

ИЗУЧЕНИЕ ПЛАТФОРМЫ JAVA ENTERPRISE EDITION В КОНТЕКСТЕ ЕЕ МНОГООБРАЗИЯ

Бабков В.С., к.т.н., доц. каф. ПМИ,
куратор академических программ
ORACLE Academy и CISCO Network Academy
в Донецком национальном техническом университете

Платформа Java EE (Enterprise Edition) с момента своего появления в 1999 завоевала огромную популярность, прежде всего, в корпоративной среде, среде веб-приложений и других отраслях, которые предъявляют высокие требования с точки зрения масштабируемости, распределенности, гибкости и надежности в сочетании с высокой степенью повторного использования кода [1].

В настоящее время (последняя версия Java EE 6 – 2009 г.) платформа в «чистом виде» представляет собой множество спецификаций и стандартов и не позволяет эффективно решать задачи без множества высокоуровневых надстроек и реализаций. Данные надстройки постоянно повышают уровень абстрагированности от деталей и дают возможность сконцентрироваться исключительно на проблемах решаемой задачи, а не того, как эта проблем будет решаться средствами платформы. Поэтому платформа Java EE как снежный ком обрастает множеством вспомогательных технологий, каркасов и библиотек для решения разнообразнейших задач. Фактически современная Java EE – это целая экосистема, которая включает как средства разработки, сборки, развертывания так и множество библиотек самого разнообразного назначения [2].

В процессе изучения данной платформы, как в рамках учебных курсов, так и в рамках специализированных тренингов возникает проблема, каким технологиям обучать в виду их постоянной изменчивости, появления новых стандартов и т.д. Фактически возникает вопрос: «Что есть тренд?», - для Java EE платформы.

В конце 2010 г ресурсом <http://www.zereturnaround.com/> проводился анализ, результаты которого могут прояснить тенденцию популярности различных компонентов платформы на 2011 год. Условно компоненты были разделены на такие категории:

- средства автоматизации сборки приложений;
- интегрированные среды разработки;
- средства развертывания приложений;
- стандарты;
- каркасы.

Все данные взяты из источника <http://www.zereturnaround.com/> [3].

Рис. 1 показывает, что в категории «Средства автоматизации сборки приложений» Maven и Ant занимают примерно равные доли по популярности. При этом, по опыту автора, можно сказать, что изучение Maven проходит значительно сложнее, чем Ant, однако в дальнейшем при усложнении проектов гибкость Maven компенсирует сложность его изучения. Однажды созданная для проекта структура Maven-репозитория и его конфигурация с лихвой перекрываются удобством пересборки, автоматического тестирования приложений в разнообразных конфигурациях и с учетом зависимостей от большого числа внешних компонентов.

Популярность среды разработки Eclipse (рис. 2) легко объясняется тем фактом, что именно данная IDE обладает наиболее широким набором модулей расширения (плагинов) для облегчения задачи конструирования Enterprise-приложений (плагины для GWT, Spring и т.д.) [4].

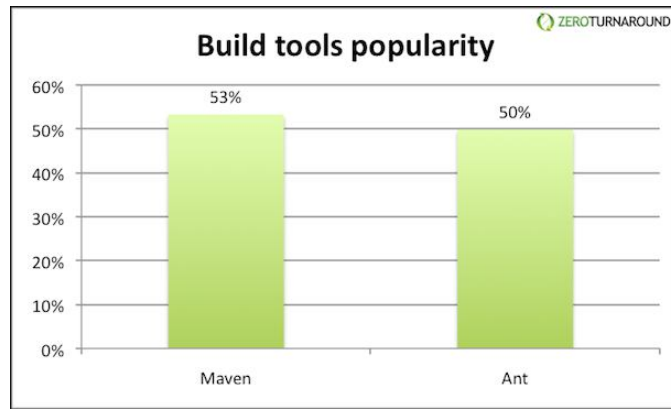


Рис. 1 – Сравнение популярности средств автоматизации сборки приложений на платформе Java EE [3]

