
Разработка объектно-ориентированных систем программирования интегрированных в среду Eclipse

1. Расширение среды Eclipse

Владимир Юрьевич Романов,
Московский Государственный Университет им. М.В.Ломоносова
Факультет Вычислительной Математики и Кибернетики
vromanov@cs.msu.su,
vladimir.romanov@gmail.com

Структура курса

1. Расширение среды Eclipse
2. Разработка распознавателей ЯП
3. Использование Graphical Editing Framework
4. Использование метамодели языка UML 2.0
5. Использование промежуточного представления компилятора
6. Генерация кода виртуальной машины Java

Расширение среды Eclipse

- **Архитектура среды разработки Eclipse фирмы IBM.**
- **Технология расширения среды Eclipse.**
- **Расширение существующих и создание новых видов, редакторов и перспектив в среде Eclipse.**
- **Особенности интеграции систем моделирования в среду Eclipse.**

Разработка распознавателей языков программирования (1)

- Разработка распознавателей объектно-ориентированных языков программирования.
- Спецификация грамматик языков программирования. Разрешение конфликтов при спецификации грамматик.

Разработка распознавателей языков программирования (2)

- Описание атрибутивных грамматик с помощью генератора компиляторов ANTLR.
- Использование атрибутивных грамматик для описания синтаксиса языка Java.

Использование Graphical Editing Framework

- Использование Graphical Editing Framework (GEF) – базового инструмента для построения графического интерфейса систем моделирования.
- Основные элементы, используемые для построения графического интерфейса.
- Инструментарий для разработки графических редакторов.

Использование метамодели языка UML 2.0

- Использование реализованной фирмой IBM метамодели языка UML 2.0.
- Взаимосвязь реализации UML и EMF.
- Средства языка UML для визуализации структуры и поведения программных систем.
- Средства графической нотации языка UML для визуализации UML-модели.

Использование промежуточного представления компилятора

- Построение классов промежуточного представления компилятора средствами построения моделей среды Eclipse.
- Интеграция промежуточного представления компилятора в среду Eclipse.
- Компиляция программы в промежуточное представление компилятора.

Генерация кода виртуальной машины Java

- Система команд и структура программного код виртуальной машины Java.
- Генерация кода виртуальной машины Java с помощью пакета Byte Code Engineering Library.
- Интеграция сгенерированного кода виртуальной машины Java в среду Eclipse.

Требования к платформе Eclipse (2)

