



# Spring Framework

## Module 1 – Введение

Evgeniy Krivosheev  
Andrey Stukalenko  
Vyacheslav Yakovenko  
Last update: Feb, 2012

# Spring Framework – Знакомство

- Spring – легковесный, но в тоже время гибкий и всеобъемлющий фреймворк для создания Java SE и Java EE приложений;
- Spring – фреймворк с открытым исходным кодом;
- Spring – фреймворк приложения, а не одного его слоя;
- Spring содержит в себе несколько отдельных фреймворков;

# Spring Framework – Знакомство

- Род Джонсон (Rod Johnson) создал Spring в 2003г;
- Spring «вырос» из книг «Expert One on One Java J2EE Design and Development» и «J2EE Development Without EJB»;
- Главная идея Spring – упростить традиционный подход к разработке J2EE приложений;

# Spring Framework – Знакомство

- В настоящее время Spring Framework разрабатывается под эгидой SpringSource, которая является подразделением VMWare ([vmware.com](http://vmware.com))

# Spring Framework – Знакомство

- Основные ресурсы:
  - [springsource.com](http://springsource.com)
  - [blog.springsource.com](http://blog.springsource.com)
  - @SpringSource (<https://twitter.com/#!/SpringSource>)
  - [infoq.com/spring](http://infoq.com/spring)

# Spring Framework – Знакомство

## Spring помогает:

- Создать логичную архитектуру;
- Облегчить разработку J2EE приложений;
- Соединить фреймворки разных уровней;
- Эффективно организовать middle-tier объекты, как с использованием EJB, так и без;
- Уменьшить кол-во синглтонов и property-файлов;
- Упростить процесс тестирования;
- Использовать разные сервера приложений / сервлет-контейнеры (WebLogic, Tomcat, Resin, JBoss, Jetty, Geronimo, WebSphere);

# Spring Framework – Знакомство

## Достоинства Spring:

- Не внедряется в код – легко мигрировать на другой контейнер или следующую версию Spring;
- Можно вести разработку без связи с классами фреймворка, также можно использовать библиотеки, не использующие Spring;
- Не обязательно наличие сервера приложений;
- Способствует повторному использованию кода;
- Приветствует ООД в J2EE приложениях, предполагает Design to Interface а также DDD и DMD;
- Рассматривает приложение как взаимодействие именованных сервисов (не путать с SOA);
- Позволяет перерабатывать реализацию сервиса, не затрагивая все приложение;

# Spring Framework – Знакомство

## Достоинства Spring:

- Способствует выделению конфигурационных значений в отдельные xml или property-файлы;
- POJO классы могут быть протестированы отдельно без тяжеловесного контейнера;
- Использует хорошие готовые решения (ORM, logging abstraction, connection pool, remoting protocols, etc.);
- Упрощает использование этих технологий, и объединяет их для создания целостной архитектуры.



# Spring Framework – Знакомство

## Недостатки обычного J2EE подхода:

- EJB is overused;
  - Распределенность не должна быть встроена в компонентную модель;
- Многие J2EE-паттерны являются обходными путями (workarounds);
- J2EE приложения сложно тестировать;

## Достоинства обычного J2EE подхода:

- EJB – это стандарт;
- Поддержка многих крупных вендоров;
- Широкое распространение среди разработчиков;
- Множество вспомогательных средств разработки;

# Spring Framework – Знакомство

## Сравнение EJB и Spring:

- Для большинства J2EE приложений подходят и Spring, и EJB
- Распределенные транзакции – лучше EJB, но можно использовать Spring + JTA

