

Eclipse:

разработка RCP-, Web-,
Ajax- и Android-приложений
на Java



ОСНОВНЫЕ ВОЗМОЖНОСТИ
РАЗРАБОТКИ ПРИЛОЖЕНИЙ
В СРЕДЕ Eclipse

ГРАФИЧЕСКИЕ СИСТЕМЫ
SWT и JFace

СОЗДАНИЕ
Eclipse-ПЛАГИНОВ

РАЗРАБОТКА ПРИЛОЖЕНИЙ
НА ОСНОВЕ
Eclipse-ПРОЕКТОВ RAP,
GWT, Riena, SCA, Scout,
WTP, DTP и BIRT

Тимур Машнин

Eclipse: разработка RCP-, Web-, Ajax- и Android-приложений на Java

Санкт-Петербург

«БХВ-Петербург»

2013

УДК 681.3.06
ББК 32.973.26-018.2
М38

Машнин Т. С.

М38 Eclipse: разработка RCP-, Web-, Ajax- и Android-приложений на Java. — СПб.: БХВ-Петербург, 2013. — 384 с.: ил. — (Профессиональное программирование)

ISBN 978-5-9775-0829-2

Книга посвящена разработке в среде Eclipse широкого круга Java-приложений. Рассмотрены основы работы в среде Eclipse, использование инструментов отладки, тестирования и рефакторинга кода. Описана командная разработка приложений, их интернационализация и локализация, создание GUI-интерфейса на основе библиотеки SWT и набора Java-классов JFace. Показаны особенности разработки приложений RCP и Android, а также Web- и Ajax-приложений на основе Eclipse-проектов RAP, GWT, Riena, SCA, Scout, WTP, DTP, VIRT. Материал книги сопровождается большим количеством примеров с подробным анализом исходных кодов.

Для программистов

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Игорь Шишигин</i>
Зав. редакцией	<i>Екатерина Капальгина</i>
Редактор	<i>Анна Кузьмина</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Марины Дамбиевой</i>

Подписано в печать 30.09.12.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 30,96.

Тираж 1200 экз. Заказ №

"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

Первая Академическая типография "Наука"
199034, Санкт-Петербург, 9 линия, 12/28

ISBN 978-5-9775-0829-2

© Машнин Т. С., 2013
© Оформление, издательство "БХВ-Петербург", 2013

Оглавление

Введение	7
Проект Business Intelligence and Reporting Tools (BIRT).....	8
Проект Eclipse Data Tools Platform (DTP).....	8
Проект Eclipse	8
Проект Eclipse Modeling Project.....	10
Проект Mylyn.....	11
Проект RT	12
Проект SOA Platform	13
Проект SOA Tools Platform	15
Проект Tools.....	15
Проект Test and Performance Tools Platform	16
Проект Eclipse Web Tools Platform.....	17
Глава 1. Платформа Eclipse и средства разработки Java	19
Архитектура платформы Eclipse и среда Eclipse SDK.....	19
Страница <i>Welcome</i>	21
Рабочая область Workbench	23
Разработка приложений платформы Java SE.....	32
Среда разработки Eclipse SDK.....	32
Пример создания простого Java-приложения.....	33
Навигация по Java-коду	35
Подсказки	37
Запуск выполнения кода.....	38
Расширенные настройки создания JAR-файла	41
Сборка проекта.....	42
Среда разработки Eclipse IDE for Java Developers	44
Инструменты Mylyn.....	45
Интеграция с Maven.....	50
Средства работы с XML	55
Глава 2. Отладка, тестирование и рефакторинг кода	61
Отладка Java-кода	62
Тестирование Java-кода.....	71
Рефакторинг	77

Глава 3. Командная разработка кода	79
CVS	80
Subversion	92
Плагин Subclipse.....	93
Локальный SVN-репозиторий.....	99
Плагин Subversive	101
Git	105
Mercurial.....	117
Глава 4. Интернационализация и локализация приложений.....	123
Глава 5. Графические системы SWT и JFace.....	129
SWT-приложения.....	134
Связывание данных	150
JFace-приложения	155
XWT-приложения	158
Глава 6. Разработка Eclipse-плагинов	160
Мастер <i>Plug-in Project</i>	160
Создание Eclipse-плагина	160
Создание OSGi-модуля.....	173
Мастер <i>Fragment Project</i>	177
Мастер <i>Feature Project</i>	178
Мастер <i>Plug-in from Existing JAR Archives</i>	181
Глава 7. Создание RCP-приложений	182
Глава 8. Создание Android-приложений	193
Инсталляция ADT-плагина	193
Описание ADT-плагина.....	196
Перспектива <i>DDMS</i>	198
Перспективы <i>Hierarchy View</i> и <i>Pixel Perfect</i>	205
Мастера ADT-плагина	207
Мастер <i>Android Project</i>	207
Запуск Android-приложения из среды Eclipse	211
Подготовка к публикации Android-приложения.....	215
Layout-редактор ADT-плагина	218
Редактор файла <i>AndroidManifest.xml</i> ADT-плагина.....	223
Мастер <i>Android XML File</i>	236
Тип ресурса <i>Layout</i>	236
Тип ресурса <i>Values</i>	238
Тип ресурса <i>Drawable</i>	240
Тип ресурса <i>Menu</i>	244
Тип ресурса <i>Color List</i>	246
Тип ресурса <i>Property Animation</i> и <i>Tween Animation</i>	248
Тип ресурса <i>AppWidgetProvider</i>	252
Тип ресурса <i>Preference</i>	255
Тип ресурса <i>Searchable</i>	260
Мастер <i>Android Icon Set</i>	264
Мастер <i>Android Test Project</i>	265

Глава 9. Создание RAP-приложений	268
Глава 10. Создание GWT-приложений	277
Глава 11. Создание приложений на основе платформы Riena	286
Глава 12. Разработка SCA-приложений	297
Глава 13. Разработка приложений на основе платформы Scout	304
Глава 14. Разработка Web-приложений на основе платформы WTP	312
Servlet + JSP	315
Servlet + JSP + JPA	318
Web + EJB	326
Application Client	331
Web-сервисы	332
Глава 15. Управление данными с DTP	341
Глава 16. Создание отчетов с BIRT	348
Глава 17. Использование инструментов Eclipse Modeling Tools	359
EMF	359
GMF	363
Xtext	367
ATL	370
Список литературы	377
Предметный указатель	379

Введение

Eclipse (<http://www.eclipse.org/>) — это сообщество разработчиков и пользователей проектов создания платформ для разработки программного обеспечения, включающих в себя расширяемые каркасы, инструменты и среды выполнения. Управляет проектами Eclipse некоммерческая организация Eclipse Foundation, членами которой являются более 160 всемирно известных компаний ИТ-индустрии (<http://www.eclipse.org/membership/showAllMembers.php>).

Проект Eclipse впервые был представлен сообществу Open Source компанией IBM в 2001 г., а в 2004 г. для его управления была создана организация Eclipse Foundation с советом директоров, состоящим из представителей некоторых компаний — членов организации, являющихся стратегическими разработчиками и потребителями, и включающим в себя выборных представителей разработчиков Open Source и поставщиков Eclipse-плагинов.

ПРИМЕЧАНИЕ

Сообщество Open Source — это сообщество разработчиков и пользователей программного обеспечения с открытым исходным кодом, которое распространяется на условиях лицензий организации Open Source Initiative (OSI) (<http://www.opensource.org/>). Система Open Source разработки программного обеспечения возвращает контроль над ПО пользователю, который может изучать и изменять исходный код ПО, определяет отсутствие произвольной цены за ПО, каких-либо ограничений технологий и отсутствие монополии ПО.

Разработкой проектов Eclipse занимаются разработчики компаний, членов организации Eclipse Foundation, а также независимые разработчики. При этом организация Eclipse Foundation предоставляет с помощью профессиональных сотрудников репозитории кода, базы данных багов, осуществляет рассылки и группы новостей, поддерживает Eclipse-сайт, управляет лицензией Eclipse Public License (EPL), предоставляет помощь в реализации процесса разработки с выпуском ежегодного релиза продуктов, организует конференции и встречи.

Далее приведен список проектов Eclipse (<http://www.eclipse.org/projects/listofprojects.php>), связанных с ними продуктов (<http://www.eclipse.org/downloads/>) и Eclipse-плагинов (<http://marketplace.eclipse.org/>).

В данной книге используется релиз Eclipse 3.7 Indigo.