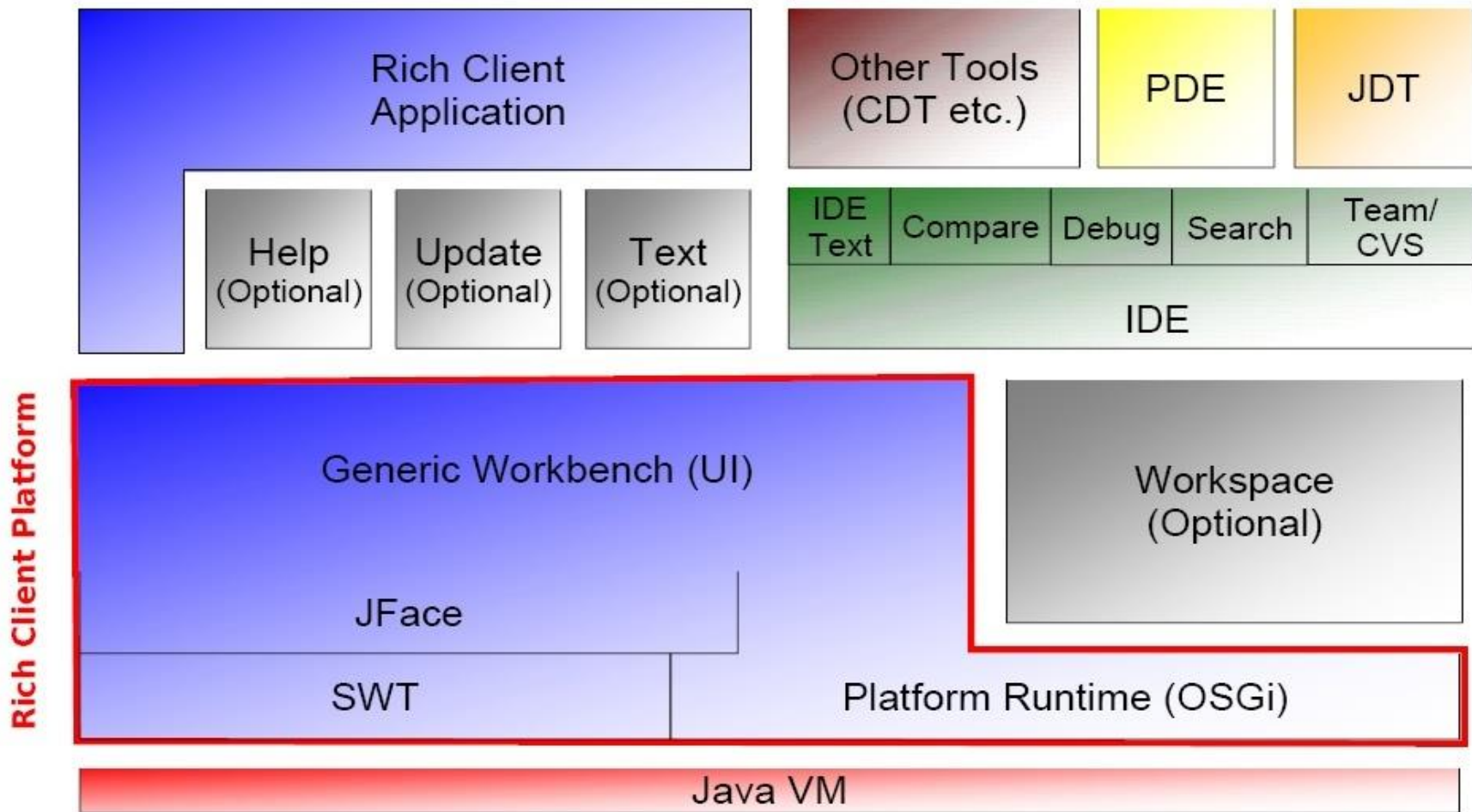
- Облегчить бесшовную интеграцию документов различного типа и и разработанных разными провайдерами
- Выполняться на различных операционных системах, включая Windows, Linux, Mac OS X, Solaris AIX, и HP-UX.

