

Содержание

Об авторе	5
О рецензентах	6
Предисловие	13
Вопросы, освещаемые в книге	13
Что нужно для чтения этой книги	14
Для кого эта книга	15
Соглашения	15
Поддержка клиентов	16
Сообщения об ошибках	16
Незаконное воспроизведение содержимого книги	16
Вопросы	16
1. Знакомство с NetBeans	17
Введение	17
Получение NetBeans	18
Установка NetBeans	21
Microsoft Windows	21
Mac OSx	21
Linux и Solaris	22
Другие платформы	22
Процедура установки	22
Первый запуск NetBeans	27
Конфигурирование NetBeans для разработки	
Java EE-приложений	28
Интегрирование NetBeans со сторонним сервером приложений ...	28
Интегрирование NetBeans с СУРБД стороннего производителя ...	30

Развертывание нашего первого приложения	34
Подсказки NetBeans для эффективной разработки	37
Автозавершение кода	37
Шаблоны кода	40
Клавиши быстрого вызова	41
Изучение визуальных индикаторов NetBeans	44
Резюме	46
2. Разработка веб-приложений, содержащих Сервлеты и JSP	47
Создание нашего первого веб-приложения	47
Изменение кода, сгенерированного NetBeans	52
Разработка сервлета	67
Добавление сервлета к нашему приложению	67
Безопасность веб-приложений	75
Реализация аутентификации на основе формы	77
Конфигурирование безопасности, специфичной для GlassFish	83
Фрагменты JSP	87
Создание фрагмента JSP в NetBeans	87
Резюме	90
3. Улучшение функциональности JSP с помощью JSTL и пользовательских тегов	91
Базовые теги JSTL	92
Условное отображение части страницы тегом <c:if>	92
Отображение взаимоисключающей разметки с помощью тега <c:choose>	95
Итерация по массивам или коллекциям с помощью тега <c:forEach>	98
SQL-теги JSTL	101
Извлечение данных из базы данных с помощью тега <sql:query>	103

Изменение данных в базе данных с помощью тега <sql:update>	107
Вставка данных в базу данных	107
Обновление данных базы данных	110
Удаление данных из базы данных	112
Заключительные замечания о JSTL	115
Пользовательские теги JSP	115
Резюме	121
4. Разработка веб-приложений с использованием JavaServer Faces 2.0	122
Введение в JavaServer Faces	122
Разработка нашего первого приложения JSF	123
Создание нового проекта JSF	123
Шаблоны фэйслетов	142
Добавление шаблона фэйслета к нашему проекту	144
Использование шаблона	144
Составные компоненты	148
Резюме	152
5. Создание изящных веб-приложений с помощью PrimeFaces	153
Наш первый проект PrimeFaces	153
Использование компонентов PrimeFaces в наших приложениях JSF	156
Представления, снабженные вкладками	160
Интерфейсы мастера	165
Дополнительная информация	170
Резюме	170

6. Взаимодействие с базами данных через API Персистентности Java	171
Создание нашей первой JPA-сущности	172
Добавление к сущности сохраняемых полей	179
Создание DAO	180
Автоматизированная генерация JPA-сущностей	186
Именованные запросы и JPQL	192
Проверка допустимости со стороны бинов	193
Отношения сущностей	194
Генерация JSF-приложений из сущностей JPA	199
Резюме	205
7. Реализация уровня бизнес-логики на сеансовых бинах	206
Введение в Сеансовые бины	206
Создание сеансового бина в NetBeans	207
Доступ к бину из клиента	216
Выполнение клиента	219
Транзакции, управляемые сеансовым бином	219
Реализация аспектно-ориентированного программирования с помощью перехватчиков	221
Реализация класса перехватчика	222
Декорирование EJB аннотацией @Interceptors	223
Служба таймера EJB	224
Генерация сеансовых бинов из JPA-сущностей	225
Резюме	230
8. Контексты и Инжекция зависимости	231
Введение в CDI	231
Квалификаторы	238
Стереотипы	242
Типы привязки перехватчика	245
Резюме	248

9. Обмен сообщениями в JMS и Управляемых сообщением бинах	249
Введение в JMS	249
Создание проекта и ресурсов JMS	251
Создание пункта назначения JMS	252
Отправка сообщений пункту назначения	255
Обработка сообщений JMS с помощью управляемых сообщением бинов	259
Резюме	263
10. Веб-сервисы SOAP в JAX-WS	264
Введение в веб-сервисы	264
Создание простого веб-сервиса	265
Тестирование нашего веб-сервиса	270
Разработка клиента для нашего веб-сервиса	272
Представление EJB в качестве веб-сервисов	277
Реализация новых веб-сервисов в виде EJB	277
Представление существующего EJB в виде веб-сервисов	280
Создание веб-сервиса из существующего WSDL	282
Резюме	284
11. RESTful веб-сервисы в JAX-RS	285
Создание RESTful веб-сервиса из существующей базы данных	286
Анализ сгенерированного кода	288
Тестирование нашего RESTful веб-сервиса	292
Разработка клиента RESTful веб-сервиса	296
Резюме	301
Приложение А. Отладка корпоративного приложения отладчиком NetBeans	302
Отладка корпоративных приложений	302
Резюме	308

Приложение Б. Выявление проблем с производительностью при помощи профилировщика NetBeans	309
Профилрование нашего приложения	309
Резюме	313
Алфавитный указатель	314

Предисловие

Java EE 6 является последней версией спецификации Java EE, в которую добавлено несколько новых возможностей для упрощения разработки корпоративных приложений. В эту последнюю версию Java EE были включены новые версии существующих API Java EE. Так, например, JSF 2.0 значительно упрощает разработку веб-приложений, а в JPA 2.0 появилась возможность использования нового API Критериев, кроме этого, имеется несколько других улучшений. Были улучшены Селансовые бины EJB для поддержки асинхронных вызовов метода, а также произведено несколько других серьезных улучшений. Сервлет 3.0 добавляет несколько новых возможностей, таких как вызовы дополнительного метода и необязательность создания дескриптора развертывания `web.xml`. Дополнительно к Java EE было добавлено несколько новых API, включая JAX-RS, который упрощает разработку RESTful веб-сервисов, и CDI, помогающий интегрировать различные уровни типичного корпоративного приложения.

NetBeans также претерпел обновление для поддержки полной функциональности, предоставляемой Java EE 6, чтобы еще больше упростить и ускорить разработку Java EE 6-совместимых приложений. В этой книге мы исследуем все возможности NetBeans, которые предназначены для разработки корпоративных приложений Java EE 6.