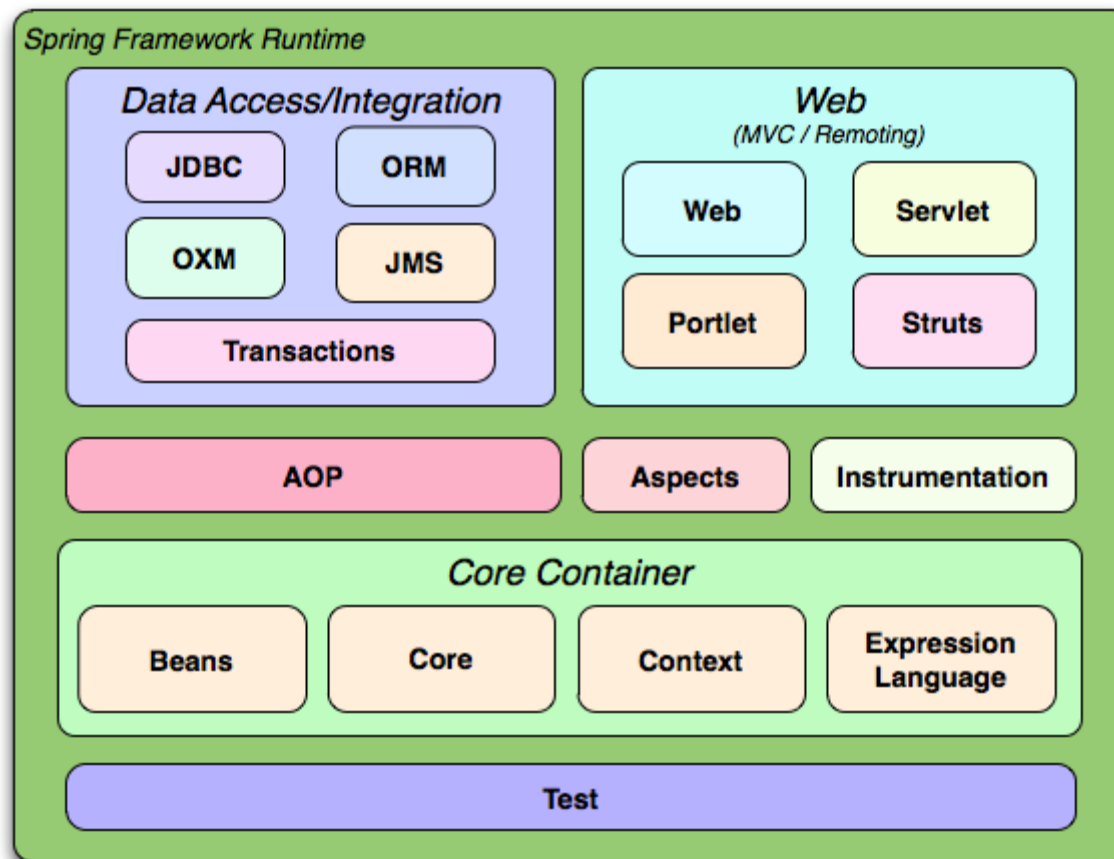
# Spring Framework – Знакомство

JDK	Spring 1.2.x	Spring 2.0.x	Spring 2.5.x	Spring 3.x	Spring 3.1
1.3	Да	Да, но могут быть недоступны некоторые функции из JDK 5	Нет	Нет	Нет
1.4	Да	Да, но могут быть недоступны некоторые функции из JDK 5			Нет
5	Не использует	Поддерживает, но не полностью использует	Более полное использование	Да	Да
6	Не использует	Нет	Да	Да	Да
7	Нет	Нет	Нет	Нет	Да

# Spring Framework :: Структура фреймворка

- SpringSource выпускает целую коллекцию продуктов::
  - **Spring Framework** – основной фреймворк, предоставляющий ключевую базовую функциональность;
  - **Spring Security** – мощный настраиваемый фреймворк, упрощающий реализацию авторизации и контроля доступа и предоставляющий возможность декларативного управления безопасностью;
  - **Spring WebServices (Spring WS)** – фреймворк, упрощающий создание document-driven веб-сервисов с использованием SOAP;
  - **Spring Mobile** – расширение Spring MVC (часть Core) для разработки мобильных веб-приложений;
  - **Spring Data** – поддержка технологий доступа к данным;
  - **Spring Batch** – фреймворк, упрощающий обработку пакетных операций;
  - **Spring WebFlow** – фреймворк, направленный на поддержку создания многошаговых, «процессных» веб-приложений;
  - **Spring BlazeDS Integration** – фреймворк, упрощающий создание RIA и интеграцию с Adobe FLEX
  - и т.д.

