

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

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, BIRT. 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 Капалыгина</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 Riена	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



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).

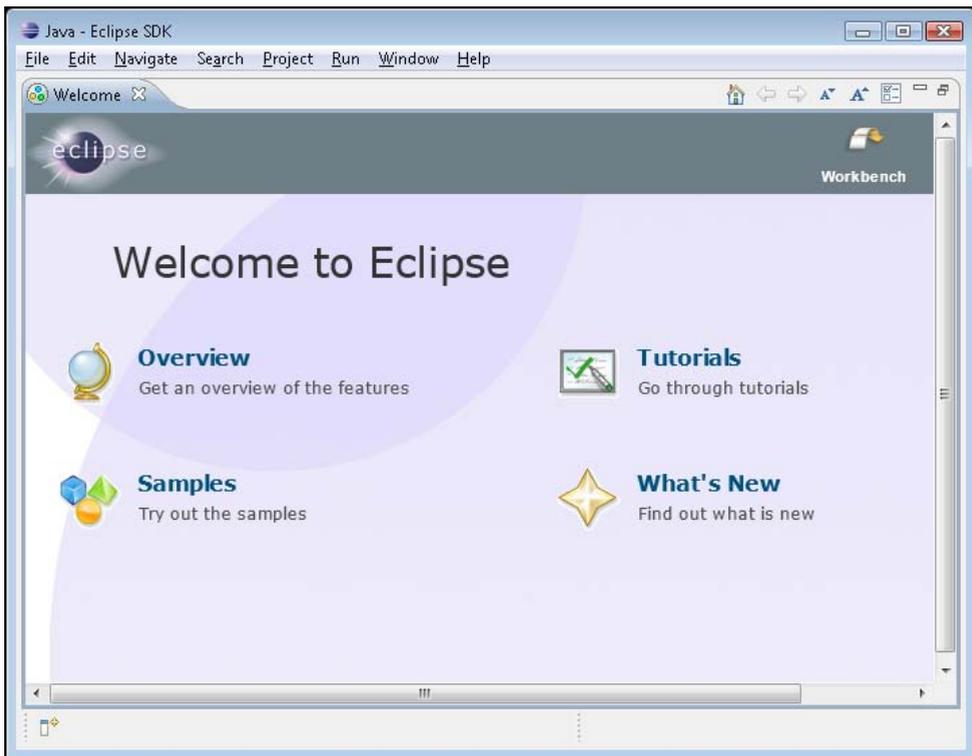


Рис. 1.1. Страница приветствия **Welcome**

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-плагины и RCP-приложения;
- ◆ **Samples** — переход на страницу примеров, содержащую кнопки, которые открывают окна **Cheat Sheet** в Workbench-окне, последовательно проводящие

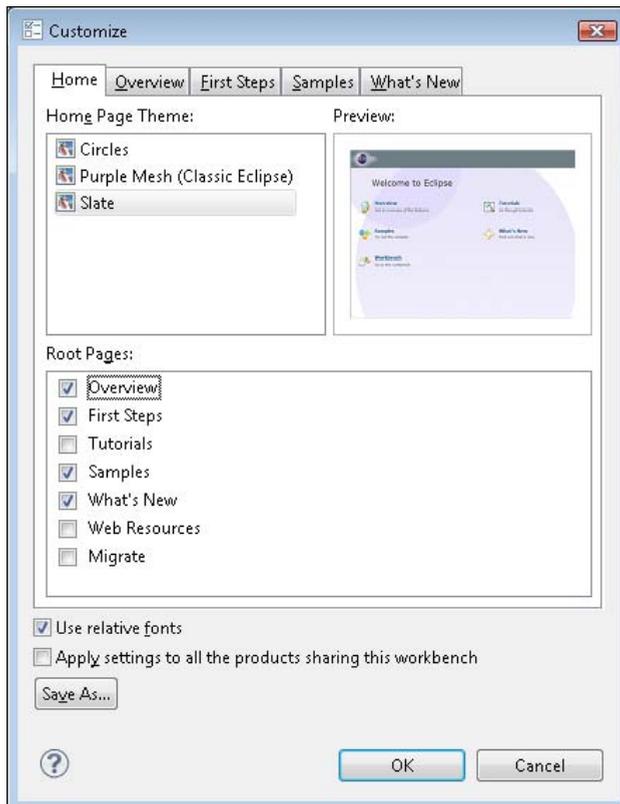


Рис. 1.2. Диалоговое окно изменения страницы **Welcome**

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).

