

Советы и трюки для эффективной работы в Eclipse IDE

Введение

Данная статья содержит советы и трюки, которые позволят повысить производительность разработчика при написании встраиваемых приложений в открытой интегрированной среде разработки Eclipse.

В качестве источников при составлении данной подборки выступили:

- Сообщество разработчиков QNX — инженеры, преподаватели;
- Портал Foundry27, являющийся платформой для открытой разработки QNX, в котором имеется форум, посвященный среде Eclipse;
- Форумы проекта Eclipse.org;
- Другие сайты и блоги, содержащие информацию, связанную с Eclipse.

26 советов, описанных ниже, были собраны из всех этих источников и отмечены как интересные и полезные для разработчиков. Они позволяют повысить эффективность работы при использовании Eclipse IDE.

О проекте Eclipse

Eclipse IDE является интегрированной средой разработки с открытым исходным кодом. Впервые она была выпущена под лицензией Common Public License, и сейчас распространяется под лицензией Eclipse Public License.

Eclipse IDE — одна из самых популярных сред разработки в мире. Ее интерфейс доступен на дюжине языков, сообщество Eclipse включает по крайней мере 100 проектов, 1000 спонсоров, 170 членских организаций и миллионы пользователей.

QNX Software Systems является важным пользователем и спонсором, с длинной и разнообразной историей развития проекта. QNX в сотрудничестве с IBM Rational Software разработали Eclipse C Development Tools (CDT) для обеспечения поддержки разработки на языке C/C++. Фирменная среда разработки QNX Momentics также основана на Eclipse.

Совет 1: Отображение помощника Key Assist

Функция Key Assist открывает всплывающее окно, в котором отображаются все горячие клавиши, активные в данном контексте.

1. Нажмите сочетание клавиш *Ctrl+Shift+L* для того, чтобы открыть окно Key Assist.

Полезной особенностью помощника является то, что вы можете открыть окно конфигурации горячих клавиш путем повторного нажатия сочетания *Ctrl+Shift+L*.



Рис. 1. Использование функции Key Assist

Совет 2: Настройка горячих клавиш

Если вы обнаружили, что определенные сочетания клавиш используете наиболее часто, вы можете открыть окно настроек и добавить более удобные и короткие сочетания.

1. Нажмите сочетание клавиш *Ctrl+Shift+L* для того, чтобы открыть окно Key Assist.
2. Нажмите еще раз *Ctrl+Shift+L* для того, чтобы открыть окно настроек.

Настройки в окне Key Binding позволяют пользователю добавить сочетание клавиш для команды и выбрать контекст для ее исполнения. Например, вы можете добавить свое сочетание клавиш, которое будет работать в окне редактирования исходного текста, но не будет работать в окне отладки.



Рис. 2. Добавление сочетаний горячих клавиш для часто используемых функций

Совет 3: Самые полезные горячие клавиши

- Автозавершение слова — **Alt+ /** — Последовательные нажатия перебирают варианты
- Следующий редактор — **Ctrl+F6** — Вызывает диалог с выбором окна редактора
- Следующая перспектива — **Ctrl+F8** — Вызывает диалог с выбором перспективы
- Добавить/Убрать отступ — **Tab/Shift+Tab**
- Перемещение выделенного фрагмента кода вниз/вверх — **Alt+Up/Down**
- Инкрементальный поиск — **Ctrl+J**
- Максимизация окна — **Ctrl+M**
- Поиск закрывающей/открывающей скобки — **Ctrl+Shift+P** — Курсор должен находиться после

интересующей скобки

- Удаление строки — **Ctrl+D**

Совет 4: Рефакторинг

Eclipse IDE обеспечивает некоторые возможности для облегчения рефакторинга исходного кода. В таблице приведены горячие клавиши, часто используемые при рефакторинге.