Вопросы, освещаемые в книге

Глава 1 «*Знакомство с NetBeans*» на стр. 17 представляет введение в NetBeans, а также знакомит нас с подсказками, экономящими время, и приемами, которые позволяют более эффективно разрабатывать приложения Java.

Глава 2 «*Разработка веб-приложений содержащих Сервлеты и JSP*» на стр. 47 демонстрирует, как с помощью NetBeans упрощается разработка веб-приложений путем использования API Сервлета и JavaServer Pages.

Глава 3 «*Улучшение функциональности JSP с помощью JSTL и пользовательских тегов*» на стр. 91 показывает, как с помощью NetBeans можно создавать удобные в сопровождении веб-приложения, используя преимущества Стандартной библиотеки тегов JSP (JSTL), а также поясняет, как написать наши собственные теги JSP.

Глава 4 «*Разработка веб-приложений с использованием JavaServer Faces 2.0*» на стр. 122 объясняет, как с помощью NetBeans можно существенно облегчить раз-

работку веб-приложений, использующих возможности стандартного каркаса компонентов JavaServer Faces 2.0.

Глава 5 «Создание изящных веб-приложений с помощью PrimeFaces» на стр. 153 посвящена вопросу разработки изящных полнофункциональных веб-приложений Ajax путем использования библиотеки компонентов PrimeFaces JSF, поставляемой с NetBeans.

Глава 6 «Взаимодействие с базами данных через API Персистентности Java» на стр. 171 объясняет, как с помощью NetBeans упрощается разработка приложений, использующих возможности API Персистентности Java (JPA), включая автоматическую генерацию сущностей JPA из существующих схем баз данных. В этой главе также объясняется, как можно сгенерировать завершенные веб-приложения из существующей схемы базы данных всего несколькими щелчками мыши.

Глава 7 «Реализация уровня бизнес-логики на сеансовых бинах» на стр. 206 наглядно демонстрирует, насколько NetBeans упрощает разработку сеансовых бинов EJB 3.1.

Глава 8 «Контексты и Инжекция зависимости» на стр. 231 показывает, как с помощью нового API CDI, введенного в Java EE 6, можно упростить интегрирование различных уровней нашего корпоративного приложения.

Глава 9 «Обмен сообщениями в JMS и Управляемых сообщением бинах» на стр. 249 посвящена технологиям обмена сообщениями Java EE, таким как Служба обмена сообщениями Java (JMS) и Управляемые сообщением бины (MDB), демонстрируя функциональность NetBeans, которая упрощает разработку приложений, использующих возможности этих API.

Глава 10 «Веб-сервисы SOAP в JAX-WS» на стр. 264 объясняет, как с помощью NetBeans можно облегчить разработку веб-сервисов SOAP на основе API Java для веб-сервисов XML (JAX-WS).

Глава 11 «RESTful веб-сервисы в JAX-RS» на стр. 285 рассматривает JAX-RS, новое дополнение к спецификации Java EE, которое упрощает разработку RESTful веб-сервисов.

«Приложение А. Отладка корпоративного приложения отладчиком NetBeans» на стр. 340 представляет введение в отладчик NetBeans и то, как его можно использовать для обнаружения дефектов нашего приложения.

«Приложение Б. Выявление проблем с производительностью при помощи профилировщика NetBeans» на стр. 348 представляет профилировщик NetBeans, объясняя возможности его использования для анализа проблем производительности наших приложений.

Что нужно для чтения этой книги

Для чтения этой книги нужно установить Комплект разработчика Java (Java Development Kit (JDK)) 1.6 или более новую версию и IDE NetBeans 7.0 в редакции Java EE.

Для кого эта книга

Эта книга адресована трем различным категориям разработчиков:

- Java-разработчикам (не обязательно знакомым с NetBeans), желающим стать специалистами в Java EE 6, которые хотят использовать NetBeans для разработки Java EE-приложений;
- пользователям NetBeans, желающим познакомиться с тем, как использовать свой IDE для разработки Java EE 6-приложений;
- опытным Java EE 6-разработчикам, желающим узнать, как с помощью NetBeans можно облегчить разработку их Java EE 6-приложений.

Соглашения

В этой книге вы обнаружите несколько стилей текста, которые разделяют различные виды информации. Вот некоторые примеры этих стилей и пояснение их значений.

Элементы кода в тексте показаны следующим образом: «XML-теги `<servlet>` и `<servlet-mapping>` используются для фактического конфигурирования нашего сервлета».

Блок кода отображается следующим образом:

```
<servlet-mapping>
  <servlet>SimpleServlet</servlet>
  <url-pattern>*.foo</url-pattern>
</servlet-mapping>
```

Когда мы хотим привлечь ваше внимание к определенной части блока кода, соответствующие строки или элементы будут выделены жирным шрифтом:

```
<b>Меню приложения</b>
<ul>
  <li/> <a href="main.jsp">Main</a>
  <li/> <a href="secondary.jsp">Secondary</a>
</ul>
Текущая страница: <%= pageName %>
```

Любой ввод или вывод командной строки записывается следующим образом:

```
javac -cp /opt/sges-v3/glassfish/lib/javaee.jar
net/ensode/glassfishbook/simpleapp/SimpleServlet.java
```

Важные (ключевые) слова в тексте выделяются курсивом.

Слова, которые вы видите на экране, в меню или диалоговых окнах, например, появляются в тексте наподобие этого: «Здесь мы должны щелкнуть по элементу **Развертывание приложения** в разделе **Развертывание** основного экрана».

Для обозначения последовательно выполняемых действий используется символ | (вертикальная черта): «Щелкните по узлам **Ресурсы** (Resources) | **JDBC** | **Пулы соединений** (Connection Pools) в панели навигации».

Веб-адреса выделяются подчеркиванием и специальным шрифтом, например: «Мы сможем увидеть отображение файла `dataentry.html`, вводя в адресной строке обозревателя <http://localhost:8080/formhandling>».



Предупреждения или важные примечания появляются в тексте таким образом.



Советы и рекомендации появляются наподобие этого.



Символ в правом нижнем углу нечетной страницы указывает, что листинг кода имеет продолжение на следующем развороте страниц.

Поддержка клиентов

Теперь, когда вы являетесь счастливым обладателем книги, у нас имеется возможность помочь вам извлечь максимум пользы из вашего приобретения.