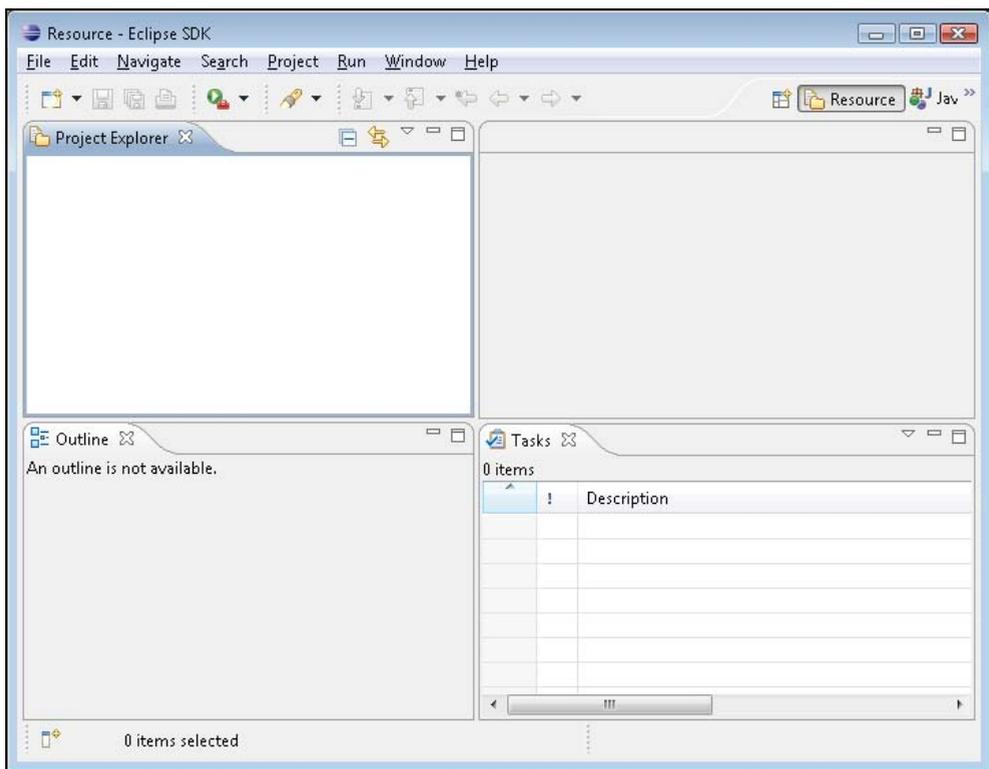


Рис. 1.3. Перспектива **Resource** среды Eclipse SDK

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

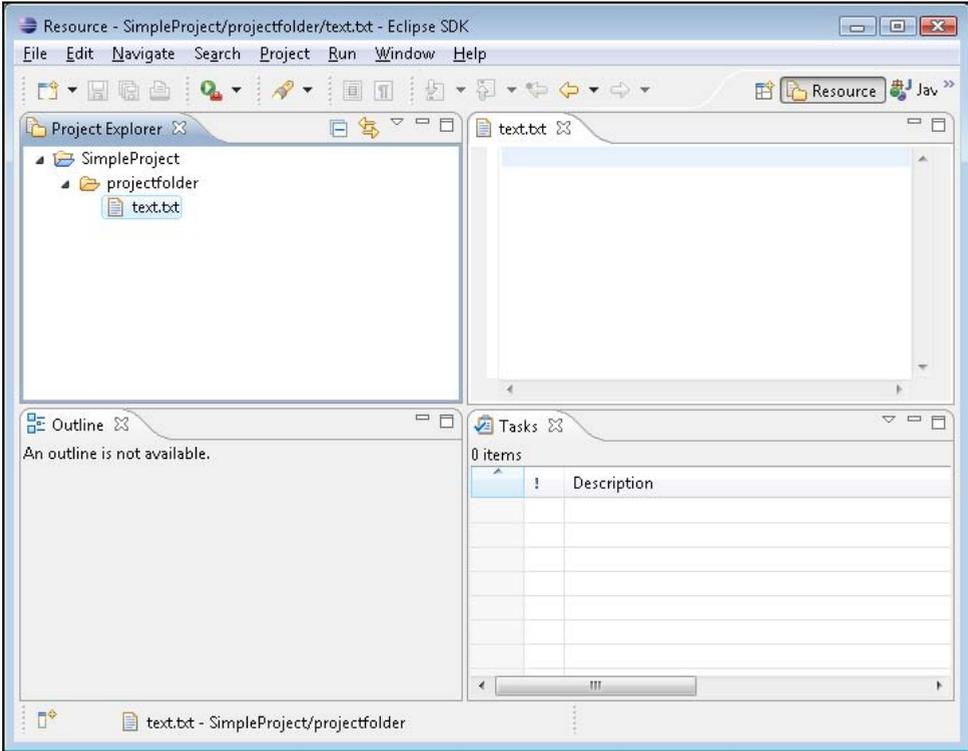


Рис. 1.4. Создание простого Eclipse-проекта с папкой и текстовым файлом