# Spring Framework :: Структура фреймворка



# Spring Framework :: Ядро

## ■ Core + Beans

- Inversion of Control (IoC) и Dependency Injection (DI);
- Описание зависимостей;
- Реализация базового IoC контейнера;
- Устраняет необходимость создания синглтонов программистом;
- Позволяет разделить конфигурацию зависимостей и логику программы;

## ■ Context

- Доступ к объектам;
- Интернационализация (internationalization - I18N);
- Оповещение о событиях (event-propagation);
- Загрузка ресурсов (resource-loading);

## ■ Expression Language

- скриптовый язык выражений, который позволяет получить доступ к Java компонентам из JSP;
- является расширением JSP 2.1 спецификации;

# Spring Framework :: Доступ к данным

## ■ JDBC

- обертка для упрощения работы с JDBC;
- единообразная обработка исключений для всех СУБД;
- поддержка DAO (Data Access Objects);

## ■ ORM

- интеграция с популярными ORM API – Hibernate, JPA, JDO, iBatis и т.д.;
- возможность использовать весь функционал соответствующих ORM совместно с функционалом Spring (например, декларативным управлением транзакциями);

## ■ Transactions

- обеспечивает поддержку программного и декларативного управления транзакциями;
- поддержка управления транзакциями для POJO;

■ OXM – универсальная поддержка различных имплементаций Object/XML отображений (JAXB, XMLBeans, Xstream и т.д.)

■ JMS – поддержка Java Messaging Service

# Spring Framework :: AOP

- AOP
  - Реализация Aspect-Oriented Programming;
  - Crosscutting (секущее, поперечное) свойство для нескольких классов, модулей;
  - дополняет OOP, обогащая его другим типом модульности, который позволяет локализовать код реализации crosscutting логики в одном модуле;
  - применим к POJO;
  - соответствует AOP Alliance;
- Aspects – интеграция с AspectJ;
- Instrumentation – различный инструментарий и реализации загрузчиков классов (classloaders) для специфичных серверов приложений
- Примеры:
  - журналирование приложений;
  - контекстно-зависимая обработка ошибок;
  - оптимизации выполнения программ;



# Spring Framework :: Web

- Web Core
  - загрузка файлов из нескольких частей (Спецификация Servlet 3.0 внесет коррективы в ближайшее время);
  - инициализация IoC контейнера с использованием servlet listeners;
  - реализация веб-ориентированного application context;
- Web-Servlet
  - реализация Model-View-Controller для веб-приложения
- Web-Portlet
  - Реализация MVC для использования в portlet-окружении
  - повторяет функциональность Web Servlet модуля
- Web-Struts
  - поддержка интеграции с веб-слоем классического Struts
  - не рекомендуется к использованию, начиная со Spring 3.0
  - Рекомендуется миграция на Struts 2.0 и использование встроенной в него интеграции со Spring или переход на использование Spring MVC

# Spring Framework :: Test support

- Поддержка тестирования компонентов Spring с помощью Junit или TestNG;
- Обеспечивает корректную загрузку Application Context при тестировании;
- Предоставляет Mock объекты для изолированного тестирования кода;

# Spring Framework :: Структура фреймворка

- Spring в Managed - среде:
  - Декларативное управление безопасностью;
  - Декларативные транзакции;
  - Контроль соединений с базой данных;
  - Примеры: JBoss, BEA WebLogic, IBM WebSphere;
  
- Spring в Non-managed среде:
  - Приложение само должно отслеживать все ресурсы;
  - Примеры: Jetty, Tomcat, J2SE приложение;
  
- Spring позволяет работать в non-managed среде как в managed, без внедрения тяжеловесных решений (в том числе и с использованием JTA);

# Упражнения

- №1: Установка и настройка рабочей среды:
  - 15 мин – при использовании виртуальной машины;
  - 45 мин – при самостоятельной настройке среды окружения;
- №2: Установка и настройка вспомогательных средств:
  - 30 мин

# Вопросы!?