Загрузите примеры кода для этой книги

Файлы с примерами кода для данной книги можно загрузить с сайта издательства:

<http://www.dmk-press.ru/>.

Сообщения об ошибках

Хотя мы делаем все возможное для того, чтобы не допустить ошибок в наших изданиях, вы можете встретить в тексте какие-либо неточности. В таком случае мы будем рады, если вы сообщите нам об этом.

Сообщения об ошибках в русскоязычном издании этой книги можно оставить на сайте издательства «ДМК Пресс»: <http://www.dmk-press.ru/contacts1/contacts/>.

Незаконное воспроизведение содержимого книги

Пиратские копии – повсеместная проблема. Если вам встретились незаконным образом растиражированные экземпляры данной книги (в любом формате), пожалуйста, сообщите источник публикации, написав письмо по адресу dm@dmk-press.ru или copyright@packtpub.com.

Вопросы

Вы можете присылать любые вопросы, касающиеся данной книги, по адресу dm@dmk-press.ru или questions@packtpub.com. Мы постараемся разрешить возникшие проблемы.

Знакомство с NetBeans

В этой главе мы рассмотрим, как приступить к работе с NetBeans. Будут затронуты следующие темы:

- введение;
- получение NetBeans;
- установка NetBeans;
- первый запуск NetBeans;
- конфигурирование NetBeans для разработки Java EE-приложений;
- развертывание нашего первого приложения;
- подсказки NetBeans, повышающие эффективность разработки.

Введение

NetBeans является *Интегрированной средой разработки* (Integrated Development Environment (IDE)) и, в дополнение к этому, платформой. Хотя первоначально IDE NetBeans мог использоваться только для разработки приложений Java, начиная с версии 6 NetBeans поддерживает несколько языков программирования. Это либо встроенная поддержка, либо поддержка, осуществляемая путем установки дополнительных плагинов. NetBeans изначально поддерживает следующие языки программирования: Java, JavaFX, C, C++ и PHP. Groovy, Scala, Ruby и другие языки поддерживаются через дополнительные плагины.

В дополнение к тому что NetBeans является IDE, он также является платформой. Разработчики могут использовать API NetBeans как для создания плагинов NetBeans, так и для создания автономных приложений.



С краткой историей NetBeans можно познакомиться по адресу: <http://NetBeans.org/about/history.html>.

Хотя IDE NetBeans поддерживает несколько языков программирования, всё-таки его базовым языком является Java, поэтому эта IDE наиболее удобна для разработки на языке Java. Как Java IDE, NetBeans имеет встроенную поддержку приложений Java SE (Standard Edition), которые обычно работают на настольных компьютерах или

ноутбуках; приложений Java ME (Micro Edition), которые обычно работают на портативных устройствах, таких как сотовые телефоны или PDA; и приложений Java EE (Enterprise Edition), которые обычно работают на серверах «большого железа» и могут поддерживать тысячи параллельных пользователей.

В этой книге мы сосредоточимся на изучении возможностей NetBeans, используемых при разработке Java EE-приложений, а также на том, как максимально полно использовать возможности NetBeans, позволяющие разрабатывать приложения Java EE более эффективно.

Некоторые из функций NetBeans, которые мы рассмотрим, позволяют существенно ускорить разработку веб-приложений путем использования JSF, API Сервлета, а также страниц JSP, предоставляя отправные точки для артефактов такого рода. Также мы сможем задействовать палитру NetBeans для копирования и вставки фрагментов кода – методом перетаскивания¹ в наши страницы JSP, включая HTML– и JSP–разметку. Помимо этого будет рассмотрено, как с помощью NetBeans автоматизировать генерацию JPA-сущностей² из существующей схемы базы данных.

В дополнение к веб-разработке еще будет рассмотрено, как с помощью NetBeans упрощается разработка Enterprise JavaBeans (EJB) и будет рассмотрена возможность упрощения разработки веб-сервисов. Мы также рассмотрим, как можно запросто написать и EJB, и клиентов веб-сервиса, воспользовавшись возможностями некоторых превосходных функций NetBeans.

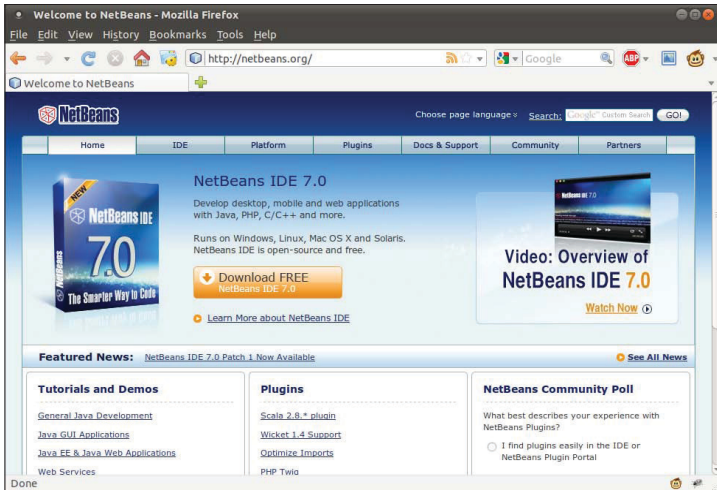
Перед тем как воспользоваться преимуществами всех вышеупомянутых функций NetBeans, нам, конечно, нужно будет установить NetBeans, как это описано в следующем разделе.

Получение NetBeans

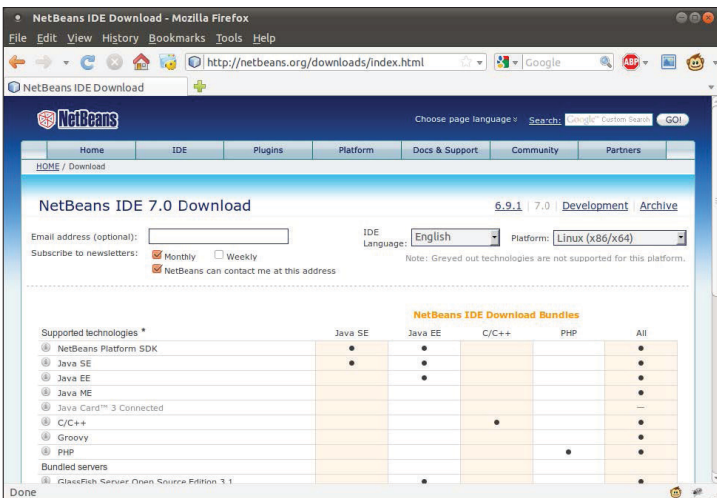
NetBeans можно загрузить по адресу: <http://www.netbeans.org>.

