напрасной траты времени читателя. Именно это делает данная книга. Она состоит из двух частей: Groovy и Grails. Часть, посвящённая Groovy, включает первые девять глав.

Глава 1 представляет краткое введение в Groovy, показывает варианты его использования и по шагам объясняет, как загрузить и установить Groovy на компьютере.

Глава 2 в основном предназначена для того, чтобы облегчить Java-разработчикам переход от синтаксиса Java к Groovy. Показаны главные сходства и отличия.

Глава 3 обсуждает типы данных и управляющие структуры. Типы данных Groovy включают простые типы данных и коллекции. Управляющие структуры состоят из циклических структур и условных структур.

Глава 4 показывает Groovy с объектно-ориентированной стороны. Groovy – это полностью объектно-ориентированный язык, поэтому Java-разработчики будут чувствовать себя как дома.

Глава 5 касается темы, которая, возможно, наиболее трудна для понимания Java-разработчиками: замыканий. На примерах эта глава пытается раскрыть загадку замыканий и показать варианты их использования.

Глава 6 демонстрирует другой новый для Java-разработчиков элемент: билдер. Билдеры появляются там, где проявляются динамические возможности Groovy, а эффективность стремительно повышается.

Глава 7 полностью посвящена базам данных и тому, как значительно Groovy упрощает JDBC API.

Конечно же, ни одна книга по программированию не является завершённой без упоминания о тестировании. Глава 8 посвящена тестированию и показывает, как Вы можете использовать динамические возможности Groovy для тестирования классов с множеством внешних зависимостей.

Глава 9 завершает часть, посвящённую Groovy, разнообразными советами. Работа с XML, файлами, командной строкой и регулярными выражениями – некоторые из обсуждаемых тем.

Глава 10 начинает часть, посвящённую Grails, демонстрирует варианты использования Grails, показывает как его загрузить и установить, а также по шагам объясняет, как создать первое приложение Hello World.

Глава 11 затрагивает Web-уровень, который в Grails состоит из двух основных частей: контроллеров и представлений. Эта глава даёт советы по выполнению общих задач данного уровня.

Глава 12 касается уровня данных, где выполняется сохранение доменных классов в базу данных. Для этого Grails использует Hibernate, но делает это на основе нового предметно-ориентированного языка, называемого GORM, который существенно упрощает работу с Hibernate.

Глава 13 раскрывает тему, которая может быть в новинку для Java-разработчиков: скаффолдинг. Как будет видно, скаффолдинг – это больше, чем просто генерация кода.

Глава 14 затрагивает важную тему для любого реального приложения: безопасность. В ней показывается, как защитить приложение от частых атак и добавить поддержку аутентификации и авторизации.

При всей важности тестирования я решил добавить ещё одну главу по этой теме. В ней показывается, как выполнять тестирование Grails-артефактов. Как будет видно в главе 15, в отличие от тестирования большинства Web-приложений тестировать Grails-приложения действительно просто.

В последней 16 главе смешены разнообразные советы по Grails. Сервисный уровень, Web-сервисы и журналирование – некоторые из обсуждаемых тем.

## Предварительные требования

Эта книга охватывает версию Groovy 1.5.6 и версию Grails 1.0.3. Каждая является последней стабильной версией на момент написания книги. Эта книга по шагам описывает процесс установки Groovy и Grails. Единственным предварительным требованием для установки Groovy и Grails является наличие установленного комплекта JDK 5.0 или выше.