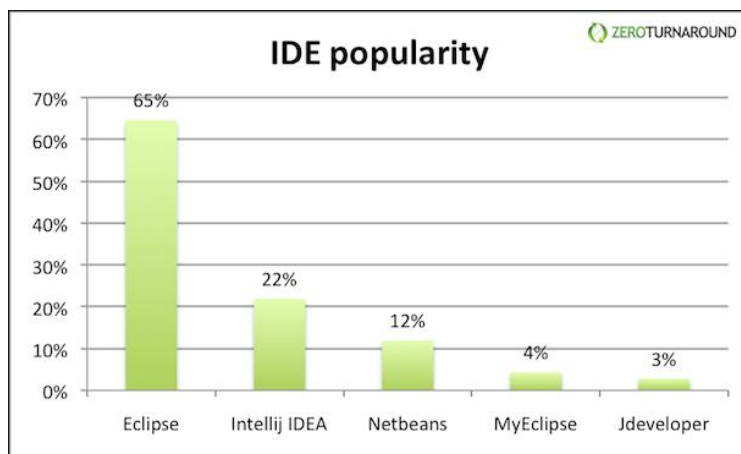


Рис. 2 – Сравнение популярности сред разработки на платформе Java EE [3]

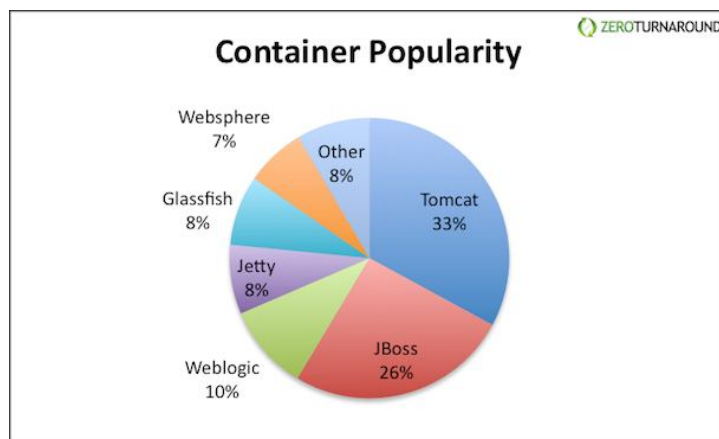


Рис. 3 – Сравнение популярности средств развертывания приложений на платформе Java EE [3]

На рис. 3 показано, что наиболее популярными средствами развертывания являются Tomcat (как контейнер сервлетов) и JBoss (как полноценный сервер приложений). По опыту преподавания Enterprise-платформы можно утверждать, что большинство сред разработки легко интегрируются с любыми серверами приложений, поэтому желательно, в общих чертах ознакомиться как минимум с таким набором (Tomcat, JBoss, Glassfish, Jetty) – как с представителями разных весовых категорий и функциональности.

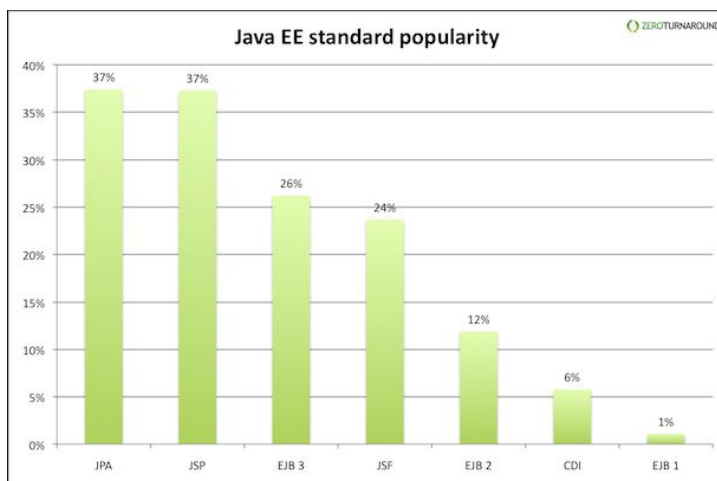


Рис. 4 – Сравнение популярности стандартов платформы Java EE [3]

Рис. 4 показывает необходимость сконцентрироваться на 4 базовых стандартах, которые уже много лет не теряют популярности, совершенствуясь по мере выхода новых версий: JPA (доступ к БД), EJB (инкапсуляция логики в серверных компонентах), JSP (создание веб-страниц на Java) и JSF (интерфейс пользователя).