¹ Англоязычное наименование метода – «drag and drop». – *Прим. перев.*

² Аббревиатура JPA означает Java Persistence API (API Персистентности Java), стандартный инструмент объектно-реляционного отображения, включенный в Java EE. – *Прим. перев.*



Для загрузки NetBeans нам нужно щелкнуть по кнопке (см. рисунок выше) **Загрузить бесплатно IDE NetBeans 7.0** (Download Free NetBeans IDE 7.0). Точное имя кнопки может быть другим; оно зависит от текущей версии NetBeans. Щелчок по этой кнопке переадресует нас на страницу, на которой перечислены все доступные комплекты поставки дистрибутива NetBeans.



Различные дистрибутивы NetBeans содержат различные комплекты поставки NetBeans, которые, в свою очередь, предоставляют разные уровни функциональных возможностей. В следующей таблице перечислены различные доступные комплекты поставки NetBeans и описана функциональность, которую они предоставляют:

Комплект поставки NetBeans	Описание
Java SE	Позволяет разрабатывать приложения Java для настольных компьютеров
Java EE	Позволяет разрабатывать приложения Java SE (обычные приложения для настольных компьютеров) и Java Enterprise Edition (корпоративные приложения, работающие на «большом железе»)
C/C++	Позволяет разрабатывать приложения, написанные на языках C или C++
PHP	Позволяет разрабатывать веб-приложения с использованием популярного языка PHP с открытым исходным кодом
All	Включает в себя функциональность всех комплектов поставки NetBeans

Для изучения примеров, приведенных в этой книге, необходим комплект *Java EE* или комплект *All*.



Все снимки экрана в этой книге были сделаны для случая использования комплекта *Java EE*. NetBeans может выглядеть немного по-другому, если будет использован комплект *All*, особенные отличия в этом случае можно будет заметить в виде некоторых дополнительных пунктов меню.

Официально поддерживаются следующие платформы:

- Windows 7/Vista/XP/2000;
- Linux x86;
- Linux x64;
- Solaris x86;
- Solaris x64;
- Mac OSx.

Дополнительно NetBeans может выполняться на любой платформе, содержащей Java 6 или ее более новую версию. Также доступна для загрузки версия NetBeans, не зависящая от операционной системы, которая будет выполняться на любой из этих платформ.



Хотя не зависящая от операционной системы версия NetBeans может быть выполнена на всех поддерживаемых платформах, рекомендуется получить версию NetBeans, которая специфична для вашей конкретной платформы.

Страница загрузки NetBeans сама определит используемую нами операционную систему для получения доступа к соответствующему дистрибутиву, а используемая платформа будет выбрана по умолчанию. Если дело обстоит иначе или если мы загружаем NetBeans, который собираемся установить на другой рабочей станции,

использующей иную платформу, требуемую платформу можно выбрать из выпадающего списка (см. рис. на стр. 19), называемого **Платформа** (Platform).

После того как будет выбрана корректная платформа, мы должны щелкнуть по кнопке **Загрузить** (Download), соответствующей комплекту NetBeans, который мы хотим установить. Для разработки Java EE-приложений нам нужен комплект *Java EE* или комплект *All*. После этого дистрибутив NetBeans будет загружен в указанный нами каталог.



Для использования приложения Java EE его нужно развернуть на сервере приложений. На рынке существует несколько серверов приложений, между тем и комплект *Java EE*, и комплект *All* NetBeans содержат в себе GlassFish и Tomcat. Tomcat является популярным контейнером сервлета с открытым исходным кодом, он может использоваться для развертывания приложений, использующих Сервлеты, JSP и JSF, однако он не поддерживает других технологий Java EE, таких как EJB или JPA. GlassFish является 100-процентным Java EE совместимым сервером приложений. Мы будем использовать поставляемый в комплекте сервер приложений GlassFish для развертывания и выполнения наших примеров.

Установка NetBeans

NetBeans требует, чтобы был доступен *Комплект разработчика Java* (Java Development Kit (JDK)) версии 6.0 или более новой, прежде чем мы сможем его установить.



Поскольку эта книга адресована опытным разработчикам Java, мы не будем тратить много времени на объяснения того, как установить и сконфигурировать JDK, так как мы можем обоснованно предположить, что каждый из читателей целевого рынка, которому адресована эта книга, безусловно устанавливал JDK. Инструкции по установке JDK 6 можно найти по адресу: <http://www.oracle.com/technetwork/Java/Javase/index-137561.html>. Читатели, желающие использовать Mac OSx, могут получить инструкции по установке и загрузке JDK для их платформы по адресу https://developer.apple.com/library/mac/#documentation/Java/Conceptual/Javal4Development/02-JavaDevTools/JavaDevTools.html#//apple_ref/doc/uid/TP40001884-SW1.

Установка NetBeans немного отличается в зависимости от поддерживаемой платформы. В следующих нескольких разделах мы объясним, как установить NetBeans на каждой поддерживаемой платформе.

Microsoft Windows

Для платформ Microsoft Windows NetBeans загружается в виде исполняемого файла с названием, подобным **netbeans-7.0-ml-java-windows.exe** (точное имя зависит от версии NetBeans и комплекта NetBeans, который был выбран для загрузки). Чтобы установить NetBeans на платформах Windows, просто перейдите к папке, в которую был загружен NetBeans, и дважды щелкните по исполняемому файлу.

Mac OSx

Для Mac OSx загруженный файл называется наподобие **netbeans-7.0-ml-java-macosx.dmg** (точное имя зависит от версии NetBeans и комплекта NetBeans, который был выбран для загрузки). Для установки NetBeans перейдите к расположению, в которое был загружен файл, и дважды щелкните по файлу.

Установщик Mac OSx содержит четыре комплекта: NetBeans, GlassFish, Tomcat и OpenESB, – эти четыре пакета должны быть установлены индивидуально. Они могут быть установлены простым двойным щелчком по каждому из них. Пожалуйста, обратите внимание, что GlassFish должен быть установлен до установки OpenESB.

Linux и Solaris

Для Linux и Solaris NetBeans загружается в форме shell-сценария. Имя файла будет похоже на **netbeans-7.0-ml-java-linux.sh**, **netbeans-7.0-ml-java-solaris-x86.sh** или **netbeans-7.0-ml-java-solaris-sparc.sh**, в зависимости от версии NetBeans, выбранной платформы и выбранного комплекта NetBeans.