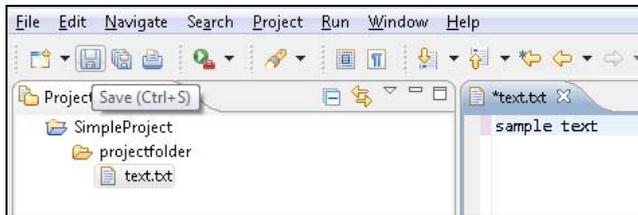


Рис. 1.5. Сохранение изменений текстового файла

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-

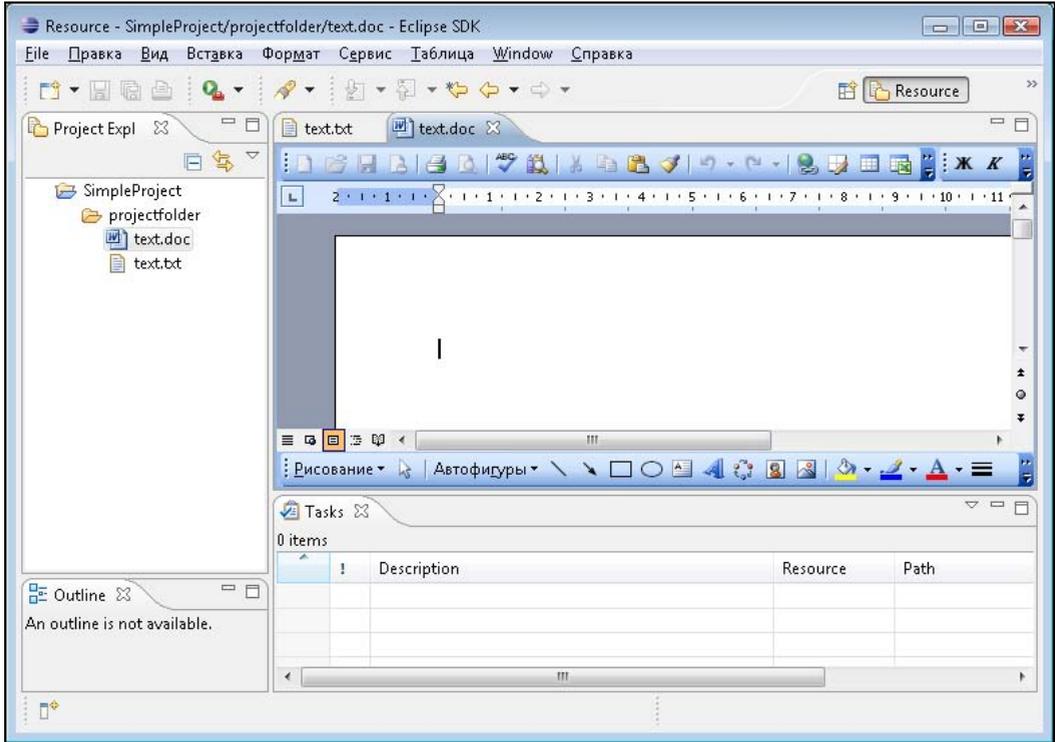


Рис. 1.6. Создание текстового файла с расширением, не совместимым с текстовым редактором Eclipse-платформы