- Запуск — **Alt+Shift+T** — Запуск меню рефакторинга
- Переименование — **Alt+Shift+R** — переименование выделенного идентификатора во всем проекте, отображение конфликтов и "затенений" имен.
- Константа — **Alt+C** — Экспорт и именование константы.

IDE также обеспечивает предварительный просмотр изменений, таким образом, вы можете просмотреть их в пошаговом режиме.



Рис. 4. Выбор файлов для рефакторинга

Совет 5: Иерархия вызовов

Иерархия вызовов отображает полный список всех функций, которые вызывают выделенный идентификатор. Для использования функции отображения иерархии вызовов:

1. Выделите интересующий идентификатор и нажмите *Ctrl+Alt+H*.

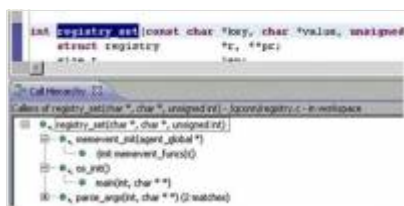


Рис. 5а. Использование функции отображения иерархии вызовов

Например, если вы изменили описание функции, добавив новый параметр, в иерархии вызовов вы сможете увидеть все функции, которые затронет данное изменение.

Отображение всех сущностей, вызываемых функцией.

Иерархия вызовов также позволяет отобразить список всех сущностей, которые вызываются функцией. Для отображения списка:

1. Нажмите кнопку «hierarchy tree» для переключения режима отображения дерева вызовов.



Рис. 5б. Переключение режима отображения дерева вызовов.

Данная функция позволит вам быстро найти места, где код производит вызов системных функций ОС или сторонних библиотек.

Совет 6: Навигация с помощью гиперссылок

Eclipse IDE обеспечивает гиперссылки для определений и прототипов.



Рис. 6. Использование гиперссылок для отображения определений и прототипов.

Для использования данной функции:

1. С нажатой кнопкой *Ctrl* наведите курсор мыши на интересующий идентификатор (имя функции, структуры и т.п.) для отображения гиперссылки.
2. Нажмите на ссылку для перехода к определению идентификатора.

Совет 7: Использование закладок

Закладки могут быть полезны для быстрого перемещения между различными частями программы. Для использования закладок:

1. Перейдите в необходимое место программы.
2. Нажмите правой кнопкой мыши на сером бордюре слева.
3. Выберите пункт *Add Bookmark*.

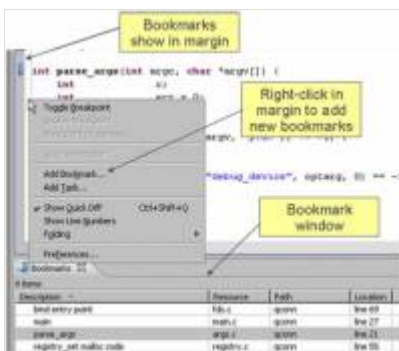


Рис. 7. Добавление и просмотр закладок.

Для просмотра списка закладок:

1. Перейдите в меню: *Select Window > Show View > Other ... > General > Bookmarks.*

Совет 8: Запрос аргументов при запуске программы на исполнение

Если Вы запускаете приложение в консольном режиме, и оно требует ввод аргументов, у Вас может быть необходимость вводить разные значения при каждом запуске. Для того, чтобы не плодить множество отладочных конфигураций и не редактировать их постоянно, можно воспользоваться возможностью Eclipse IDE, позволяющей запрашивать аргументы при запуске.

Eclipse позволяет добавить в строке аргументов запуска специальную переменную, после чего, при каждом запуске конфигурации, IDE будет выводить окно с запросом аргументов.

Для установки аргументов запуска:

1. В диалоговом окне *Launch Configuration* выберите вкладку *Arguments tab* (см. Рис. 8а).
2. Когда Вы увидите запрос ввода аргументов запуска приложения, напечатайте в строке ввода в последовательности со всеми остальными параметрами, которые Вы хотите передать программе при выполнении, строку *`\${string_prompt}*.