Прежде чем NetBeans можно будет установить на этих платформах, загруженный файл должен быть сделан исполняемым файлом. Это можно выполнить с помощью командной строки, перейдя к каталогу, в который был загружен установщик NetBeans, и выполнив следующую команду:

```
chmod +x./filename.sh
```

Нужно заменить **filename.sh** именем нашего файла, соответствующим платформе и комплекту NetBeans, который мы устанавливаем.

После того как файл станет исполняемым файлом, его можно будет установить из командной строки:

```
./filename.sh
```

И вновь нужно заменить **filename.sh** именем нашего файла, соответствующим платформе и комплекту NetBeans, который мы устанавливаем.

Другие платформы

Для других платформ NetBeans может быть загружен в виде независимого от платформы zip-файла. Имя zip-файла будет чем-то вроде этого: **netbeans-7.0-201007282301-ml-java.zip** (точное имя файла может измениться в зависимости от конкретной версии NetBeans и комплекта NetBeans, который был выбран для загрузки).

Чтобы установить NetBeans на одной из этих платформ, нужно просто извлечь файлы из zip-архива в любой подходящий каталог.

Процедура установки

Несмотря на то что способ выполнения установщика немного отличается для различных платформ, установщик ведет себя похожим для большинства из них способом.



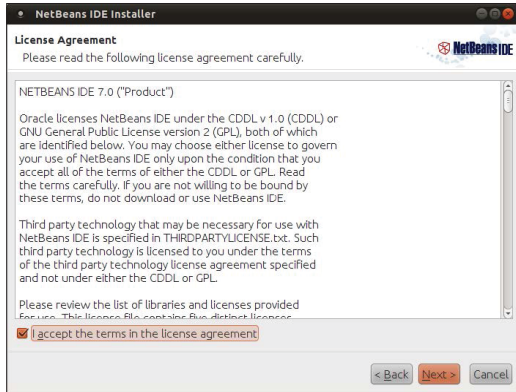
Исключением из этого правила является установщик Mac OSx. Под Mac OSx каждый отдельный компонент (NetBeans, GlassFish, Tomcat и OpenESB) поставляется со своим собственным установщиком и должен быть установлен индивидуально. GlassFish в этом случае должен быть установлен до установки OpenESB. Другим исключением является независимый от платформы zip-файл. В этом случае, по существу, нет никакого установщика вообще, а вся установка этой версии NetBeans состоит в извлечении zip-файла в любой подходящий каталог.

После запуска файла установки NetBeans для нашей платформы мы должны будем увидеть окно, подобное показанному на следующем снимке экрана:



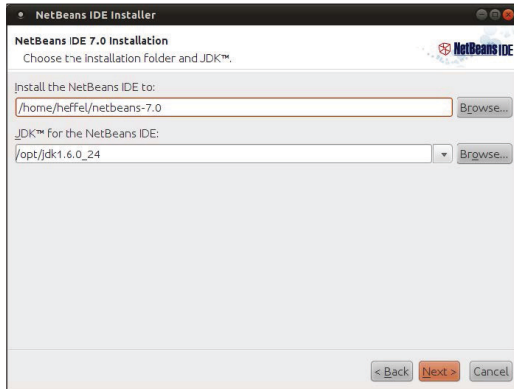
Показанные пакеты могут изменяться в зависимости от загруженного комплекта NetBeans; выше показан снимок экрана, который соответствует комплекту поставки Java EE.

На этом этапе нам нужно нажать на кнопку с надписью **Далее>** (Next>), чтобы продолжить установку.



NetBeans имеет двойную лицензию. Лицензии для NetBeans включают *Публичную лицензию GNU* (GNU Public License (GPL)) v.2 с *Исключением пути к классу* (Class Path Exception (CPE)) и *Общую лицензию на разработку и распространение* (Common Development and Distribution License (CDDL)). Обе эти лицензии одобрены организацией *Инициатива по открытию исходного кода* (Open Source Initiative (OSI)).

Для продолжения установки NetBeans нужно выбрать флажок **Я принимаю условия лицензионного соглашения** (I accept the terms in the license agreement), после чего нажать на кнопку **Далее>** (Next>).



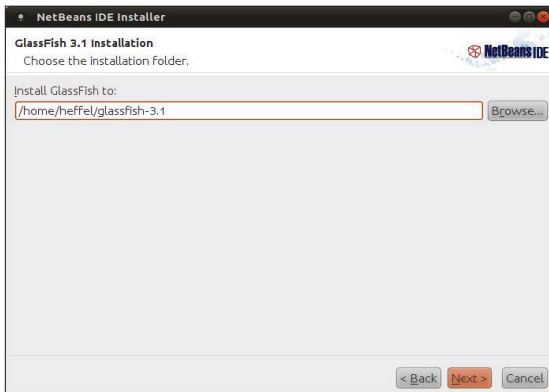
На данном этапе мы должны либо принять условия лицензионного соглашения по JUnit, либо отказаться от установки JUnit.

Теперь установщик запросит у нас каталог, в который должен быть установлен NetBeans, и каталог JDK, который будет использовать NetBeans¹. Мы можем либо выбрать новое значение для него, либо принять значение, используемое по умолчанию.

После того как мы выберем соответствующие каталоги для установки NetBeans и JDK, мы должны щелкнуть по кнопке **Далее>** (Next>) для продолжения установки.

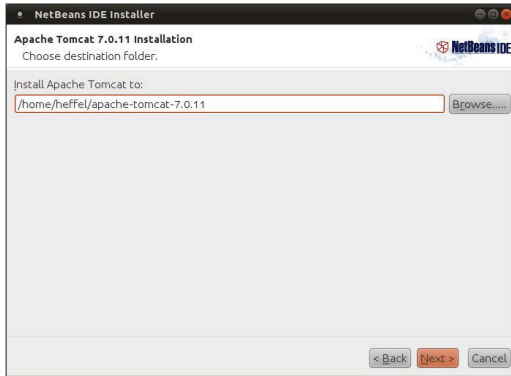


NetBeans использует значение переменной среды `JAVA_HOME` для заполнения поля каталога расположения JDK.

