

Гадание на кофейной гуще

Андрей Ковалев

Пусть читатель простит мне непоследовательность, но меня всегда увлекала астрология, и сегодня, в полнолуние (хотя сетка легла, а в Москве туман и я не могу точно сказать, что сегодня именно полнолуние) хочется погадать. Чтобы придерживаться темы нашей колонки, гадание следует проводить на кофейной гуще. О чем гадать? Естественно, о механизме работы в гуще кофейных зерен. Ведь по моему скромному мнению `JavaBeans` можно наиболее точно перевести как "кофейные зерна".

Некоторое время я переводил этот термин дословно наиболее распространенным русским эквивалентом "бобы", коротко и понятно. Однако, поработав с ними некоторое время, я стал понимать, что бобами называют скорее плоды какао, а `Java`скими, то есть кофейными, называются именно "*зерна*", хотя это и не так сильно звучит, как "*бобы*". По привычке буду называть эти "компоненты многообразного использования" бобами, радуясь благозвучности слова и опуская различия между кофе и какао, что для программирования особого значения не имеет.

Желание затронуть эту тему возникло у меня не только по тому, что задача их освоения стоит передо мной непосредственно, но также и потому, что эта технология венчает все достижения самой технологии `Java`. Именно такое понимание можно получить, читая спецификацию и многочисленные обсуждения технологии "`JavaBeans`". Создается впечатление, что стоит написать еще десяток — другой бобов, и наступит всеобщее благоденствие, программирование сведется к одной лишь бизнес-логике, а пользовательский интерфейс будет строиться из этих самых бобов, как дом из кирпичей.

Где же они, эти кирпичи? Пока они существуют в виде разного рода обучающих примеров. `Beanbox`, `BML` и прочие библиотеки представляют собой такую смесь красивых, но практически ненужных компонентов, что возникает резонный вопрос,

зачем все это нужно!?

По существу, технология JavaBeans" является наиболее сложной из технологий, относящихся к разработке пользовательских интерфейсов. Об этом свидетельствует и [история развития технологий](#). Спецификация выпущена в спешном порядке 16 октября 1996 года, 5 августа 1997 зарегистрировано стотысячная загрузка BDK. После этого в истории бобы не упоминаются, а появляются в виде новой клиент-серверной технологии EJB, которая в свою очередь принимает очертания J2EE в декабре 1999 года и начинается ее активное лицензирование. Бизнес есть бизнес, ничто не стоит на месте.

Удивительный пакет BDK предлагает практическое изучение бобов не единственном примере beanbox, документация привлекает своей лаконичностью и в то же время заставляет гадать о применимости технологии в силу неоднозначности. Возникает чувство, что все сосредоточено вокруг свойств бобов.

Прежде чем углубиться в технологию, рассмотрим основные отличия JavaBeans от обычных компонентов.

1. прежде всего, боб надо построить, то есть использовать средство его разработки и описать его там, по-простому же надо просто создать этот маленький кирпич пользовательского интерфейса;
2. во-вторых, применяя бобы, разработчик прежде всего уделяет внимание его свойствам, с помощью которых можно сделать его многократно используемым или оформить в виде шаблона, на основе которого можно будет строить другие бобы;
3. далее, можно разработать средства настройки свойств боба во время его работы (run-time);
4. наконец, желательно иметь возможность сохранять заданные бобу свойства, чтобы не начинать процедуру настройки каждый раз сначала.

В остальном бобы ничем не отличаются от обычных компонентов, также выглядят внешне, хотя могут состоять из нескольких компонентов, также обрабатывают события, хотя механизм обработки имеет свою специфику, связанную с обновлением свойств, компилируются в те же наборы классов, хотя могут храниться и распространяться в виде отдельных jar-файлов.

вот пришел админ, сеть заработала, а продолжение следует...