Eclipse Software Development Kit

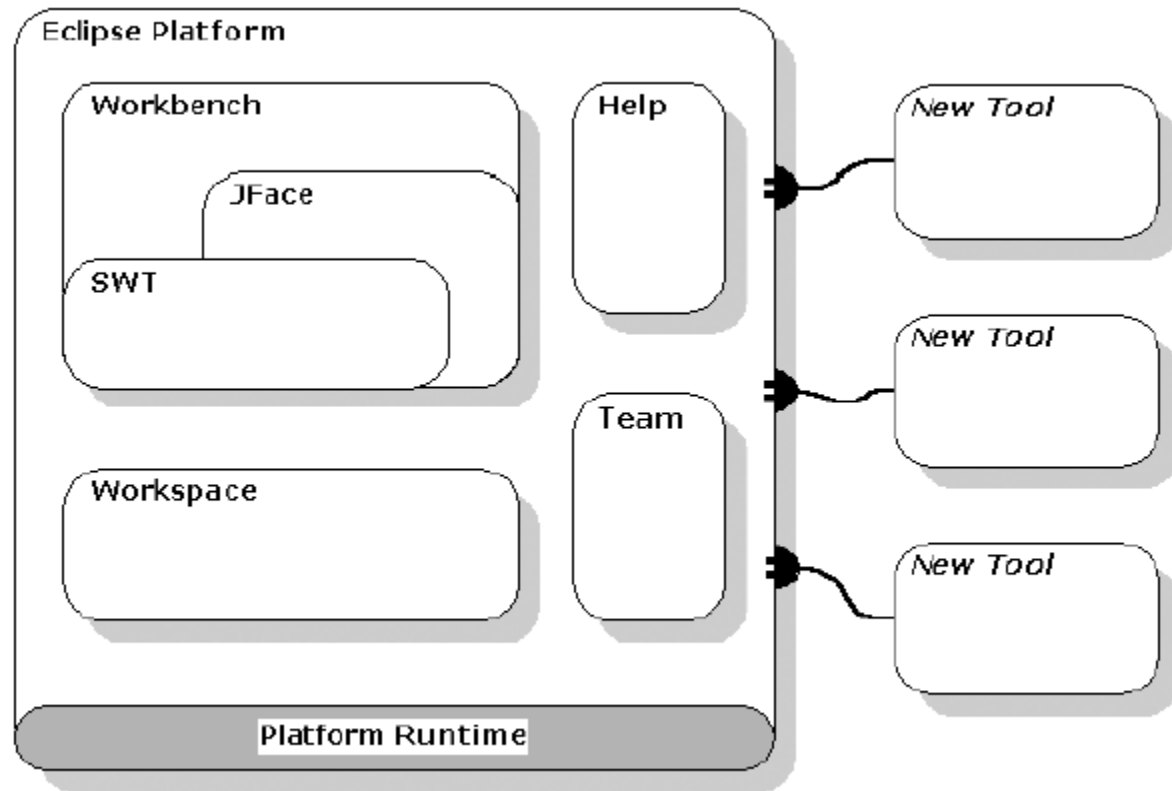
- **Eclipse Platform**
Небольшое ядро (core) и набор подключаемых модулей
- **Java Development Tools (JDT)**
инструментарий для программирования на Java
- **Plug-in Development Environment (PDE)**
среда разработки подключаемых модулей