Проект Business Intelligence and Reporting Tools (BIRT)

Проект Business Intelligence and Reporting Tools (BIRT) обеспечивает создание сложных отчетов Java/Java EE Web-приложений для отображения в Web-браузере. Для этого проект BIRT предоставляет два компонента — Eclipse-дизайнер отчетов и BIRT-среду выполнения сервера приложений. С проектом BIRT связан набор Eclipse-плагинов Business Intelligence, Reporting and Charting и продукты Eclipse IDE for Java and Report Developers и BIRT RCP Report Designer.

Проект Eclipse Data Tools Platform (DTP)

Проект Eclipse Data Tools Platform (DTP) обеспечивает среду для разработки и управления системами данных и призван облегчать управление источниками данных, драйверами источников данных, а также помогать в разработке и тестировании команд и SQL-запросов к источникам данных. С проектом DTP связан набор Eclipse-плагинов Database Development.

Проект Eclipse

Проект Eclipse состоит из подпроектов — Eclipse Platform, Java development tools (JDT) и Plug-in Development Environment (PDE). Вместе реализации этих трех подпроектов составляют среду разработки Eclipse SDK (Software Development Kit), предназначенную для создания программного обеспечения, основанного на Eclipse-платформе, а также для развития самой Eclipse-платформы.

Проект Eclipse представляет продукт Eclipse Classic.

Проект Eclipse Platform обеспечивает базовые каркасы и сервисы, на основе которых создаются все остальные расширения в виде Eclipse-плагинов, а также предоставляет среду выполнения для загрузки, интеграции и запуска Eclipse-плагинов.

Проект Eclipse Platform состоит из следующих подпроектов:

- ◆ Ant — интеграция инструмента Ant, включая выполнение Ant-файлов из среды Eclipse и Ant-задач для среды Eclipse, предоставление интерфейса пользователя для выполнения Ant-файлов и инструментов разработки Ant-файлов;
- ◆ Core — управление ресурсами и среда выполнения;
- ◆ CVS — интеграция инструмента CVS;
- ◆ Debug — независимый от языка механизм отладки кода, обеспечивающий запуск кода, поиск исходного кода, определение и регистрацию контрольных точек отладки, вывод отладочных сообщений, интерфейс пользователя;
- ◆ Releng — тестирование и сборка релизов проекта Eclipse;
- ◆ Search — опция поиска ресурсов панели инструментов;

- ◆ SWT — набор компонентов Standard Widget Toolkit для создания графического интерфейса пользователя Java-приложений;
- ◆ Team/Compare — функция управления версиями ресурсов и их хранилищем, а также сравнения и связывания ресурсов в иерархическую структуру;
- ◆ Text — базовая инфраструктура для редакторов текста;
- ◆ User Assistance — набор компонентов, помогающий пользователю в использовании Eclipse-приложений и включающий в себя набор приветственных страниц, дающих первоначальное представление о приложении, Help-систему документации и учебные примеры;
- ◆ UI — компоненты графического интерфейса пользователя;
- ◆ Update — сервис поиска и инсталляции обновлений, а также управления конфигурацией инсталляции.

Проект Java development tools (JDT) с помощью JDT-плагина обеспечивает среду разработки Java-приложений, включая создание Eclipse-плагинов.

JDT-плагин добавляет перспективу **Java** в панель инструментов и Java-группу шаблонов в команду **New** меню **File**, а также предоставляет набор окон, редакторов и других инструментов для работы с Java-кодом.

Проект JDT включает в себя следующие подпроекты:

- ◆ APT — инструмент Annotation Processing Tool обработки аннотаций Java 5.0;
- ◆ Core — Java-компилятор, управление структурой Java-проекта, функция поиска, форматирование Java-кода и другие базовые компоненты Java-инфраструктуры;
- ◆ Debug — запуск JVM в режиме отладки, вычисление выражений в контексте кадра стека и другие Java-функции отладки;
- ◆ Text — Java-редактор с функциями форматирования, автозавершения, подсказок и др.;
- ◆ UI — окно **Package Explorer**, окно **Type Hierarchy**, окно **Outline** и другие UI-компоненты среды разработки Java IDE.

Проект Plug-in Development Environment (PDE) предоставляет набор инструментов для создания, тестирования, отладки, сборки и развертывания Eclipse-плагинов и других продуктов. С проектом PDE связан PDE-плагин, добавляющий соответствующую перспективу и набор шаблонов и инструментов в среду разработки.

Проект PDE включает в себя следующие подпроекты:

- ◆ PDE Build — основанная на инструменте Ant сборка Eclipse-плагинов;
- ◆ PDE UI — специализированные редакторы, шаблоны, окна и другие графические компоненты работы с Eclipse-плагинами;
- ◆ PDE API Tools — инструмент анализа Eclipse-плагинов на предмет различного рода ошибок;
- ◆ PDE Incubator — инкубатор создания новых инструментов разработки Eclipse-плагинов.

Проект Eclipse Modeling Project

Проект Eclipse Modeling предназначен для объединения Eclipse-проектов, представляющих различные технологии разработки программного обеспечения, основанные на моделях.

Проект Eclipse Modeling объединяет проекты Amalgam, EMF, EMFT, GMP, GMT, MDT, M2M, M2T, TMF.

Проект EMF (Eclipse Modeling Framework Project) представляет собой платформу моделирования с возможностью генерации кода для создания инструментов и приложений на основе структурированной модели данных. Для моделей, которые описаны в формате XMI, EMF обеспечивает инструменты и среду выполнения для создания из модели набора классов Java, представляющих модель, а также обеспечивает создание основы Eclipse-редактора модели и Eclipse-мастера создания экземпляра модели.

Проект EMF имеет следующие подпроекты.

- ◆ Проект EMF (Core) из модели в формате XMI обеспечивает генерацию Java-классов, набора классов адаптеров и базового редактора модели. Модель может быть создана с помощью аннотированного Java-кода, XML-документа и инструментов моделирования, таких как Rational Rose, с последующим импортом в EMF.
- ◆ Проект CDO (Connected Data Objects) обеспечивает Java-репозиторий для хранения совместно используемых EMF-моделей на основе трехуровневой архитектуры, состоящей из клиентского EMF-приложения, CDO-сервера и базы данных, которая может быть реляционной, объектной или файловой и связь которой с CDO-сервером осуществляется с помощью подключаемого к CDO-серверу адаптера. Для совместного использования и разработки моделей CDO-сервер обеспечивает хранение истории ветвей версий графа объектов модели аналогично таким репозиториям, как Subversion или Git.
- ◆ Проект EMF Compare осуществляет сравнение и слияние EMF-моделей.
- ◆ Проект Model Query (MQ) обеспечивает инфраструктуру для определения и выполнения запросов к набору элементов EMF-модели и их содержимому.
- ◆ Проект Model Transaction (MT) выполняет управление EMF-ресурсами.
- ◆ Проект Net4j — расширяемая клиент-серверная система, основанная на Eclipse Runtime и Spring Framework. В частности, клиент-серверный протокол взаимодействия по умолчанию системы CDO реализован с помощью Net4j Signalling Platform.
- ◆ Проект Service Data Objects (SDO) — среда разработки приложений, ориентированных на данные, включающая в себя архитектуру и программный интерфейс, позволяющий работать с данными из различных источников.
- ◆ Проект Teneo обеспечивает хранение EMF-моделей в реляционной базе данных, используя Hibernate или EclipseLink.

- ◆ Проект Validation Framework (VF) осуществляет проверку EMF-модели, гарантируя целостность модели.

Проект Eclipse Graphical Modeling Project (GMP) обеспечивает создание графического редактора для EMF-модели на основе EMF и GEF, где Graphical Editing Framework (GEF) представляет собой платформу создания насыщенных графических редакторов и представлений для Workbench-системы платформы Eclipse. Проект GMP объединяет подпроекты GMF Tooling (генерация графического редактора), GMF Runtime (разработка графического редактора на основе EMF и GEF), GMF Notation (создание описания диаграммы, которое связывает EMF и GEF), Graphiti (основанный на Eclipse каркас, упрощающий разработку редакторов диаграмм).

Проект Textual Modeling Framework (TMF) обеспечивает разработку текстового синтаксиса и текстового редактора для EMF-моделей.

Проект Model Development Tools (MDT) объединяет реализации стандартов визуального моделирования, включая UML2.

Проект Model to Model Transformation (M2M) обеспечивает трансформацию одной EMF-модели в другую EMF-модель.

Проект Model to Text Transformation (M2T) обеспечивает трансформацию EMF-модели в исходный код.

Проект Generative Modeling Technologies (GMT) объединяет различные проекты, связанные с областью Model Driven Engineering (MDE) модельно-ориентированных разработок.