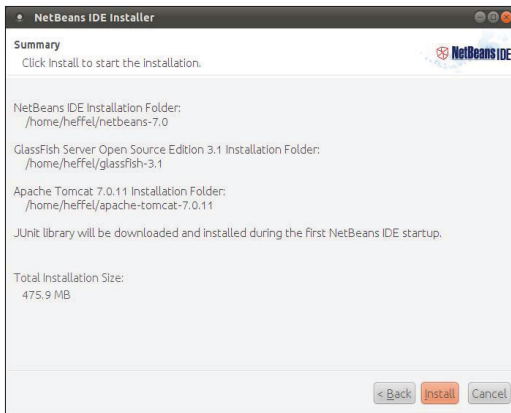


¹ В системе может быть установлено несколько JDK, и они могут иметь различные версии. – Прим. перев.

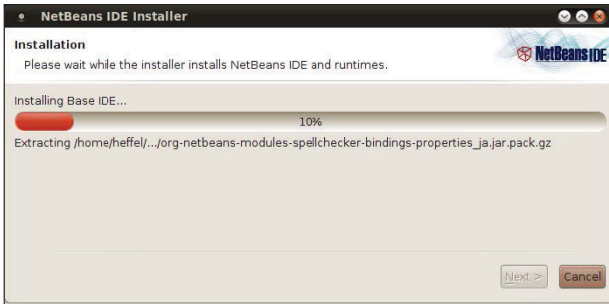
После этого установщик запросит у нас каталог установки сервера приложений GlassFish; мы можем или ввести (выбрать) каталог, или принять значение, предлагаемое по умолчанию.



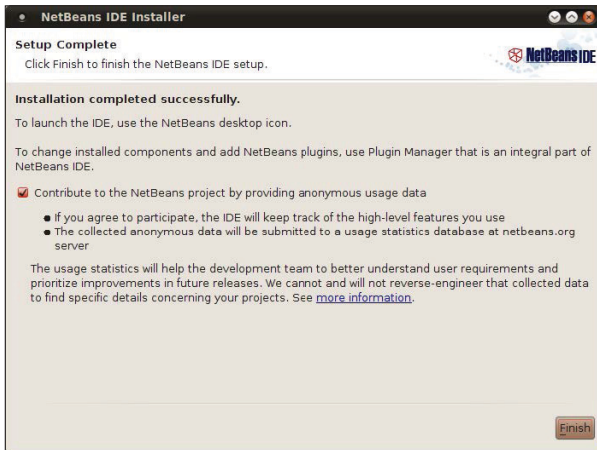
На следующем шаге мастера установщик запросит у нас каталог для установки Tomcat, очень популярного контейнера сервлета, который поставляется вместе с NetBeans.



На этом этапе установщик выведет на экран сводную информацию о выбранных нами опциях установки. После ознакомления со сводными данными нужно щелкнуть по кнопке **Установить** (Install), чтобы начать установку.



После описанных выше действий начнется установка. Установщик выводит на экран индикатор выполнения, указывающий, как далеко продвинулся процесс установки.



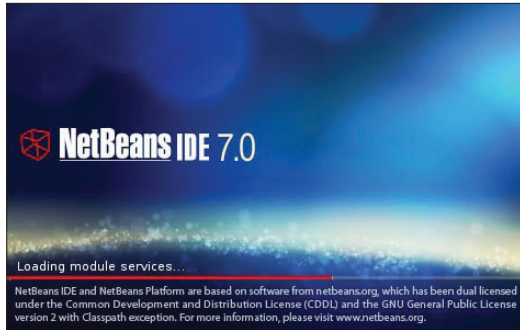
После того как NetBeans и все связанные с ним компоненты будут установлены, установщик сообщит нам об успешной установке, предоставляя опцию для ввода анонимных данных об использовании продукта. После того как мы сделаем наш выбор, можно просто щелкнуть по кнопке **Завершить** (Finish), чтобы выйти из установщика.

Для большинства платформ установщик помещает значок NetBeans на рабочий стол. Значок будет похож на следующее изображение:

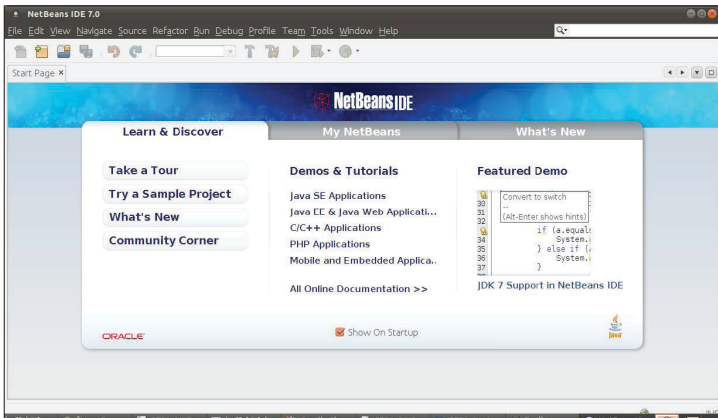


Первый запуск NetBeans

Мы можем запустить NetBeans, дважды щелкнув по его значку на рабочем столе, после чего мы увидим экранную заставку NetBeans, которая будет отображаться в течение всего времени, пока он запускается.



После того как NetBeans запустится, мы увидим страницу со ссылками на демонстрационные примеры, учебные пособия, образцы проектов и т. д.



Значением по умолчанию для NetBeans является показ этой начальной страницы при каждом запуске. Если мы не хотим видеть эту страницу при каждом запуске NetBeans, мы можем отключить данный режим, сняв флажок **Показывать при запуске** (Show on Startup) в нижней части страницы. Мы всегда сможем вернуть отображение начальной страницы, перейдя в главное меню к **Справка** (Help) | **Начальная страница** (Start Page).

Конфигурирование NetBeans для разработки Java EE-приложений

NetBeans поставляется предварительно сконфигурированным с сервером приложений GlassFish v.3 и с СУРБД (RDBMS) JavaDB. Если мы хотим использовать включенные в дистрибутив сервер GlassFish v.3 и СУРБД JavaDB, то нам не нужно ничего делать для конфигурирования NetBeans.

Вместе с тем мы можем интегрировать NetBeans с другими серверами приложений Java EE, например такими, как JBoss, Weblogic или WebSphere, а также с другими системами реляционных баз данных, такими, например, как MySQL, PostgreSQL, Oracle, или любой другой СУРБД, поддерживающей JDBC, что в общем-то означает – с любой СУРБД¹.