Архитектура платформы Eclipse

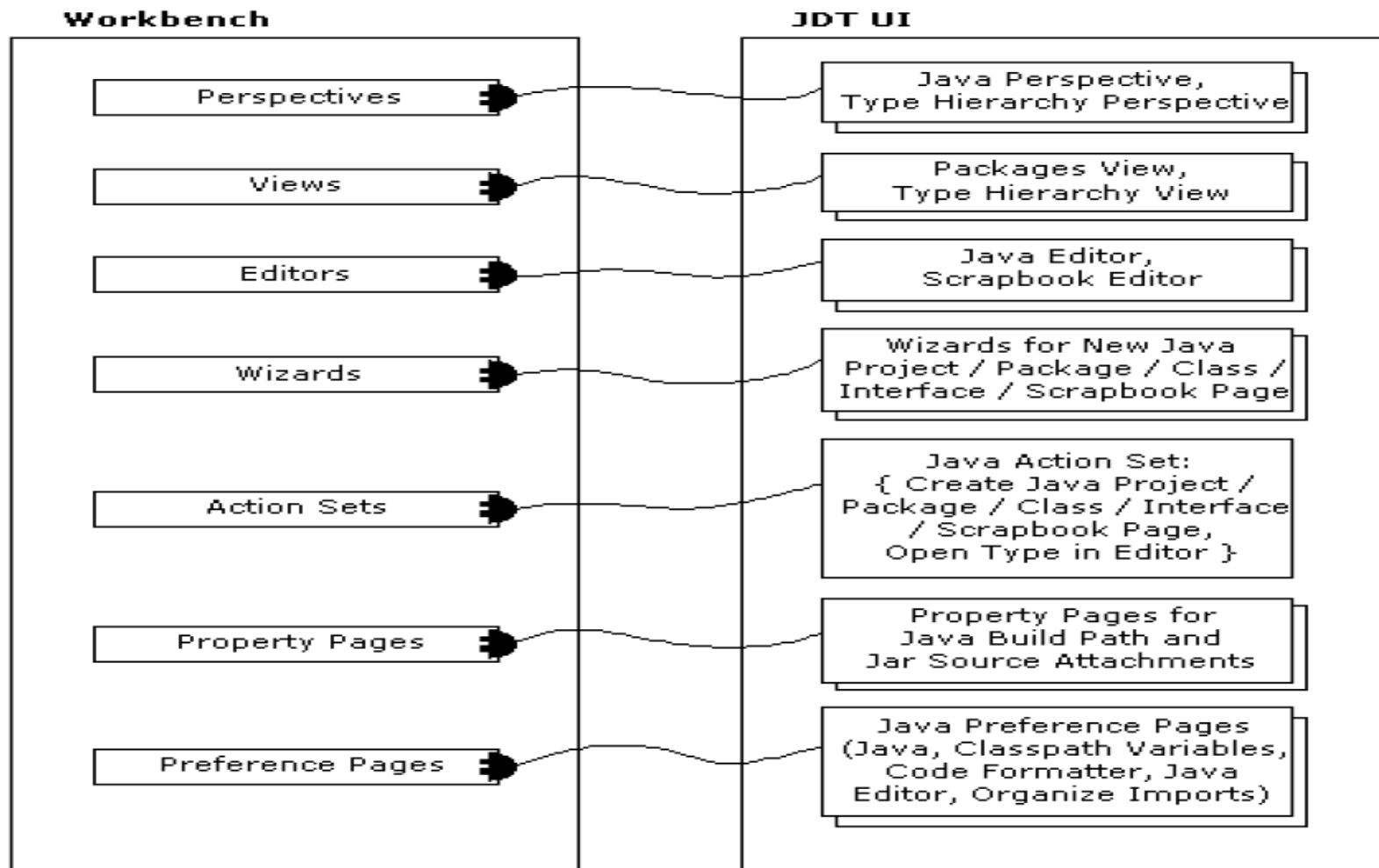
Архитектура платформы Eclipse



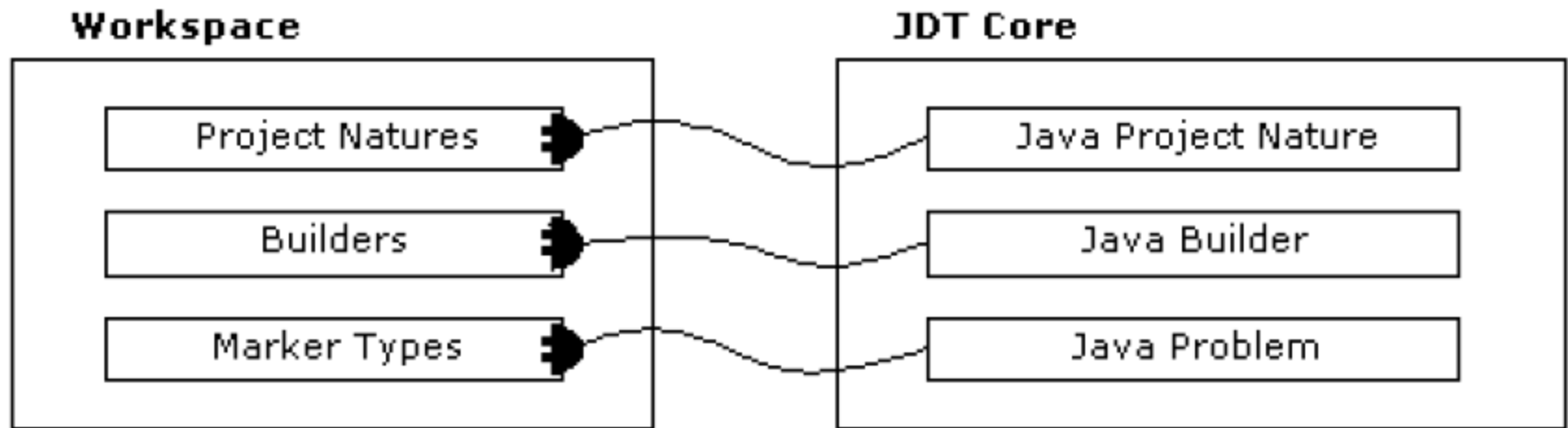
Архитектура платформы Eclipse



Архитектура JDT (1)



Архитектура JDT (2)



Архитектура ядра Eclipse.