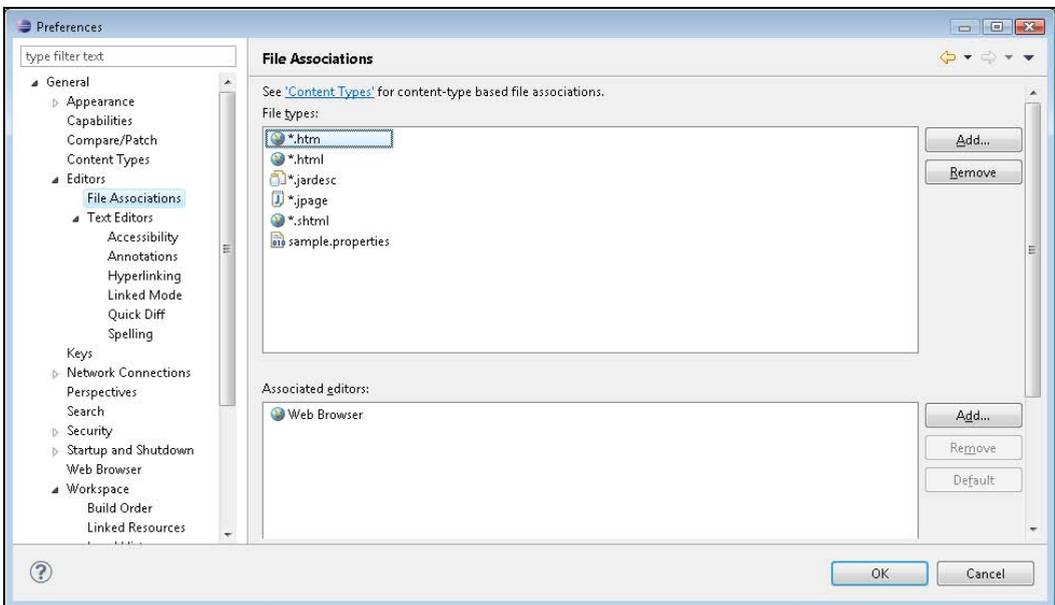


Рис. 1.7. Настройка соответствия файлового расширения определенному редактору

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**.

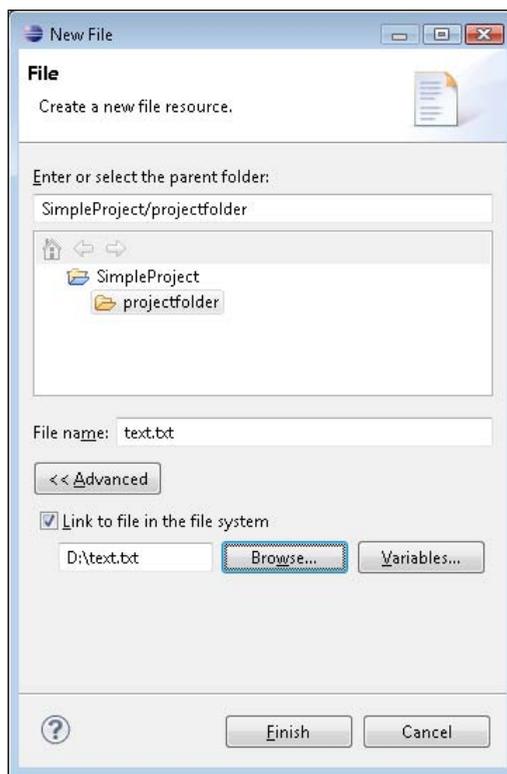


Рис. 1.8. Создание связанного файла

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**.