Проект Modeling Amalgamation Project (Amalgam) предназначен для улучшения пакетирования, интеграции и использования компонентов проекта Eclipse Modeling.

Проект Eclipse Modeling Framework Technology (EMFT) объединяет новые технологии, расширяющие и дополняющие EMF.

Компоненты проекта Eclipse Modeling представляют плагины раздела Modeling репозитория Eclipse-релиза, а также продукт Eclipse Modeling Tools.

Проект Mylyn

Проект Mylyn представляет расширение Eclipse-платформы, предназначенное для управления задачами и жизненным циклом приложений (application lifecycle management, ALM). С проектом Mylyn связан набор Eclipse-плагинов Mylyn.

Mylyn-плагины обеспечивают создание, редактирование и просмотр локальных и удаленных задач, распределение задач по времени с отслеживанием их состояния выполнения и связыванием с задачей определенного контекста, что значительно упрощает поиск и навигацию ресурсов в среде Eclipse и ускоряет и повышает эффективность разработки программного обеспечения.

Проект RT

Проект RT объединяет проекты, посвященные различным средам выполнения на основе платформы OSGi/Equinox. Проект RT состоит из подпроектов Apricot, Eclipse Communication Framework, EclipseLink, Equinox, embedded Rich Client Platform, Gemini, Jetty, Rich Ajax Platform, Riena, SMILA, Virgo.

Проект Apricot обеспечивает разработчиков приложений, управляющих контентом, репозиторием контента, доступным с помощью программного интерфейса API. Предназначение проекта Apricot — создание реализации репозитория контента для среды Eclipse на основе платформы Nuxeo.

Проект Nuxeo (<http://www.nuxeo.com/>) предлагает открытую Java-платформу для хранения и управления электронными документами и медиаконтентом.

Проект Eclipse Communication Framework предоставляет платформу для создания коммуникационных Eclipse-приложений, обеспечивая с помощью набора API встраивание в приложение таких компонентов, как клиент мгновенных сообщений, чат, многопользовательский редактор, передача файлов, общий Web-браузер, передача/получение телефонных звонков, клиент удаленных сервисов.

Проект EclipseLink обеспечивает взаимодействие с различными источниками данных, такими как реляционные базы данных, XML-документы, Web-сервисы баз данных. Проект EclipseLink представляет реализацию таких стандартов, как Java Persistence API (JPA), Java API for XML Binding (JAXB), Java Connector Architecture (JCA), Service Data Objects (SDO).

Проект Equinox представляет реализацию спецификации OSGi R4 и обеспечивает основу среды выполнения платформы Eclipse.

Проект embedded Rich Client Platform предоставляет среду выполнения, созданную на основе платформы Eclipse Rich Client Platform путем выделения из нее ограниченного набора, для запуска eRCP-приложений, использующих графическую библиотеку eSWT. Среда eRCP предназначена для работы во встроенных и мобильных устройствах и может быть развернута на таких платформах, как Windows Mobile 2003/2005, Windows Desktop, Nokia Series 60/80 и QT.

Проект Gemini — Enterprise Modules Project представляет собой модульную реализацию технологии Java EE, включающую следующие модули:

- ◆ Gemini Web — Java EE-сервер приложений, работающий в среде выполнения Equinox;
- ◆ Gemini Blueprint — обеспечивает развертывание Spring-приложений в среде выполнения OSGi;
- ◆ Gemini JPA — реализация спецификации Java Persistence API (JPA) для среды выполнения OSGi;
- ◆ Gemini DB Access — доступ к базам данных с помощью JDBC в среде выполнения OSGi;
- ◆ Gemini Management — реализация спецификации Java Management Extensions (JMX) для среды выполнения OSGi;

◆ Gemini Naming — реализация спецификации JNDI Services для среды выполнения OSGi.

Проект Jetty предоставляет Servlet-контейнер и HTTP-сервер с поддержкой Web Sockets, OSGi, JMX, JNDI, JASPI, AJP.

Проект Rich Ajax Platform (RAP) обеспечивает создание RIA-приложений (Rich Internet Application) с использованием Web-реализаций SWT, JFace и Workbench. Программный интерфейс RAP-платформы имеет сходство с программным интерфейсом RCP-платформы, что позволяет легко конвертировать RCP-приложения в RAP-приложения. RAP-платформа включает в себя среду выполнения Equinox со встроенным сервером Jetty и Servlet-контейнером, библиотеки RWT (RAP Widget Toolkit), Web-JFace и Web-Workbench.

RAP-приложение может быть развернуто в любом Servlet-контейнере с сопутствующими плагинами RAP-платформы или в среде выполнения отдельной RAP-платформы. Приложения, созданные и запущенные на платформе RAP, доступны из Web-браузера с помощью HTTP-запроса.

Проект Riena позволяет создавать многоуровневые корпоративные клиент-серверные приложения, основываясь на SOA-возможностях среды Equinox. На основе платформы Riena компоненты корпоративного приложения разрабатываются для целевых платформ и затем разворачиваются на клиентской и серверной сторонах.

Проект SMILA представляет платформу для создания масштабируемых серверных систем обработки неструктурированных данных.

Проект Virgo предоставляет сервер приложений для развертывания Java EE- и Spring-приложений.

С проектом RT связаны RT-плагины, а также продукт Eclipse for RCP and RAP Developers.

Проект SOA Platform

Проект SOA Platform предоставляет платформу сервис-ориентированной архитектуры SOA (Service-Oriented Architecture), включающую в себя SOA-среду выполнения и набор инструментов для разработки и развертывания SOA-проектов.

Проект SOA Platform состоит из подпроектов BPEL Designer, BPMN2 Modeler Project, BPMN modeler, eBAM, eBPM, Java Workflow Tooling, Mangrove — SOA Modeling Framework, SCA Tools, Stardust, Swordfish.

Проект BPEL Designer обеспечивает среду Eclipse инструментами определения, создания, редактирования, развертывания, тестирования и отладки WS-BPEL 2.0 процессов. Проект BPEL Designer основывается на спецификации Web Services Business Process Execution Language Version 2.0 (<http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html>), определяющей бизнес-процессы как набор взаимодействий между Web-сервисами. Проект BPEL Designer предоставляет графический редактор для моделирования BPEL-процессов, EMF-модель, представляющую WS-BPEL 2.0

спецификацию, среду выполнения для развертывания и выполнения BPEL-процессов, отладчик и валидатор BPEL-процессов.

Проект BPMN2 Modeler Project снабжает среду Eclipse графическим редактором для создания и редактирования BPMN-диаграмм с использованием BPMN EMF-модели. BPMN-редактор позволяет моделировать бизнес-процессы на основе спецификации Business Process Model and Notation (BPMN) Version 2.0 (<http://www.omg.org/spec/BPMN/2.0/>).

Проект eBAM (extended Business Activity Monitoring) предоставляет платформу для мониторинга различного рода сервисов и приложений, состоящую из среды выполнения, которая основана на среде Equinox, и инструмента eBAM Designer, включенного в поставляемую среду Eclipse и предназначенного для определения конфигурации мониторинга. Среда выполнения eBAM и инструмент eBAM Designer взаимодействуют с базой данных по умолчанию DBMS H2 для хранения настроек и отчетов.

Проект eBPM (eclipse Business Process Management) предоставляет среду выполнения, основанную на OSGi, и набор инструментов для среды Eclipse, обеспечивая развертывание и управление декларативными OSGi-сервисами. Проект eBPM позволяет разрабатывать, конфигурировать и управлять OSGi-сервисами, основываясь на концепции BPM (Business Process Management). eBPM-инструменты среды Eclipse обеспечивают использование BPMN-редактора для моделирования процессов с последующим созданием BPEL-описания процессов из BPMN-модели. eBPM-среда выполнения расширяет OSGi-среду выполнения BPEL-средой выполнения для оркестрации OSGi-сервисов.

Проект Java Workflow Tooling (JWT) предоставляет инструменты разработки и выполнения рабочих процессов, основанные на SOA-архитектуре. Инструмент WE (Workflow Editor) используется для моделирования рабочих процессов, а инструмент WAM (Workflow engine Administration and Monitoring tool) предназначен для мониторинга и развертывания рабочих процессов. Кроме того, JWT-инструменты позволяют трансформации BPMN-to-JWT, JWT-to-BPMN, JWT-to-XPDL, обеспечивая совместимость платформ Business Process Management (BPM).

Проект Mangrove — SOA Modeling Framework предназначен для интеграции инструментов редактирования, развертывания и выполнения проектов Eclipse SOA Platform. Проект Mangrove служит центральным контейнером для SOA-редакторов, сред выполнения и платформ, обеспечивая трансформацию между SOA-редакторами, синхронизацию SOA-артефактов, взаимодействие с SOA-репозиториями и др.