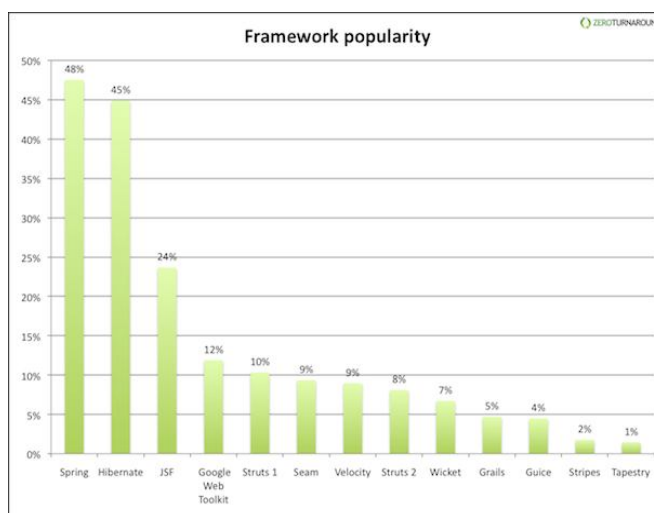


Рис. 5 – Сравнение популярности каркасов для платформы Java EE [3]

Аналогично рис. 5 позволяет выбрать в качестве ключевых компонентов для изучения платформы Java EE такие каркасы, как Spring (общая концепция и контейнер), Hibernate (доступ к данным), JSF, GWT, Struts (реализация клиентского интерфейса).

Проведенный анализ позволяет рекомендовать строить схему обучения технологиям платформы Java EE таким образом, чтобы максимально захватить вышеперечисленные трендовые компоненты. При этом крайне полезным является комплексный подход, при

котором контраст между использованием разных технологий и средств разработки демонстрируется на примере разработке достаточно сложного многофункционального приложения по схеме с нарастающей сложностью. На тренингах, проводимых автором, типичным примером такого приложения является Enterprise-система для хранения и обработки пользовательских изображений. Основные функции: клиентский интерфейс для загрузки и манипуляции изображениями, многопользовательский режим с разграничением доступа, наличие встроенных средств администрирования, хранение данных в БД, масштабируемая система на основе серверных компонентов (EJB) и т.д. В этом случае, зарекомендовавшая себя на практике с положительной стороны схема развития приложения выглядит так:

- реализация логики в коде сервлетов, интерфейса пользователя в виде JSP-страниц;
- перенос логики в серверные компоненты EJB, замена Java-кода в страницах на JSTL;
- размещение информации в БД и доступ средствами JPA и Hibernate;
- построение приложения на каркасе Struts;
- построение приложения на каркасе Spring;
- реализация интерфейса пользователя средствами JSF;
- реализация интерфейса пользователя средствами GWT.

При этом рассматриваются два варианта сборки приложения: с помощью Ant (на первых этапах) и с помощью Maven (на всех последующих).

Выводы

Изучение платформы Java EE на основе создания комплексного Enterprise-приложения позволяет максимально охватить популярные компоненты платформы, продемонстрировав при этом контраст при использовании альтернативных каркасов, средств развертывания и т.п. Важным фактором является непрерывный мониторинг состояния платформы и отслеживание трендов для того, чтобы рассматриваемый материал находился в постоянном соответствии с потребностями и предпочтениями IT-рынка.

В любом случае актуальность изучения платформы Java EE несомненна, т.к. по компетентному рейтингу, который уже много лет составляется ежемесячно компанией Tiobe (www.tiobe.com), язык Java уверенно входит в тройку лидеров (а по состоянию на март 2011 г. занимает первое место) по востребованности на рынке наряду с C++, C#, Python.

Литература

1. Java EE at a Glance [Электронный ресурс]. Режим доступа: <http://www.oracle.com/technetwork/java/javaee/overview/index.html>
2. Ian Evans, Your First Cup: An Introduction to the Java EE Platform, - Vervante publ. - 2009 – P. 52
3. Java EE Productivity Report 2011 [Электронный ресурс]. Режим доступа: <http://www.zereturnaround.com/java-ee-productivity-report-2011/>
4. Каталог расширений для IDE Eclipse [Электронный ресурс]. Режим доступа: <http://marketplace.eclipse>.