Рис. 8а. Редактирование аргументов запуска приложения.

Когда Вы запустите программу, значение, которое было задано по умолчанию может быть принято или модифицировано.

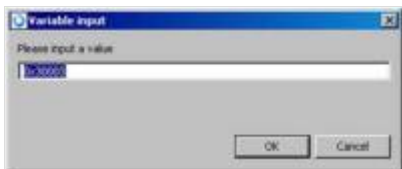


Рис. 8б. Ввод аргумента при запуске приложения.

Следует запомнить, что необходимо соблюдать последовательность ввода аргументов.

Совет 9: Использование шаблонов ввода

Шаблоны ввода кода могут быть полезны для таких блоков, как обработчики исключений, циклы и т.п.

IDE по умолчанию имеет много предустановленных шаблонов, но Вы можете добавить собственные.

Также, Вы можете включить выделенные в редакторе строки, например, временные переменные, в какую-либо специальную конструкцию.

Для использования шаблонов:

1. Введите первые несколько букв шаблона.
2. Нажмите комбинацию *Ctrl+Space* и выберите шаблон, который Вы хотите применить.
3. После того, как шаблон будет развернут, используйте кнопку *Tab* для последовательного редактирования полей шаблона.



Рис. 9. Применение шаблонов.

Совет 10: Отображение ассемблерного кода

Многие разработчики не знают, что они могут использовать Eclipse для отображения ассемблерного кода (листинга) программы. Помимо обычного редактирования исходного кода, может быть полезным изучение ассемблерного кода, особенно при использовании низкоуровневых методик отладки.

Также, иногда бывает необходимо изучить генерируемый компилятором ассемблерный код с целью более детально разобраться в программе и для ее оптимизации.

Если Вы имеете скомпилированные объектные или исполняемые файлы в составе проекта, просто откройте их для просмотра ассемблерного кода (листинга).

Данный совет применим для среды разработки Momentics под платформу QNX, при использовании других компиляторов, файлы листинга, как правило, генерируются в виде отдельных файлов (прим. переводчика).



Рис. 10. Просмотр ассемблерного кода (листинга) программы.

Совет 11: Отделение окон (представлений)

Eclipse IDE позволяет отделять отдельные окна (представления) от главного окна. Эта функция особенно полезна, если Вы используете при работе несколько мониторов.

Для того, чтобы отделить окно, просто нажмите правой кнопкой мыши на его заголовке и выберите пункт *Detach*. Для обратного присоединения, выберите в контекстном меню *Attach* или перетащите его мышкой в необходимую область главного окна.

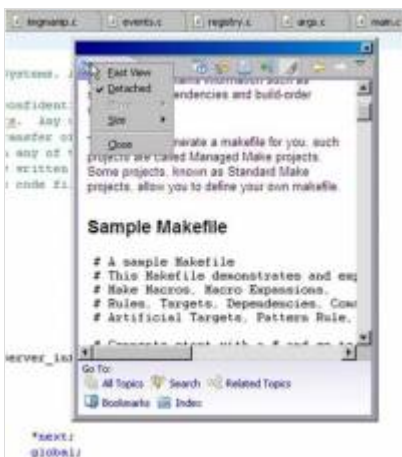


Рис. 11. Отделение и присоединение окон IDE.

Совет 12: Групповой запуск

Eclipse IDE может быть сконфигурирована для одновременного запуска нескольких процессов. Такая возможность может быть полезна при разработке и отладке нескольких взаимодействующих процессов, например:

- Сервер и его клиент;
- Драйвер и вызывающее его приложение;
- НМІ и вспомогательные процессы.



Рис. 12. Использование группового запуска.

Вы можете комбинировать различные конфигурации запуска, например, локальные скрипты или запуск удаленного процесса. Если они находятся в группе, отладчик начнет запуск каждого члена, который был приостановлен.