Подключаемые модули

- Подключаемый модуль (plug-in) – наименьшая единица функциональности платформы Eclipse, которая может быть разработана и поставлена отдельно
- Вся функциональность Eclipse реализована как подключаемые модули, за исключением небольшого ядра платформы (Platform Runtime)

Архитектура ядра Eclipse.

Структура подключаемых модулей

- **Модуль – расположен в отдельной директории, содержащей:**
 - Java Archive (файл с расширением *.jar), содержащим кодом для JVM и тексты на языке Java
 - **Файлы манифеста:**
 - plugin.xml – содержит описание точек расширения и расширения для расширений из других модулей
 - manifest.mf – содержит зависимости периода выполнения
 - **Файлы – ресурсы**
 - изображения
 - каталоги сообщений
 - HTML – страницы для выдачи помощи

- **Группа связанных модулей называется возможностью (feature) и располагаются в отдельной директории**

Архитектура ядра Eclipse.

Точки расширения модулей

- Файл манифест модуля описывает взаимосвязь модуля с другими модулями
 - Объявление неограниченного количества *точек расширения*
 - Объявление неограниченного количества *расширений*, использующих *точки расширения* из других модулей
 - Пример – минимальный файл-манифест:

```
<?xml version="1.0" encoding="UTF-8" ?>
<plugin
  name="JUnit Testing Framework"
  id="org.junit"
  version="3.7"
  provider-name="Eclipse.org">
  <runtime>
    <library name="junit.jar">
      <export name="*" />
    </library>
  </runtime>
</plugin>
```

Архитектура ядра Eclipse.

Использование точек расширения

- Информация в файлах-манифестах о точках расширения и расширениях модулей используется при инициализации *Eclipse Runtime* для создания регистра модулей
- Информация о модулях известна без их загрузки и активации
- Модули загружаются только тогда, когда выполнение их кода действительно необходимо (*ленивая загрузка*)

Разработка
простейшей точки расширения.