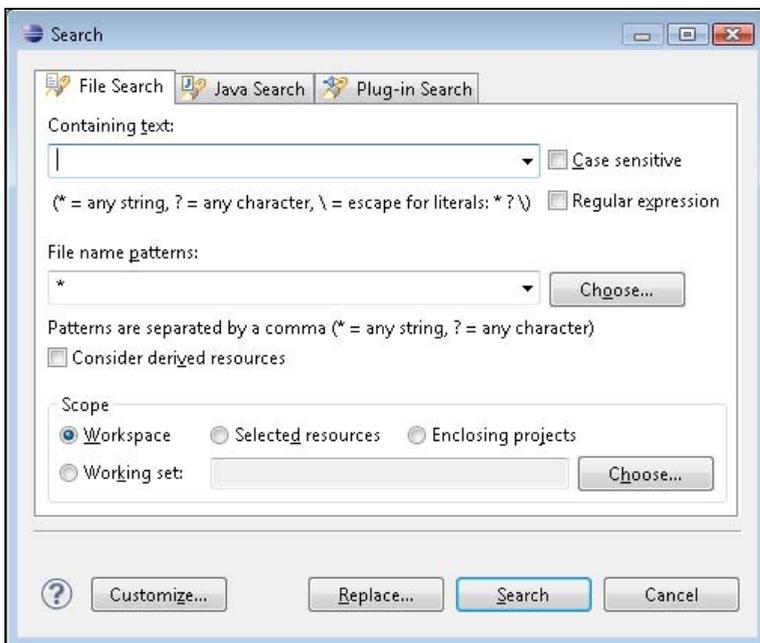


Рис. 1.9. Диалоговое окно настройки поиска ресурсов

Пометить ресурс Task-маркером или Bookmark-маркером можно с помощью команд **Add Task** или **Add Bookmark** меню **Edit** Workbench-окна либо посредством команд **Add Task** или **Add Bookmark** контекстного меню, которое появляется при нажатии правой кнопкой мыши на самом левом крае текстового редактора (рис. 1.10).

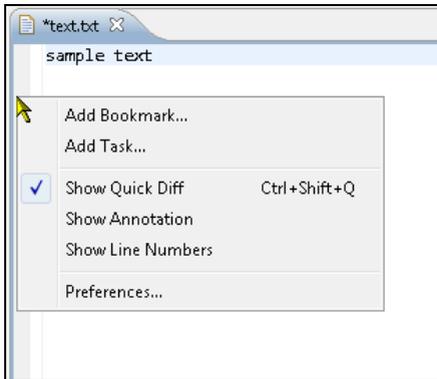


Рис. 1.10. Добавление маркеров с помощью контекстного меню текстового редактора

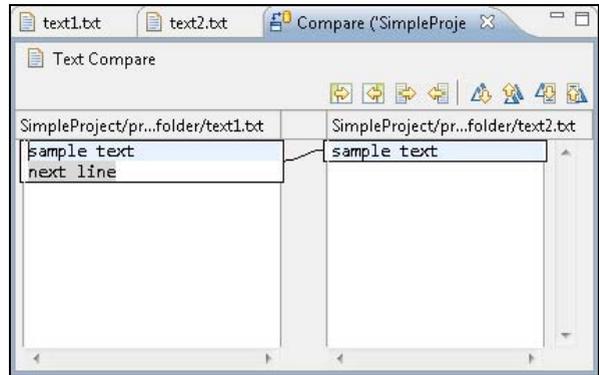


Рис. 1.11. Сравнение содержимого двух текстовых файлов

После создания Task-маркера он появится в окне **Tasks**, а после создания Bookmark-маркера маркер возникнет в окне **Bookmarks**, открыть которое можно с помощью команды **Show View** меню **Window**. Управление созданными маркерами обеспечивают контекстные меню соответствующих представлений.

Eclipse-платформа обеспечивает сравнение ресурсов (проектов, папок и файлов) между собой и сравнение версий редактируемого файла согласно его локальной истории с отображением результатов сравнения в представлении **Compare**.