Совет 13: Переменные пути

Переменные пути в Eclipse IDE схожи с символическими ссылками в операционных системах UNIX/Linux, они ссылаются на определенную директорию, находящуюся на Вашем компьютере. Переменные пути не ограничены одним проектом, Вы можете создавать переменные, которые могут быть использованы в нескольких проектах.

Эта возможность может быть полезна в случае, если Вам приходится работать на нескольких рабочих местах с одним и тем же проектом.

Для создания новой переменной со ссылкой на путь в файловой системе:

1. Запустите диалоговое окно *New Folder*.
2. Введите ссылку на директорию в файловой системе и нажмите кнопку *Variables*.

Для просмотра переменных:

1. Выберите *Windows > Preferences > General Workspace > Linked Resources*.

Теперь Вы можете увидеть все переменные, которые уже были сконфигурированы или добавить новые.

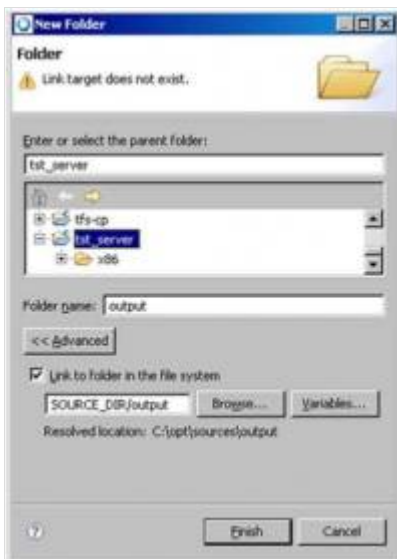


Рис. 13. Конфигурация переменной пути

Совет 14: Пользовательские действия точек останова

Пользовательские действия точек останова являются удобным помощником для отладки кода, который содержит трудно воспроизводимую ошибку.

Отладка часто включает запуск определенного фрагмента кода до тех пор, пока конкретный набор условий не приведет к сбою. Этот способ отладки может означать повторение одного и того же действия десятки, а то и сотни раз без проявления ошибки.

Пользовательские действия точек останова позволяют создать пользовательские уведомления и другие

действия в Вашем отладчике, что позволит оставить код работать и сосредоточиться на других задачах, пока не произойдет ошибка. Вы можете настроить проигрывание звука или WAV-файла, который предупредит Вас о том, когда проблемный код достигнет точки останова.

Действие точки останова может, также, сделать запись в журнале событий или просто напечатать какой-либо текст (например, значение указанной переменной), а затем продолжить выполнение программы. Последняя возможность предлагает Вам механизм эффективной вставки printf в код без перекомпиляции и загрузки кода.



Рис. 14. Настройка свойств точки останова.

Вы можете установить несколько разных опций для каждой точки останова. Для доступа к этим функциям:

1. Нажмите правой кнопкой мыши на точке останова и выберите пункт *Breakpoint Properties*. Eclipse IDE откроет окно, в котором Вы сможете добавить или изменить действия для данной точки останова.

Совет 15: Управление файлами на целевой системе

Данная функция специфична для версии Eclipse от QNX — QNX Momentics Tool Suite. Это очень полезная функция, позволяющая сэкономить время.

Версия Eclipse от QNX позволяет управлять файлами непосредственно на целевой системе. Часто, разработчикам необходимо закачивать файлы на удаленную систему или с нее, а иногда и редактировать файлы непосредственно на ней.

Если Вы используете QNX Momentics Tool Suite, среда разработки имеет файловый менеджер для работы с целевой системой, который позволяет просматривать список директорий, копировать файлы на целевую систему и с нее, а также выполнять другие действия (включая удаление и запуск исполняемых файлов) так же, как если бы Вы работали с локальной файловой системой. С помощью данного файлового менеджера Вы можете производить все действия без использования telnet, vi или ftp.