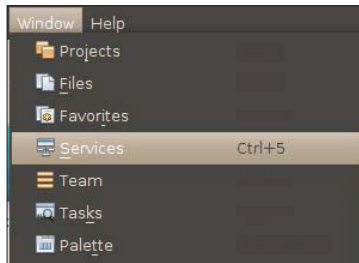
Интегрирование NetBeans со сторонним сервером приложений

Интегрировать NetBeans с сервером приложений очень просто, чтобы это сделать, мы должны выполнить следующие действия:



В этом разделе мы покажем, как интегрировать NetBeans с JBoss, данная процедура будет аналогичной и для других серверов приложений или контейнеров сервлета.

1. Для начала нам нужно щелкнуть по элементу меню **Окно (Window) | Службы (Services)**.

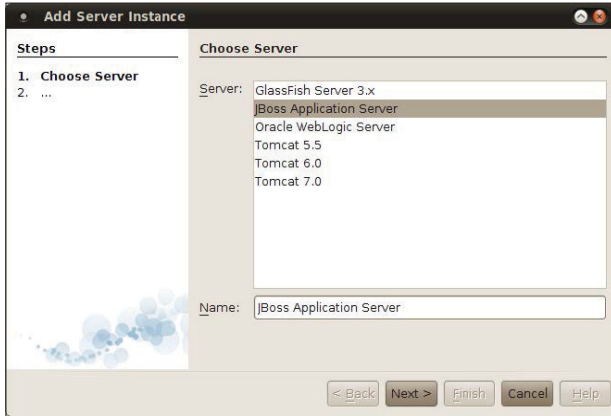


2. Затем нужно щелкнуть правой кнопкой мыши по узлу **Серверы (Servers)** в дереве окна **Службы (Services)** и выбрать **Добавить сервер... (Add Server...)** из открывшегося контекстного меню.

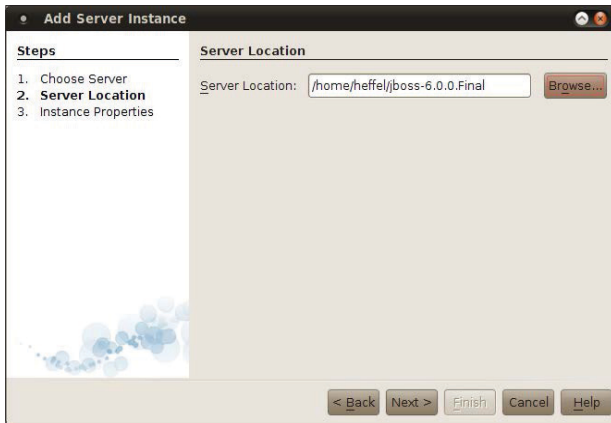


¹ Поскольку поддерживаются практически все существующие в настоящий момент реляционные системы управления базами данных. – *Прим. перев.*

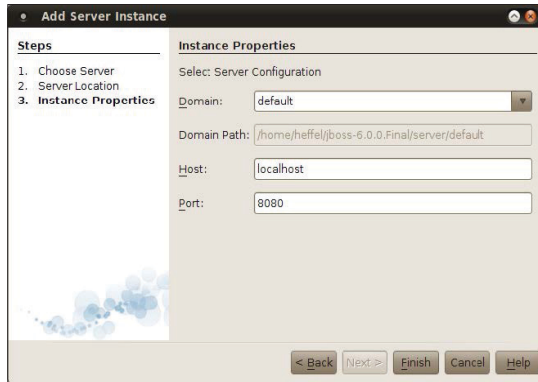
3. После этого в появившемся окне нужно выбрать из списка сервер для установки и нажать кнопку **Далее>** (Next>).



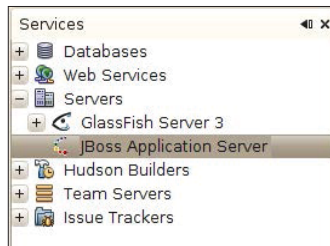
4. Затем необходимо ввести место расположения в файловой системе, где установлен сервер приложений, и нажать кнопку **Далее>** (Next>).



5. Наконец, мы должны выбрать домен, узел и порт для нашего сервера приложений, после чего щелкнуть по кнопке **Завершить** (Finish).



В окне **Службы** (Services) теперь должен отображаться наш новый, добавленный сервер приложений.



Вот и все! Мы успешно интегрировали NetBeans со сторонним сервером приложений.

Интегрирование NetBeans с СУРБД стороннего производителя

NetBeans поставляется интегрированным со встроенной СУРБД – JavaDB. Дополнительно NetBeans поставляется с драйверами JDBC для других систем СУРБД, таких, например, как MySQL и PostgreSQL. Он также поставляется с драйвером моста JDBC-ODBC для установки соединения с системами СУРБД, которые изначально не поддерживают JDBC или для которых нет свободно распространяемого доступного драйвера JDBC.

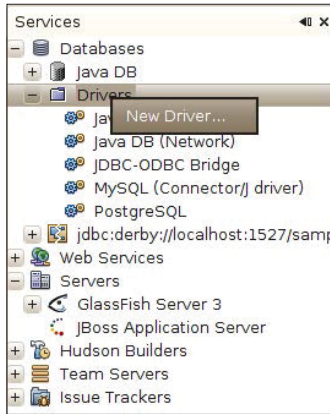
Хотя использование моста JDBC-ODBC позволяет нам соединяться с большинством систем СУРБД без необходимости получения для них драйвера JDBC, обычно лучше все-таки получить драйвер JDBC для требуемой нам СУРБД. Мост JDBC-ODBC не сможет предложить лучшую производительность, кроме того, в настоящее время доступны драйверы JDBC для огромного количества систем СУРБД.



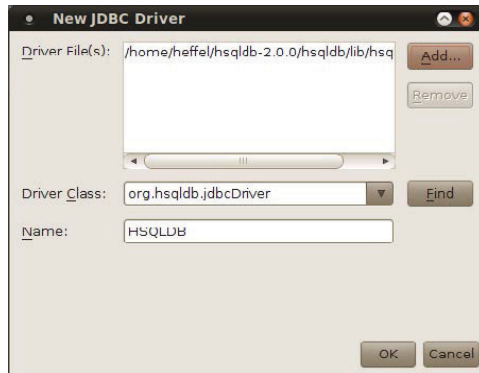
В этом разделе мы создадим соединение с HSQLDB, СУРБД с открытым исходным кодом, написанной на Java. Таким образом будет проиллюстрирована идея того, как интегрировать NetBeans со сторонней системой СУРБД, и данная процедура будет аналогичной для других систем СУРБД, таких как Oracle, Sybase, SQL Server и т. д.