Проект SCA Tools предоставляет набор инструментов для среды Eclipse, позволяющих создавать SOA-приложения на основе спецификации Service Component Architecture (SCA) (<http://www.oasis-opencsa.org/sca>). С помощью графического редактора SCA Composite Designer создается модель SCA Assembly приложения, основной единицей которой является SCA Composite-артефакт, определяющий набор взаимодействующих сервисных компонентов, использующих и/или реализующих сервисы, доступные удаленно, и являющийся единицей развертывания SCA-

приложения. SCA Composite-артефакт представлен Composite-диаграммой и XML Composite-файлом, на базе которого с помощью инструмента SCA Composite to Java Generator генерируется Java-код основы сервисных компонентов. Для развертывания SCA-приложения можно воспользоваться SCA-средой выполнения Apache Tuscany (<http://tuscany.apache.org/>).

Проект Stardust предоставляет интегрированную среду Business Process Management Suite (BPMS) для проектирования, проверки, симулирования и выполнения бизнес-процессов в среде Eclipse. Stardust-инструменты позволяют моделировать бизнес-процессы, интегрировать их с сервисами и UI-компонентами приложения и запускать их в среде выполнения Stardust Process Engine.

Проект Swordfish расширяет среду Equinox до SOA-платформы, дополняя ее реестром сервисов, системой сообщений, возможностью выполнения бизнес-процессов и др.

Проект SOA Tools Platform

Проект SOA Tools Platform (STP) завершен и большинство его подпроектов перемещено в проект SOA Platform.

Проект STP также содержит различного рода подпроекты, относящиеся к SOA-разработке, включая проект Eclipse Scout.

Проект Eclipse Scout предоставляет платформу для создания распределенных приложений уровня предприятия на основе Eclipse-платформы. Проект Eclipse Scout представлен продуктом Eclipse for Scout Developers.

Проект Tools

Проект Tools объединяет проекты разработки различных инструментов для платформы Eclipse Platform и состоит из подпроектов AJDT — AspectJ Development Tools Project, AspectJ, Ajax Tools Framework (ATF), Buckminster Component Assembly, C/C++ Development Tooling (CDT), EGL Development Tools, Graphical Editing Framework (GEF), Memory Analyzer, Object Teams, Eclipse Orbit Project, PHP Development Tools, Parallel Tools Platform (PTP), Runtime Analysis Tools, Sequoyah, Target Management, WindowBuilder, Xtend.

Проект AJDT — AspectJ Development Tools Project обеспечивает аспектно-ориентированную разработку в среде Eclipse с использованием языка программирования AspectJ.

Проект AspectJ предоставляет реализацию языка программирования AspectJ.

Проект Ajax Tools Framework (ATF) обеспечивает для среды Eclipse инструменты редактирования, отладки и мониторинга CSS-, HTML- и JavaScript-приложений.

Проект Buckminster Component Assembly облегчает разработку комплексных приложений, обеспечивая механизм материализации исходных и бинарных артефактов проекта из многочисленных проектов и репозиториев.

Проект C/C++ Development Tooling (CDT) обеспечивает в среде Eclipse разработку приложений с использованием языков C/C++.

Проект EGL Development Tools обеспечивает в среде Eclipse разработку приложений с использованием языка программирования EGL (Enterprise Generation Language).

Проект Graphical Editing Framework (GEF) предоставляет платформу для создания насыщенных графических редакторов и представлений для графического интерфейса Eclipse Workbench UI.

Проект Memory Analyzer предоставляет инструменты анализа дампа Java-кучи.

Проект Object Teams обеспечивает в среде Eclipse разработку приложений с использованием языка программирования OT/J.

Проект Eclipse Orbit Project обеспечивает репозиторий для сторонних библиотек, которые одобрены для использования в проектах Eclipse.

Проект PHP Development Tools (PDT) обеспечивает в среде Eclipse разработку Web-приложений с использованием языка программирования PHP.

Проект Parallel Tools Platform (PTP) обеспечивает в среде Eclipse разработку приложений, организующих параллельные вычисления на языках программирования C, C++ и Fortran для параллельных компьютерных систем. Проект PTP предоставляет инструменты разработки приложений на основе технологий Message Passing Interface (MPI), OpenMP, UPC и др., поддержку сред выполнения PBS/Torque, LoadLeveler, GridEngine, Parallel Environment, Open MPI и MPICH2, параллельный отладчик и поддержку интеграции параллельных инструментов. Проект PTP представлен продуктом Eclipse IDE for Parallel Application Developer.

Проект Runtime Analysis Tools (RAT) предоставляет инструменты анализа производительности и использования памяти Java-приложений.

Проект Sequoyah предоставляет инструменты разработки приложений для мобильных устройств.

Проект Target Management обеспечивает конфигурирование и управление удаленными системами, включая их соединения и сервисы.

Проект WindowBuilder предоставляет набор визуальных графических редакторов и мастеров для создания GUI-интерфейсов eRCP-, Swing-, GWT-, SWT- и XWT-приложений.

Проект Xtend добавляет поддержку языка программирования Xtend в среду Eclipse.

Проект Test and Performance Tools Platform

Проект Eclipse Test and Performance Tools Platform (TPTP) представляет архитектуру и компоненты реализации, расширяющие платформу Eclipse для включения инструментов тестирования, анализа производительности и мониторинга приложений. Этот проект позволяет тестировать, профилировать и осуществлять мониторинг кода приложений, а также создавать новые инструменты, расширяющие

TRTP-платформу. Проект TRTP состоит из подпроектов Tracing & Profiling Tools, TRTP Platform и Testing Tools.

Проект Tracing & Profiling Tools расширяет TRTP-платформу, обеспечивая сбор и анализ информации о производительности приложения. Кроме того, данный проект предоставляет каркас для создания инструментов профилирования и трассировки на основе TRTP-платформы.

Проект TRTP Platform представляет TRTP-платформу, служащую ядром и базой для других TRTP-проектов, обеспечивая общий GUI-интерфейс, стандартную модель данных, общую инфраструктуру сбора данных и коммуникаций, а также удаленную среду выполнения.

Проект Testing Tools обеспечивает редактирование, развертывание и выполнение тестов, а также предоставляет каркас для создания инструментов тестирования на основе TRTP-платформы.

Проект Eclipse Web Tools Platform

Проект Eclipse Web Tools Platform (WTP) расширяет платформу Eclipse инструментами разработки Web- и Java EE-приложений. Проект Eclipse Web Tools Platform содержит подпроекты WTP Common Tools, Dali Java Persistence Tools, Datatools, WTP EJB Tools, WTP Incubator, Java EE Module Configuration Editors, WTP Java EE Tools, JavaScript Development Tools, JavaServer Faces, Enterprise Tools for the OSGi Service Platform, Pave, Webtools Releng, Server Tools, WTP Source Editing, Web Services Tools.

Проект WTP Common Tools предоставляет общую инфраструктуру для WTP-платформы, включая такие компоненты, как Facet API, Validation, Snippets View, Extensible URI Resolver.

Проект Dali Java Persistence Tools предоставляет каркас и инструменты для определения и редактирования JPA O/R (Java Persistence API Object-Relational) отображения, обеспечивая сохранение объектов Java POJOs в реляционных базах данных.

Проект Datatools является подпроектом проекта Eclipse Data Tools Platform (DTP), обеспечивающим разработку с использованием и управление системами данных, включая определение и управление драйверами источников данных, конфигурирование доступа к источникам данных, установление соединения с источниками данных и обеспечение доступа к ним с использованием языка SQL.

Проект WTP EJB Tools добавляет поддержку спецификаций Enterprise JavaBeans 1.1, 2.0, 2.1, 3.0.

Проект WTP Incubator содержит компоненты на стадии разработки, не включенные в релиз проекта WTP.

Проект Java EE Module Configuration Editors обеспечивает редактор, объединяющий информацию конфигурационных XML-файлов и Java-аннотаций Java EE-модуля.

Проект WTP Java EE Tools предоставляет каркасы и инструменты, обеспечивающие в процессе разработки приложения создание Java EE-артефактов.

Проект JavaScript Development Tools (JSDT) расширяет Eclipse-платформу до среды разработки JavaScript-приложений, добавляя JavaScript-перспективу, мастер создания JavaScript-проекта, а также набор представлений, редакторов, мастеров и сборщиков.

Проект JavaServer Faces (JSF) Tools упрощает разработку и развертывание JSF-приложений.