Для сравнения двух ресурсов между собой необходимо в окне **Project Explorer** щелкнуть левой кнопкой мыши на одном ресурсе, нажать клавишу <Ctrl> и щелкнуть левой кнопкой мыши на другом ресурсе для одновременного выбора сразу двух ресурсов. Затем щелкнуть правой кнопкой мыши на выделенных ресурсах и в контекстном меню последовательно выбрать команды **Compare With | Each Other**. В результате будет открыто окно **Compare** с отображением различий между двумя ресурсами (рис. 1.11).

Для сравнения различных версий редактируемого файла согласно его локальной истории в окне **Project Explorer** щелкнем левой кнопкой мыши на узле файла, нажмем правую кнопку мыши и в контекстном меню последовательно выберем команды **Compare With | Local History**. В появившемся окне **History** два раза щелкнем левой кнопкой мыши на интересующей локальной версии файла — в результате будет открыто окно **Compare** с отображением различий между текущей и предыдущей версиями файла. При этом панель инструментов представления **Compare** обеспечивает функции копирования и навигации.

Локальная история файла организуется средой Eclipse при создании файла и при его модификации. При сохранении отредактированного файла его копия, имеющая идентификатор в виде даты и времени сохранения, также сохраняется, образуя локальную историю файла с возможностью ее просмотра в представлении **History Workbench**-окна. Настраивается локальная история с помощью раздела **General | Workspace | Local History** диалогового окна **Preferences**, открываемого одноименной командой меню **Window**.

Для отображения ресурсов в окне **Project Explorer** можно применять различные фильтры, для создания которых можно воспользоваться кнопкой **Add** раздела **Resource | Resource Filters** диалогового окна, открываемого командой **Properties** контекстного меню узла проекта.

Ограничить набор отображаемых в Eclipse-представлении ресурсов можно также с помощью рабочего набора Working Set, для применения которого к представлению нужно открыть меню кнопкой **View Menu** панели инструментов представления и выбрать команду **Select Working Set** или **Configure Contents**.

Для того чтобы определить используемый рабочий набор Working Set, можно выбрать команду **Customize Perspective** меню **Window** и на вкладке **Command Groups Availability** появившегося диалогового окна отметить флажки **Window**

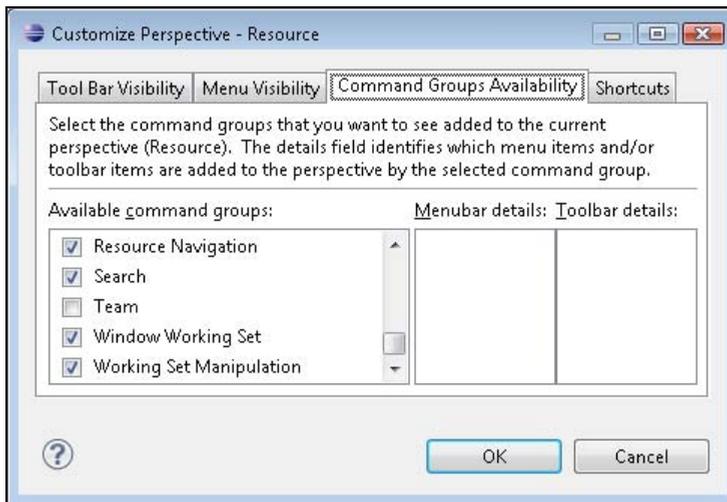


Рис. 1.12. Подключение инструментов работы с набором Working Set



Рис. 1.13. Кнопки управления набором Working Set

Working Set и **Working Set Manipulation** (рис. 1.12). В результате на панели инструментов Workbench-окна появятся кнопки **Modify window working set**, **Add the selected elements to a working set**, **Remove the selected elements from a working set** (рис. 1.13).

Разработка приложений платформы Java SE

Среда разработки Eclipse SDK

Среда Eclipse SDK содержит плагин Java development tools (JDT), расширяющий Eclipse-платформу до интегрированной среды разработки Java IDE, добавляя перспективы **Java**, **Java Browsing**, **Java Type Hierarchy**, **Debug** и набор представлений, редакторов, мастеров и других инструментов для работы с Java-кодом. JDT-плагин служит фундаментом для разработки любых Java-приложений, включая создание Eclipse-плагинов. JDT-плагин содержится во всех остальных Eclipse-продуктах, предназначенных для создания Java-приложений на основе различных платформ, и сам по себе помогает в разработке Java-кода платформы Java SE.