Добавление драйвера JDBC к NetBeans

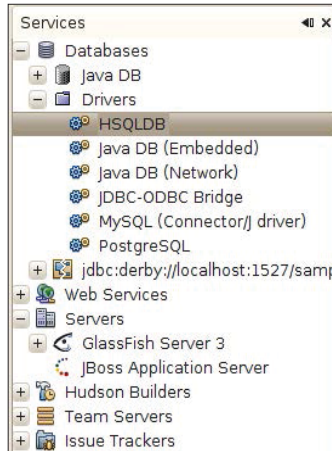
Прежде чем мы сможем установить соединение с СУРБД стороннего производителя, мы должны добавить его драйвер JDBC к NetBeans. Чтобы добавить драйвер JDBC, мы должны щелкнуть правой кнопкой мыши по узлу **Драйверы (Drivers)** узла **Базы данных (Databases)** во вкладке **Службы (Services)**. Затем выбрать из открывшегося контекстного меню пункт **Новый драйвер (New Driver)**.



После этого в открывшемся диалоговом окне нам нужно выбрать JAR-файл, содержащий драйвер JDBC для нашей СУРБД. NetBeans «угадывает» имя класса драйвера, содержащего драйвер JDBC. Если в JAR-файле находится более чем один класс драйвера, правильный драйвер может быть выбран из выпадающего меню с названием **Класс драйвера (Driver Class)**. Нам нужно щелкнуть по кнопке **ОК**, чтобы добавить драйвер к NetBeans.



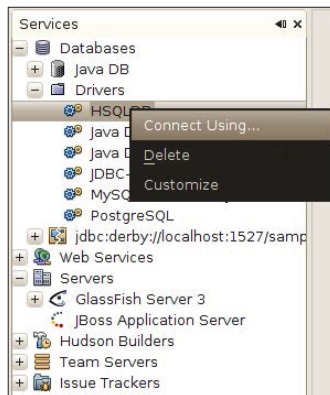
После того как мы выполним вышеупомянутую процедуру, наш новый драйвер JDBC будет выведен на экран в списке зарегистрированных драйверов.



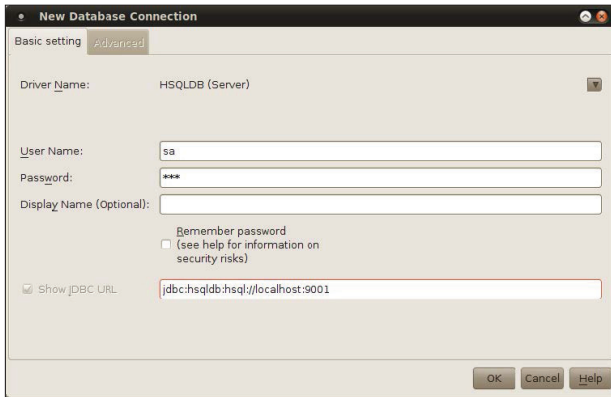
Соединение с СУРБД стороннего производителя

После того как мы добавим драйвер JDBC для нашей СУРБД к NetBeans, мы готовы установить соединение с этой СУРБД стороннего производителя.

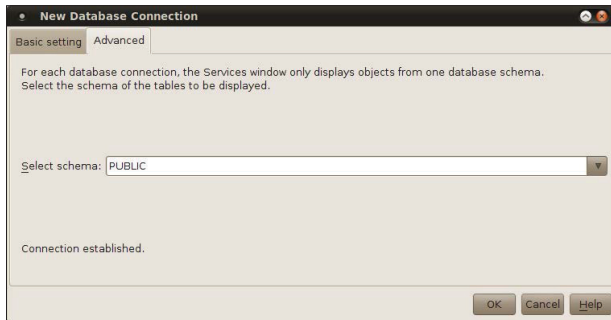
Для установки соединения с нашей СУРБД стороннего производителя мы должны щелкнуть правой кнопкой мыши по ее драйверу во вкладке **Службы** (Services), затем в открывшемся контекстном меню выбрать элемент **Установить соединение с использованием...** (Connect Using...).



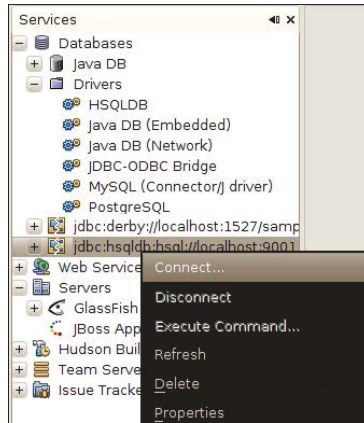
Затем мы должны ввести URL JDBC, имя пользователя и пароль для нашей базы данных.



После щелчка по кнопке **OK** NetBeans может попросить, чтобы мы выбрали схему базы данных.



После выбора схемы и щелчка по кнопке **OK** наша база данных будет показана в списке баз данных в окне **Службы (Services)**. Мы можем установить с ней соединение, щелкнув по ней правой кнопкой мыши и из открывшегося контекстного меню выбрав элемент **Установить соединение (Connect)**, после чего ввести наше имя пользователя и пароль для базы данных (при добавлении базы данных мы можем выбрать режим NetBeans – «не запоминать» пароль).



В результате проделанных действий мы успешно установили соединение NetBeans с СУРБД стороннего производителя.

Развертывание нашего первого приложения

NetBeans поставляется предварительно сконфигурированным с набором примеров приложений. Для того чтобы убедиться, что у нас все настроено правильно, мы развернем один из примеров приложений на интегрированном сервере приложений GlassFish, который поставляется в комплекте с NetBeans.

Чтобы открыть демонстрационный проект, нужно перейти в основное меню в **Файл** (File) | **Создать проект** (New Project), затем в открывшемся диалоговом окне, в списке **Категории** (Categories) выбрать **Примеры** (Samples) | **Java Web**. После того как мы выберем **Java Web** в списке категорий, на экран будет выведен список проектов в списке **Проекты** (Projects), для этого примера мы должны выбрать **Сервлет, не сохраняющий состояние** (Servlet Stateless) (Java EE 6). Этот пример является простым проектом. Он использует и сервлет, и сеансовый бин, не сохраняющий состояние, что дает нам возможность использовать оба контейнера GlassFish – и контейнер сервлета, и контейнер EJB.