Проект Enterprise Tools for the OSGi Service Platform (Libra) связывает вместе инструменты PDE и WTP, обеспечивая разработку OSGi Enterprise-приложений.

Проект Pave предоставляет платформу для создания шаблонов генерации ресурсов проекта в Workspace-пространстве, а также обеспечивает простой шаблон для создания каркаса шаблона и шаблон Crud Session Bean Pattern создания сессионного компонента с CRUD-операциями для Entity-компонентов.

Проект Webtools Releng (WTP Release Engineering) обеспечивает сборку и тестирование проекта Web Tools Platform (WTP).

Проект Server Tools обеспечивает поддержку J2EE-, JEE- и HTTP-серверов для среды Eclipse.

Проект WTP Source Editing обеспечивает редактирование ресурсов XML, XML Schema, XSL, HTML, CSS, DTD и JSP, а также предоставляет перспективу XML для среды Eclipse.

Проект Web Services Tools предоставляет для среды Eclipse набор инструментов разработки Web-сервисов.

Проект Eclipse Web Tools Platform представляют продукты Eclipse IDE for Java EE Developers и Eclipse IDE for JavaScript Web Developers.



ГЛАВА 1

Платформа Eclipse и средства разработки Java

Архитектура платформы Eclipse и среда Eclipse SDK

Платформа Eclipse является фундаментом, на основе которого с помощью Eclipse-плагинов создаются все остальные Eclipse-продукты.

В свою очередь, Eclipse-платформа состоит из набора подсистем, которые представлены также Eclipse-плагинами, работающими в среде выполнения Eclipse-платформы.

Из компонентов Eclipse-платформы можно выделить минимальный набор Eclipse-плагинов, известный как Rich Client Platform (RCP), на основе которого возможно создание любых клиентских приложений. Поэтому можно сказать, что та же среда Eclipse — это RCP-приложение. Платформа RCP включает в себя такие компоненты, как среда выполнения на основе OSGi, библиотеки SWT и JFace, графическая многооконная Workbench-среда и связанные с ней компоненты.

Eclipse-платформа может быть разделена на подсистемы согласно подпроектам проекта Eclipse Platform (*см. введение*) или в соответствии с набором основных предоставляемых функций. Такое деление по основной функциональности дает следующий набор компонентов Eclipse-платформы.

- ◆ Platform Runtime — основанная на OSGi среда выполнения.
- ◆ Workbench — набор графических инструментов, созданных на основе библиотек SWT и JFace.
- ◆ Workspace — рабочее пространство, физически представленное каталогом локальной файловой системы, в котором находятся Eclipse-проекты. Eclipse-платформа обеспечивает синхронизацию и управление Workspace-ресурсами, позволяя определить единые глобальные настройки для всех ресурсов в пределах одного рабочего пространства Workspace. Метаданные Workspace-пространства хранятся в папке .metadata его каталога. Создание своего Workspace-пространства для группы проектов определенного типа способствует грамотной организации процесса разработки. Eclipse-проект — это набор файлов, скомпо-

нованных согласно типу проекта и сопровождаемых файлом .PROJECT метаданных проекта.

- ◆ **Team** — обеспечивает командную разработку кода под контролем версий.
- ◆ **Help** — встроенная документация, содержащая набор электронных книг. При выборе меню **Help | Help Contents** открывается окно встроенного Web-браузера и запускается встроенный сервер Apache Tomcat, обеспечивающий отображение содержимого электронных книг, каждая из которых организована в виде Eclipse-плагина.

Набор Workbench-инструментов обеспечивает графический интерфейс пользователя Eclipse-платформы. Каждое Workbench-окно, открываемое при запуске среды Eclipse, содержит одну или несколько *перспектив*. Каждая перспектива Workbench-окна — это компоновка редакторов и представлений (окон) в конкретный набор, сопровождающийся определенными меню и панелями инструментов и соответствующий определенному типу выполняемой задачи. При этом одна перспектива Workbench-окна отличается от другой перспективы данного Workbench-окна отображаемым набором представлений, но использует общий набор редакторов.

Одновременно можно открыть несколько Workbench-окон с помощью выбора команды **New Window** в меню **Window**. При этом для каждого Workbench-окна может быть открыта только одна перспектива.

Сама по себе Eclipse-платформа содержит перспективы навигации ресурсов и поддержки командной разработки. Другие перспективы добавляются Eclipse-плагинами, расширяющими Eclipse-платформу до конкретной среды разработки Eclipse IDE. В частности, JDT-плагин добавляет в Eclipse-платформу перспективы, помогающие в разработке Java-приложений.

Перспектива контролирует только первоначальное отображение компоновки представлений и окна редактора. Пользователь может перекомпоновать этот набор, который сохранится при закрытии среды Eclipse.

Новая перспектива открывается с помощью команды **Open Perspective** меню **Window**.

Eclipse-плагины добавляют к Eclipse-платформе новые типы редакторов, представлений и перспектив. К существующим редакторам, представлениям и перспективам могут добавляться новые действия в меню и панелях инструментов.

Eclipse-редакторы обеспечивают открытие, редактирование и сохранение объектов. Сама Eclipse-платформа содержит только редактор текстовых ресурсов, другие редакторы добавляются Eclipse-плагинами. Eclipse-редактор загружается в соответствующее окно рабочей области Workbench при двойном щелчке мышью на ресурсе, отображаемом в представлении.

Eclipse-представления обеспечивают дополнительную информацию об объектах, с которыми идет работа в Workbench-окне, и открываются с помощью команды **Show View** меню **Window**.

Проект Eclipse Platform является подпроектом проекта Eclipse. Проект Eclipse представлен продуктом Eclipse Classic (Eclipse SDK), содержащим Eclipse-плагины

Eclipse-платформы, JDT (Java Development Tools) и PDE (Plug-in Development Environment).

ПРИМЕЧАНИЕ

Далее описывается работа со средой Eclipse SDK в операционной системе Windows.

Перед инсталляцией среды Eclipse SDK требуется установка JDK (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>).

После скачивания ZIP-архива дистрибутива среды Eclipse SDK (<http://www.eclipse.org/downloads/>) требуется просто его распаковать. Для запуска среды Eclipse SDK дважды щелкнем мышью на исполняемом файле eclipse.exe каталога дистрибутива — после чего начнется загрузка Workbench-окна.

Перед тем как Workbench-окно будет открыто, появится диалоговое окно, запрашивающее расположение Workspace-пространства в локальной файловой системе компьютера.

Страница *Welcome*

Первое, что появится на экране компьютера после определения Workspace-пространства, — это страница приветствия **Welcome** (рис. 1.1).

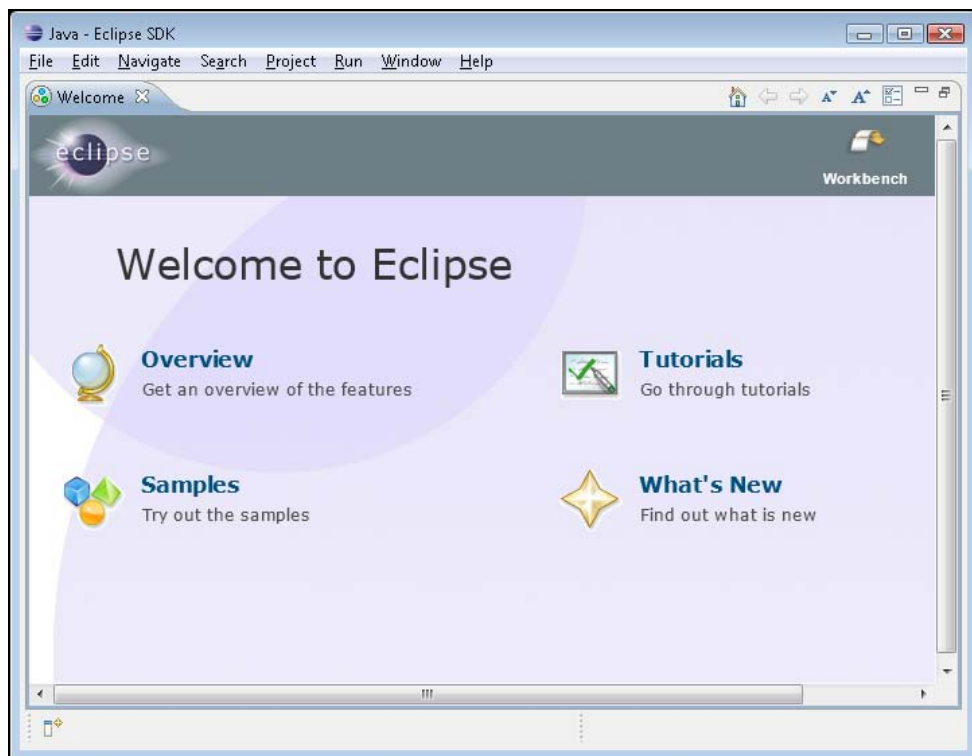


Рис. 1.1. Страница приветствия **Welcome**

Страницу **Welcome** можно также открыть с помощью команды **Welcome** меню **Help**.