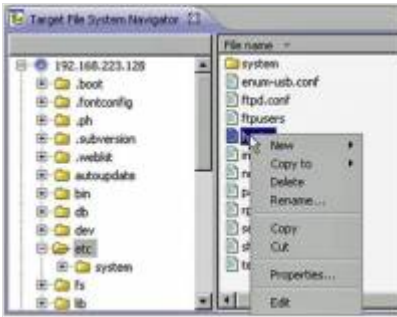


Рис. 15. Работа с файлами на целевой платформе

Совет 16: Автоматическое добавление заголовочного файла

Eclipse IDE поддерживает автоматическое включение необходимых заголовочных файлов в исходный код. Для того, чтобы узнать, в каком заголовочном файле описан Ваш идентификатор, выделите имя функции и нажмите комбинацию *Ctrl+Shift+N* или нажмите правую кнопку мыши и выберите пункт *Source > Add Include*.

Среда разработки отредактирует Ваш файл с исходным кодом, добавив директиву *include* для необходимого заголовочного файла. Например, если Вам необходима функция *fopen()*, и файл с исходным текстом не содержит ее описание, Eclipse автоматически вставит его для Вас.



Рис. 16. Автоматическое добавление заголовочного файла с описанием функции.

Совет 17: Блочное редактирование

Eclipse IDE поддерживает редактирование блоков текста. Все, что нужно для этого — выделить необходимый блок текста и выполнить одно из следующих действий:

1. Используйте *Tab* или *Backtab (Shift+Tab)* для перемещения текста вправо или влево.
2. Используйте кнопку *Alt* совместно с клавишами-стрелками для перемещения блока текста вверх или вниз.
3. Автоматическое закомментирование блока текста с помощью сочетания *Ctrl+/**. Это добавляет ко всему блоку текста комментарий в стиле C++ `«/*»`.
4. Использование сочетания *Ctrl+Shift+/** обрамляет текстовый блок символами комментария `«/* ... */»`.
5. Переформатирование блока текста в соответствии с выбранным стилем осуществляется при помощи сочетания *Ctrl+Shift+F*.



Рис. 17. Использование блочного редактирования.

Совет 18: Переформатирование исходного кода

Eclipse IDE включает настраиваемый форматировщик исходного кода с несколькими примерами стилей. Для использования этих стилей:

1. Выберите *Windows > Preferences > C/C++ > Code Formatter*.
2. Выберите один из предустановленных стилей: K&R, BSD/Allman, GNU, Whitesmiths или пользовательский.

Весь новый код будет отформатирован в соответствии с выбранным стилем. Для применения стиля к существующему коду:

1. Выделите текст, который Вы хотите переформатировать.
2. Нажмите *Ctrl+Shift+F*.

Данная функция дает гибкий контроль за отступами, положением скобок, новыми строками и т.д.

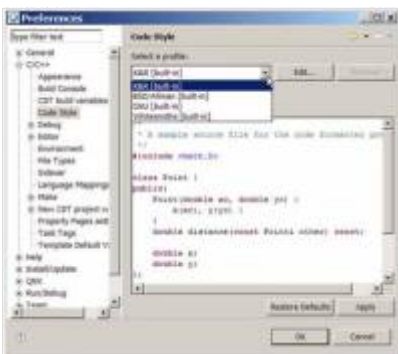


Рис. 18. Настройка стиля кода.

Совет 19: Автоматическое дополнение имен функций

Одной из самых популярных функций, ускоряющих разработку, является автоматическое дополнение имен вводимых функций. Для использования данной возможности, просто введите первые буквы имени функции и нажмите *Ctrl+Space* для отображения списка подходящих имен.

По мере дальнейшего ввода имени, список функций сокращается, и в любой момент Вы можете:

1. Выбрать необходимую функцию из списка.
2. Нажать *Enter* для добавления функции в текст.