Перспектива **Java** содержит окно редактора и представления **Package Explorer**, **Outline**, **Problems**, **Javadoc**, **Declaration** (табл. 1.3).

Таблица 1.3. Представления перспективы Java

Представление	Описание
Package Explorer	Отображает Java-проект с его структурой, определяемой сборкой проекта, в виде узлов папок и библиотек, Java-пакетов, Java-файлов с их внутренней структурой
Outline	Отображает компилируемую структуру редактируемого в данный момент Java-файла
Problems	Отображает ошибки и предупреждения сборщика проекта
Javadoc	Отображает документацию выбранного в данный момент Java-элемента
Declaration	Отображает исходный код выбранного в данный момент Java-элемента

Перспектива **Java Browsing** содержит окно редактора и представления **Projects**, **Packages**, **Types**, **Members** (табл. 1.4).

Таблица 1.4. Представления перспективы Java Browsing

Представление	Описание
Projects	Отображает Java-проект, его папки и библиотеки без возможности их раскрытия в данном представлении
Packages	Отображает при выборе в окне Projects узла список его Java-пакетов
Types	Отображает при выборе в окне Packages узла список его Java-типов
Members	Отображает при выборе в окне Types узла его содержимое

Перспектива **Java Type Hierarchy** содержит окно редактора и представление **Type Hierarchy**, отображающее иерархию Java-типа.

Перспектива **Debug** содержит окно редактора и представления **Debug, Breakpoints, Variables, Outline, Console, Tasks** (табл. 1.5).

Таблица 1.5. Представления перспективы **Debug**

Представление	Описание
Debug	Обеспечивает управление процессом отладки и запуска Java-кода
Breakpoints	Отображает список контрольных точек отладки Java-кода
Variables	Отображает информацию о переменных выбранного узла окна Debug
Outline	Отображает компилируемую структуру редактируемого в данный момент Java-файла
Console	Отображает системный вывод выполнения Java-кода
Tasks	Отображает список маркеров задач проекта

Пример создания простого Java-приложения

Для создания простого Java-приложения откроем перспективу **Java** среды Eclipse SDK, в меню **File** последовательно выберем команды **New | Other | Java | Java Project** и нажмем кнопку **Next**, введем имя проекта **Hello** и нажмем кнопку **Finish**.

В окне **Package Explorer** нажмем правой кнопкой мыши на узле проекта, в контекстном меню последовательно выберем команды **New | Other | Java | Class**, нажмем кнопку **Next**, в поле **Package** введем имя пакета `hello`, а в поле **Name** — имя класса `hello`, отметим флажок **public static void main(String[] args)** создания точки входа в приложение и нажмем кнопку **Finish** (рис. 1.14).

В результате в окне **Package Explorer** среды Eclipse отобразится иерархия проекта `Hello`, в окне редактора будет открыт файл `Hello.java`, а в **Workspace**-каталоге будет создана папка `Hello` с файлами `.PROJECT` и `.CLASSPATH` и папками `.settings`, `bin` и `src`.

Файл `.PROJECT` определяет папку `Hello` как Eclipse-проект, имеющий тип Java-проект (тег `<natures>`), и содержит команду вызова Eclipse-компилятора исходного Java-кода при инкрементальной сборке проекта (тег `<buildCommand>`).

Файл `.CLASSPATH` содержит определения папок и файлов, участвующих в построении и запуске проекта, и автоматически дополняется новыми определениями при использовании команды **Build Path** контекстного меню окна **Package Explorer** среды Eclipse.

Папка `.settings` хранит установки плагина, папка `src` — исходный Java-код, а открыв папку `bin`, можно обнаружить уже откомпилированный и готовый к запуску Java-код. Произошло это из-за того, что в разделе **General | Workspace** диалогового окна **Preferences**, открываемого одноименной командой в меню **Window**, отмечен