Страница **Welcome** содержит кнопку **Workbench**, закрывающую эту страницу, а также гиперссылки:

- ◆ **Overview** — переход на страницу обзора среды Eclipse SDK, содержащую кнопки, которые открывают Help-документацию с электронными книгами "Workbench User Guide", "Java development user guide", "Platform Plug-in Developer Guide", "JDT Plug-in Developer Guide" и "Plug-in Development Environment Guide". Эти же книги можно открыть с помощью команды **Help Contents** меню **Help**;
- ◆ **Tutorials** — переход на страницу учебных примеров, содержащую кнопки, которые открывают окна **Cheat Sheet** в Workbench-окне, последовательно проводящие пользователя через учебные примеры создания простого Java-приложения, SWT-приложения, командной разработки проекта, создания Eclipse-плагина и RCP-приложения;
- ◆ **Samples** — переход на страницу примеров, содержащую кнопки, которые открывают окна **Cheat Sheet** в Workbench-окне, последовательно проводящие

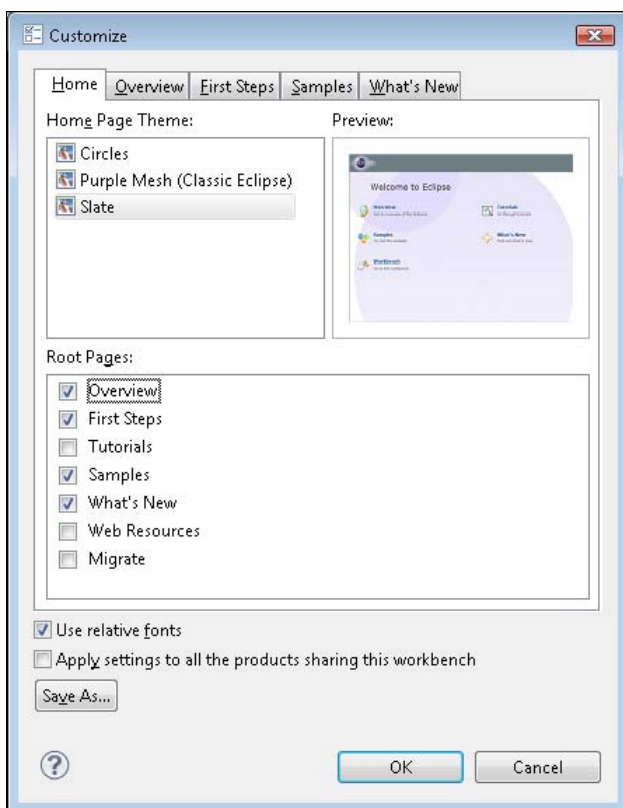


Рис. 1.2. Диалоговое окно изменения страницы **Welcome**

пользователя через учебные примеры интеграции с Workbench-окном, создания пользовательского Java-редактора и SWT-примеры;

- ◆ **What's New** — переход на страницу обзора нововведений, содержащую кнопки, которые открывают соответствующие разделы Help-документации, запускают проверку обновлений и открывают страницы сайта <http://www.eclipse.org/> в Web-браузере.

Страница **Welcome** содержит в правом верхнем углу набор кнопок управления, среди которых есть кнопка **Customize page**, позволяющая изменить внешний вид и содержание страницы Welcome (рис. 1.2).

Рабочая область Workbench

После закрытия страницы **Welcome** на экране компьютера появится содержимое рабочей области Workbench.

Первоначально отобразится перспектива **Java**. Для того чтобы переключиться в перспективу **Resource** Eclipse-платформы, последовательно выберем в меню **Window** команды **Open Perspective | Other | Resource**. В результате в Workbench-окне появятся три представления: **Project Explorer**, **Outline** и **Tasks**, а также окно для редактора ресурсов (рис. 1.3).

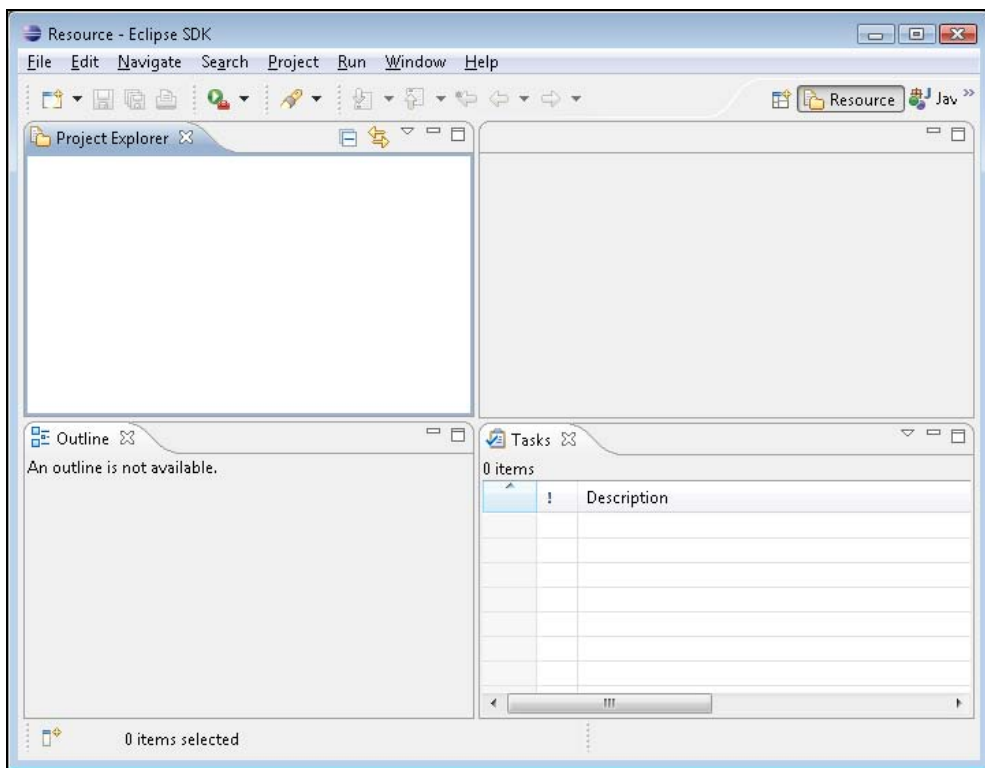




Рис. 1.3. Перспектива **Resource** среды Eclipse SDK

Для того чтобы изменить набор отображаемых представлений, можно воспользоваться командой **Show View** меню **Window**. Также среда Eclipse дает возможность перетаскивать представления в Workbench-окне мышью и изменять их размеры, минимизировать, максимизировать и закрывать представления, используя контекстное меню вкладки представления или его панель инструментов. Вернуться к первоначальному набору Eclipse-представлений позволяет команда **Reset Perspective** меню **Window**. Измененный набор представлений можно сохранить в виде новой перспективы с помощью команды **Save Perspective As** меню **Window**. Удалить перспективу можно в разделе **Perspectives** диалогового окна, открываемого командой **Preferences** меню **Window**. Настроить перспективу позволяет опция **Customize Perspective** меню **Window**.

Eclipse-представление имеет два контекстных меню. Одно меню появляется при нажатии правой кнопкой мыши на вкладке меню, а другое — при нажатии кнопки  **View Menu** панели инструментов представления.

Eclipse-представление можно определить в качестве представления **Fast View** — такое представление отображается в виде значка в левом нижнем углу Workbench-окна и при нажатии на значок сразу открывается в Workbench-окне. Установить представление в качестве **Fast View** можно с помощью кнопки  **Show View as a Fast View**, расположенной в левом нижнем углу Workbench-окна, или с помощью команды **Fast View** контекстного меню представления.

Eclipse-представление также можно определить в качестве Detached-представления — такое представление отображается в отдельном окне вне Workbench-окна. Установить представление в качестве Detached-представления можно с помощью команды **Detached** контекстного меню представления.

Платформа Eclipse имеет перспективы **Resource**, **Team Synchronizing** и **CVS Repository Exploring**.

Перспектива **Resource** имеет окно редактора и представления **Project Explorer**, **Outline** и **Tasks** (табл. 1.1).