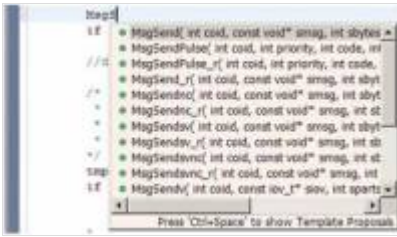


Рис. 19а. IDE отображает список подходящих функций.

После того, как Вы выбрали нужную функцию, IDE выводит подсказку для заполнения ее параметров.



Рис. 19б. Запрос параметров функции.

Совет 20: Автоматическое дополнение структур

Данная функция Eclipse IDE схожа с функцией дополнения имен функций. Введите первые буквы имени структуры, нажмите *Ctrl+Space* и IDE предложит список подходящих структур. Также, эта функция отображает имена элементов и типов структуры.



Рис. 20а. Дополнение структуры.

Вы можете настроить Eclipse IDE для автоматического дополнения структур после определенной задержки, а также следующих указанных клавиш.

Для настройки задержки:

1. Выберите *Window > Preferences > C/C++ > Editor > Content Assist*.
2. Введите время задержки в миллисекундах.

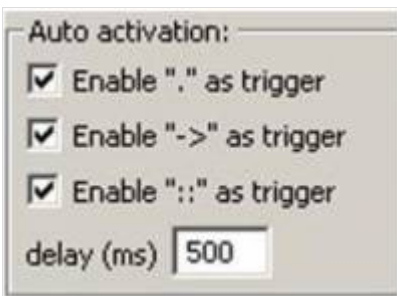


Рис. 20б. Настройка автоматического дополнения структур.

Совет 21: Прототипы, Определения и Реализации

Eclipse IDE содержит функционал, который упрощают работу с функциями:

1. Подсветка функции и нажатие кнопки *F3* позволяет перейти к ее определению.
2. Наведение курсора мыши на функцию и нажатие кнопки *F2* открывает мини-редактор, в

котором отображается ее определение в режиме «только чтение».

Таким образом, нет необходимости поиска файла, содержащего описание функции.



Рис. 21. Отображение информации о функции.

Совет 22: Раскрытие `#define`

Данная функция позволяет понять актуальное значение, которое подставляет компилятор при использовании конкретного определения `#define`.

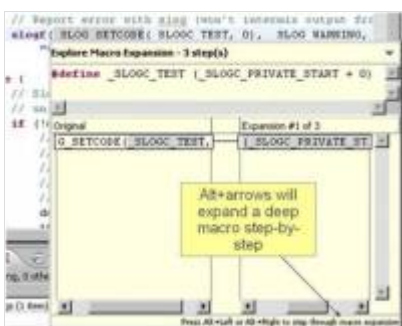


Рис. 22. Раскрытие `#define`.

Для использования:

1. Выберите идентификатор, установленный с помощью `#define`.
2. Нажмите `F2`.

IDE отображает определения, и позволяет просматривать их шаг за шагом, путем расширения. Этот метод также оптимальный способ отладки макросов. Он показывает, что на самом деле будет в коде при компиляции.

Совет 23: Undo и Redo

Eclipse IDE поддерживает отмену и повтор действий при редактировании. Для отмены внесенных изменений или повторного внесения, до сохранения файла, нажмите `Ctrl+Z`, чтобы отменить, или `Ctrl+Y`, чтобы повторно внести.

По умолчанию, IDE сохраняет последние 200 изменений, которые Вы внесли в файл. Вы можете изменить это значение в настройках среды разработки.

Eclipse IDE размещает на полях отметки напротив Ваших изменений, до сохранения файла. Чтобы отобразить оригинальный текст, который был на момент последнего сохранения файла, просто наведите курсор мышки на соответствующую отметку на полях.



Рис. 23а. Отображение отметок об изменениях на полях.



Рис. 23б. Отображение оригинального текста.

Совет 24. Локальная история

Eclipse IDE сохраняет историю всех изменений в файле с момента его первого сохранения. Локальная история является записью этих изменений, которые Вы можете просмотреть, для того, чтобы увидеть все конкретные модификации, которые были применены к файлу каждый раз, как он был сохранен.