*Таблица 1.1. Представления перспективы **Resource***

Представление	Описание
Project Explorer	Отображает дерево ресурсов
Outline	Отображает структуру файла, открытого в данный момент в редакторе
Tasks	Отображает список маркеров задач

Перспектива **Team Synchronizing** имеет окно редактора и представления **Synchronize**, **History**, **Tasks** и **Problems** (табл. 1.2).

Перспектива **CVS Repository Exploring** имеет окно редактора и представления **History** и **CVS Repositories**. Представление **CVS Repositories** отображает структуру CVS-хранилища, добавленного в Workbench-окно.

Таблица 1.2. Представления перспективы **Team Synchronizing**

Представление	Описание
Synchronize	Обеспечивает сравнение локальных и удаленных ресурсов, обновление локальных ресурсов и их передачу в репозиторий
History	Отображает список изменений ресурса в репозитории и локальную историю ресурса
Tasks	Отображает список маркеров задач
Problems	Отображает список ошибок и предупреждений

Создадим простой Eclipse-проект. Для этого в меню **File** последовательно выберем команды **New | Other | General | Project** и нажмем кнопку **Next** — запустится мастер создания проекта. Введем имя проекта SimpleProject и нажмем кнопку **Finish**. В результате средой Eclipse SDK будет создана папка SimpleProject в каталоге workspace с файлом .PROJECT описания проекта. Файл .PROJECT идентифицирует набор файлов и папок как Eclipse-проект таким образом, чтобы при переносе данного набора в другой каталог файловой системы его можно было бы импортировать в Workbench-окно с помощью команды **Import** меню **File**.

Среда Eclipse содержит большой набор мастеров создания ресурсов, импорта и экспорта ресурсов и др. Мастера призваны помочь пользователю выполнить ту или иную задачу в Workbench-окне, и текущий набор мастеров расширяется за счет Eclipse-плагинов.

Для создания папки и текстового файла проекта можно использовать команду **New** контекстного меню, появляющегося при нажатии правой кнопкой мыши на узле проекта в окне **Project Explorer**, можно воспользоваться кнопкой **New** панели инструментов Workbench-окна или выбрать команду **New** меню **File**.

Последовательно выберем команды **New | Other | General | Folder**, введем имя папки projectfolder и нажмем кнопку **Finish**. В папке projectfolder создадим текстовый файл с помощью выбора команд **New | Other | General | File**, ввода имени файла text.txt и нажатии кнопки **Finish**.

В результате созданный файл text.txt будет открыт в текстовом редакторе Workbench-окна (рис. 1.4).

Если в редакторе набрать текст, то на вкладке **text.txt** появится звездочка (*), указывающая, что изменения файла text.txt не сохранены. Для сохранения изменений файла text.txt можно нажать кнопку **Save** панели инструментов Workbench-окна (рис. 1.5).

Если создать текстовый файл с расширением, например, не txt, а doc, тогда среда Eclipse откроет созданный файл не в текстовом редакторе Eclipse-платформы, а в редакторе Microsoft Word операционной системы, который загрузится как OLE-объект в окно редактора (рис. 1.6).

Для того чтобы открыть файл в определенном редакторе, можно воспользоваться командой **Open With** контекстного меню, открывающегося при щелчке правой

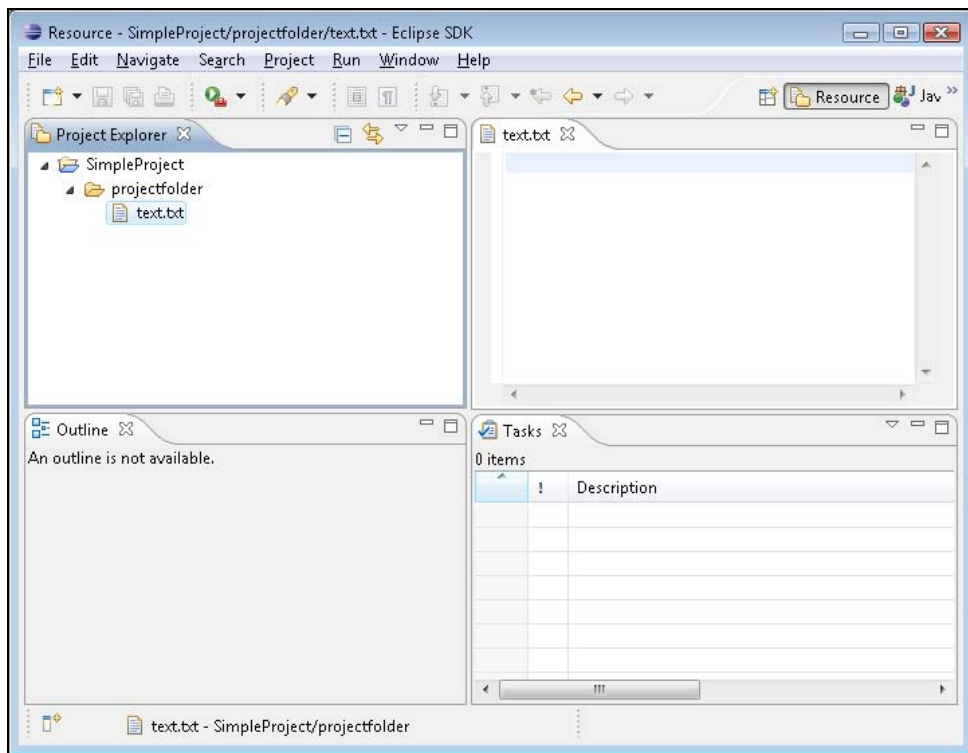


Рис. 1.4. Создание простого Eclipse-проекта с папкой и текстовым файлом



Рис. 1.5. Сохранение изменений текстового файла

кнопкой мыши на узле файла в окне **Project Explorer**. Например, при выборе команды **Open With | System Editor** текстовый файл `text.txt` откроется в Блокноте операционной системы Windows.

Для настройки текстового редактора, а также определения соответствий файловых расширений определенным редакторам можно использовать раздел **Editors** диалогового окна **Preferences**, открываемого при выборе одноименной команды в меню **Window** (рис. 1.7).

Помимо непосредственно создания папки или файла Eclipse-проекта, при котором соответствующая папка или файл появляются в каталоге Eclipse-проекта, Eclipse-платформа позволяет создание так называемых связанных ресурсов. *Связанный ресурс* — это папка или файл, физически находящиеся не в каталоге Eclipse-

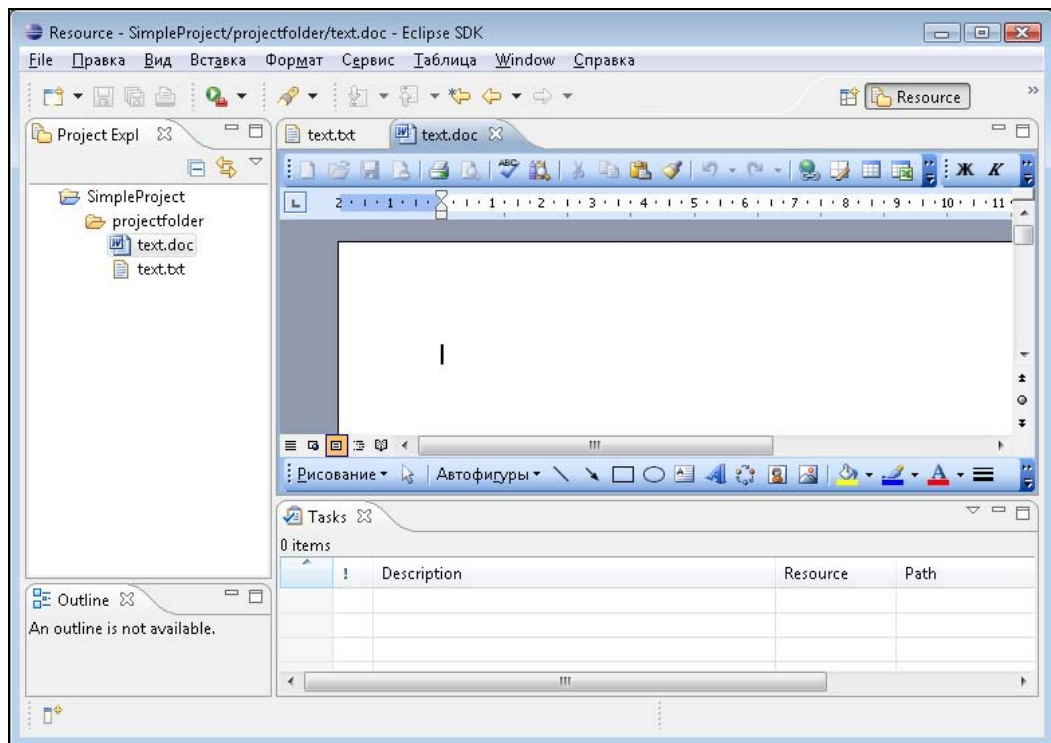


Рис. 1.6. Создание текстового файла с расширением, не совместимым с текстовым редактором Eclipse-платформы

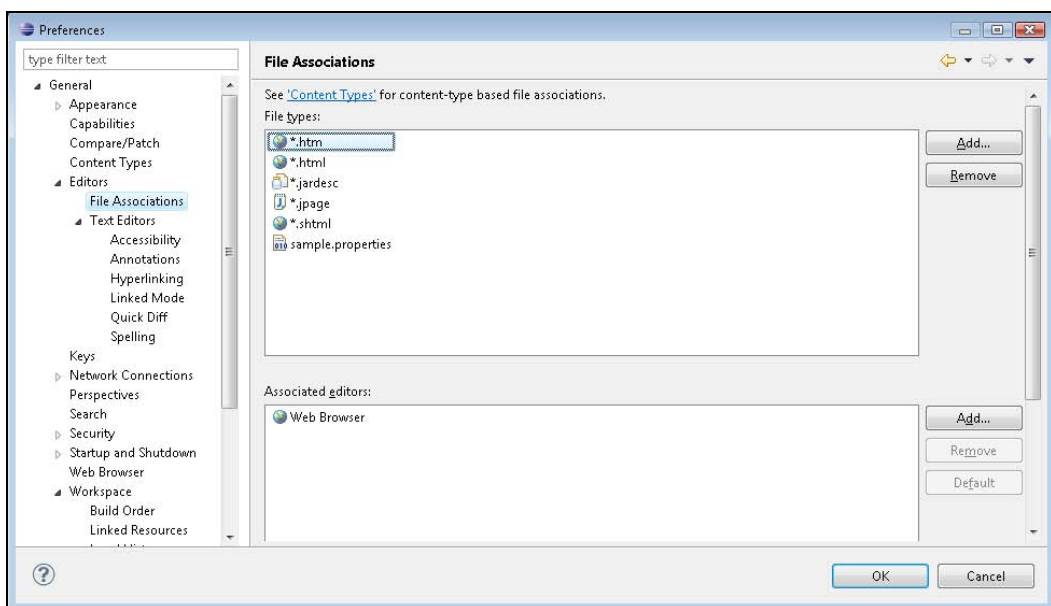


Рис. 1.7. Настройка соответствия файлового расширения определенному редактору

проекта, а в другом месте файловой системы, однако при этом присутствуют в качестве узла Workbench-окна. Создать такой связанный ресурс можно с помощью команды **New**, нажатия кнопки **Advanced**, отметки флажка **Link to file in the file system** или **Link to alternate location (Linked Folder)** и кнопки **Browse** (рис. 1.8), определяющей путь к ресурсу. Кроме кнопки **Browse** можно использовать кнопку **Variables**, которая открывает список переменных пути, и задать расположение ресурса с помощью переменной пути. Создать переменную пути можно нажатием кнопки **New** вкладки **Path Variables** раздела **Resource | Linked Resources** диалогового окна, открываемого командой **Properties** контекстного меню узла проекта или с помощью кнопки **New** раздела **General | Workspace | Linked Resources** диалогового окна **Preferences**, открываемого одноименной командой в меню **Window**.

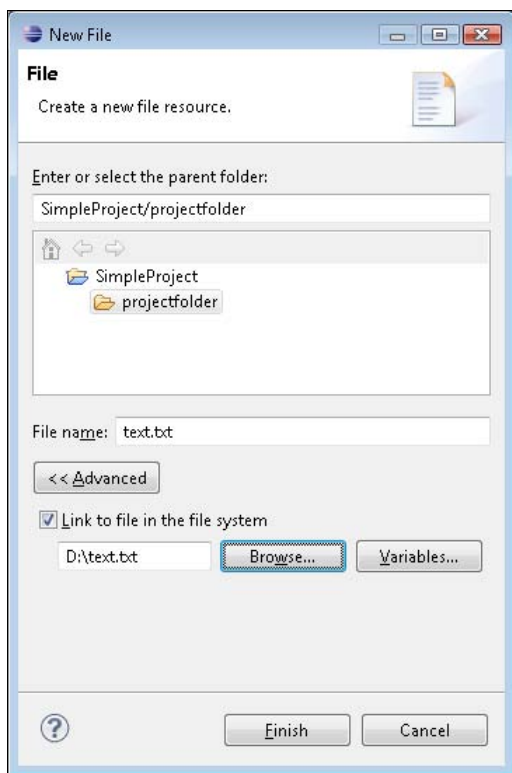


Рис. 1.8. Создание связанного файла

Список связанных ресурсов может быть отредактирован с помощью команд **Properties | Resource | Linked Resources** контекстного меню узла проекта. Отключить саму опцию связанных ресурсов можно, используя раздел **General | Workspace | Linked Resources** в окне **Preferences**, открываемом одноименной командой в меню **Window**.

Связанные ресурсы могут быть организованы в Eclipse-проекте в иерархическую структуру с помощью виртуальных папок. Виртуальная папка физически не существует в файловой системе, а присутствует в качестве узла Workbench-окна.

Создать виртуальную папку можно с помощью команд **New | Folder**, нажатия кнопки **Advanced** и отметки флажка **Folder is not located in the file system (Virtual Folder)**.

Создать ресурс проекта можно не только с помощью опции **New**. Готовую папку или файл также можно импортировать в Eclipse-проект путем перетаскивания мышью из окна, например, Проводника в Workbench-окно, используя операцию копирования/вставки или команду **Import** контекстного меню окна **Project Explorer**.

Среда Eclipse разрешает и обратную операцию экспорта папок или файлов из Workbench-окна в файловую систему компьютера с помощью перетаскивания мышью, используя операцию копирования/вставки или команду **Export** Workbench-окна.

Удалить ресурс проекта в Workbench-окне можно посредством команды **Delete** контекстного меню окна **Project Explorer**, выбора ресурса и нажатия клавиши или с помощью команды **Delete** меню **Edit**.

Контекстное меню окна **Project Explorer** позволяет также переименовывать и перемещать ресурсы проекта с помощью команд **Rename** и **Move** соответственно.

Поиск ресурсов или текста осуществляется посредством меню **Search** Workbench-окна. При этом поиск файлов может быть реализован с учетом файлового расширения и с учетом содержащегося в них текста (рис. 1.9). Результаты поиска отображаются в открывающемся представлении **Search** Workbench-окна.

Среда Eclipse дает возможность пометить ресурсы такими маркерами, как задачи **Tasks** и закладки **Bookmarks**.

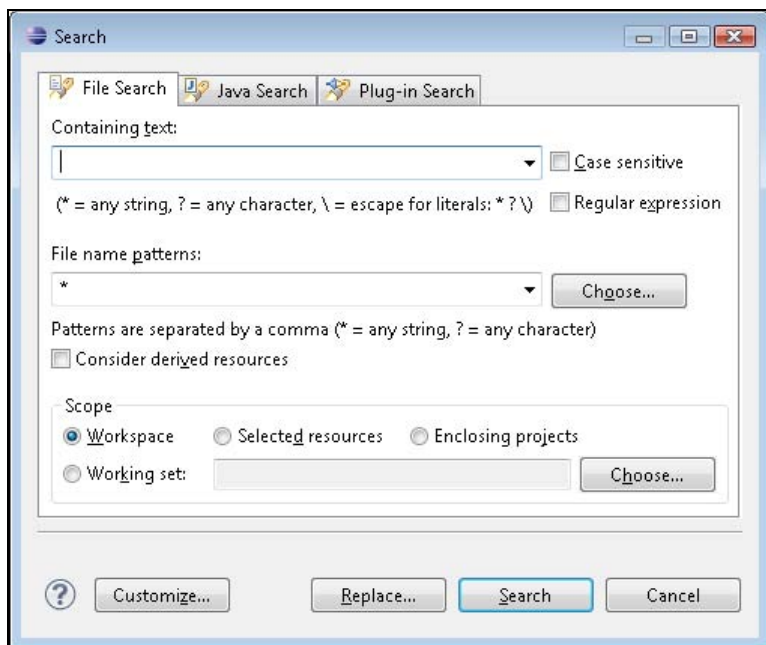


Рис. 1.9. Диалоговое окно настройки поиска ресурсов