Рис. 24а. Отображение локальной истории.

Для доступа к локальной истории:

1. Откройте *Project Navigator*.
2. Выберите *Team > Show Local History*.

Сравнение файлов

Хотя проект не обязательно должен быть связан с системами управления кодом, локальная история находится под пунктом меню групповой работы, потому что она сродни системам управления версиями.

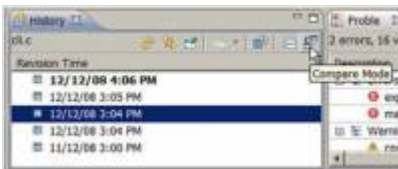


Рис. 24б. Выбор режима сравнения.

Локальная история — это как мини-система управления версиями и мини-система управления исходным кодом, поэтому, даже если Вы не используете системы контроля исходного кода, Вы всегда можете сравнить любые две сохраненные версии файла.

Если Вы внесли несколько изменений в файл, Вы можете использовать отображение side-by-side для визуального сравнения версий, затем выберите те изменения, которые хотите сохранить, и те изменения, которые хотите отменить (См. Рис. 24б и 24в).



Рис. 24в. Сравнение версий файла.

Настройка локальной истории

Сохранение локальной истории может потребовать большого объема дискового пространства. Вы можете настроить максимальный объем дискового пространства, который может быть использован данной функцией.

1. Выберите *Windows > Preferences*.
2. Настройте пределы использования дискового пространства.

Совет 25: Быстрый доступ

Всплывающее окно быстрого доступа является простым способом получить доступ к элементам Eclipse IDE, о которых Вы не знаете, где они находятся. Для использования этой функции:

1. Нажмите *Ctrl+3* для отображения всплывающего окна.
2. Введите термин, который Вы хотите найти.



Рис. 25. Использование быстрого доступа.

Совет 26: Сворачивание кода

Eclipse IDE может сворачивать в одну строчку функции, структуры и другие сущности, делая код более легким для чтения. Для настройки сворачивания:

1. Выберите *Windows > Preferences > C/C++ > Editor > Folding*.

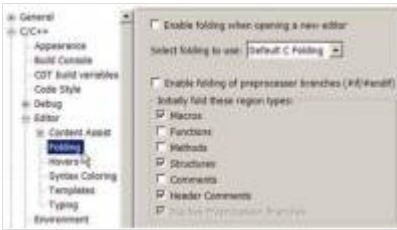


Рис. 26а. Сворачивание кода.

Следующие команды ускоряют работу со сворачиванием кода:

1. Наведение курсора мыши помогает увидеть содержимое скрытого блока кода.
2. Сочетание *Ctrl+/* (на цифровой клавиатуре, для ноутбука может быть необходима комбинация *Ctrl+Fn+.*) позволяет переключаться между режимами сворачивания (включено/выключено).
3. Сочетание *Shift+Ctrl+/* сворачивает все раскрытые блоки.

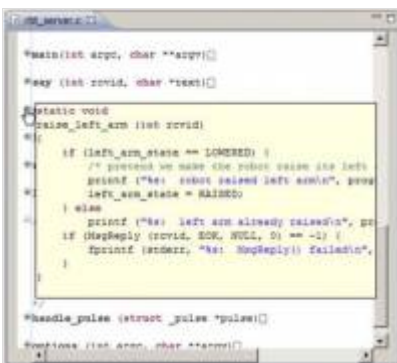


Рис. 26б. Отображение содержимого свернутого кода.

Получение Eclipse IDE

Вы можете найти Eclipse IDE на сайте проекта: www.eclipse.org.

Об авторах и переводчике

Скачать оригинал статьи на английском языке можно по [этому адресу](#).

Авторы статьи: Garry Bleasdale, Andy Gryc (QNX Software Systems).