Без использования инструмента
Plugin Development Environment

Разработка простейшей точки расширения. Атрибуты подключаемого модуля

■ Назначение подключаемого модуля

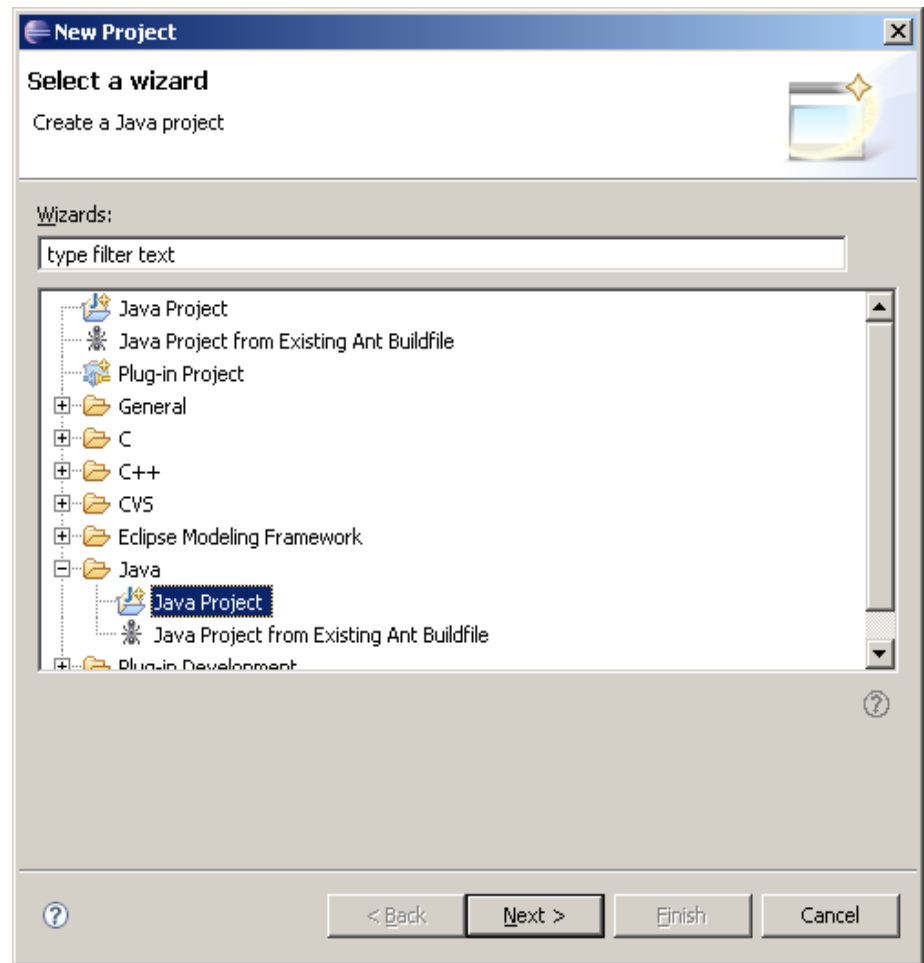
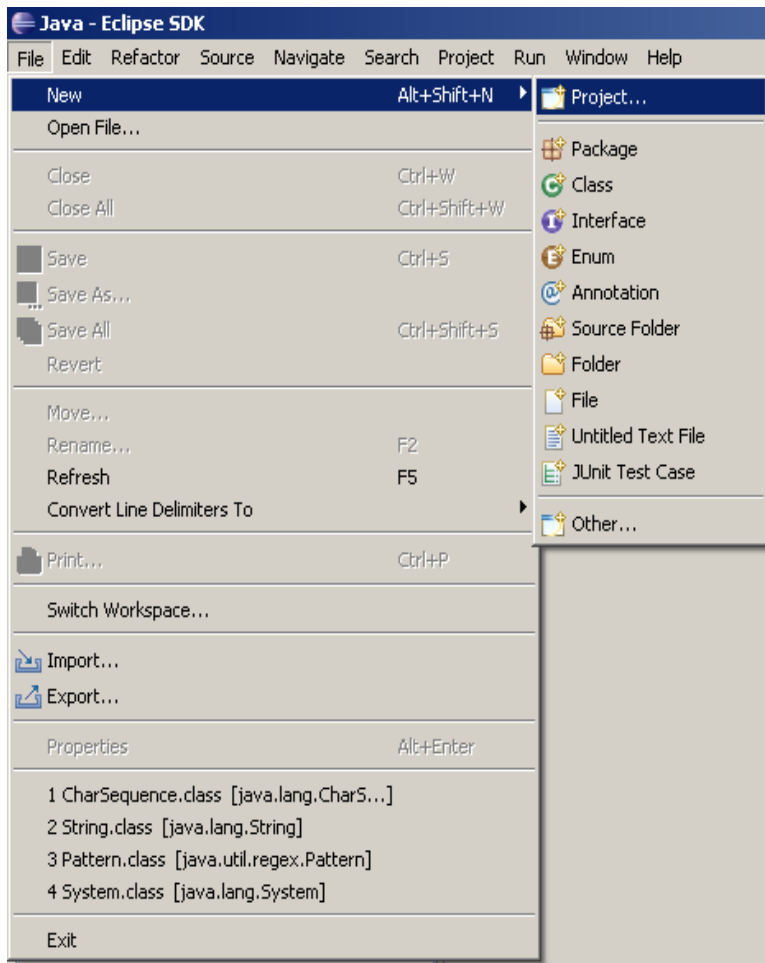
- Модуль представлен кнопкой в панели инструментов *workbench*
- При нажатии на кнопку выдает окно с сообщением *Hello Eclipse*

■ Атрибуты подключаемого модуля

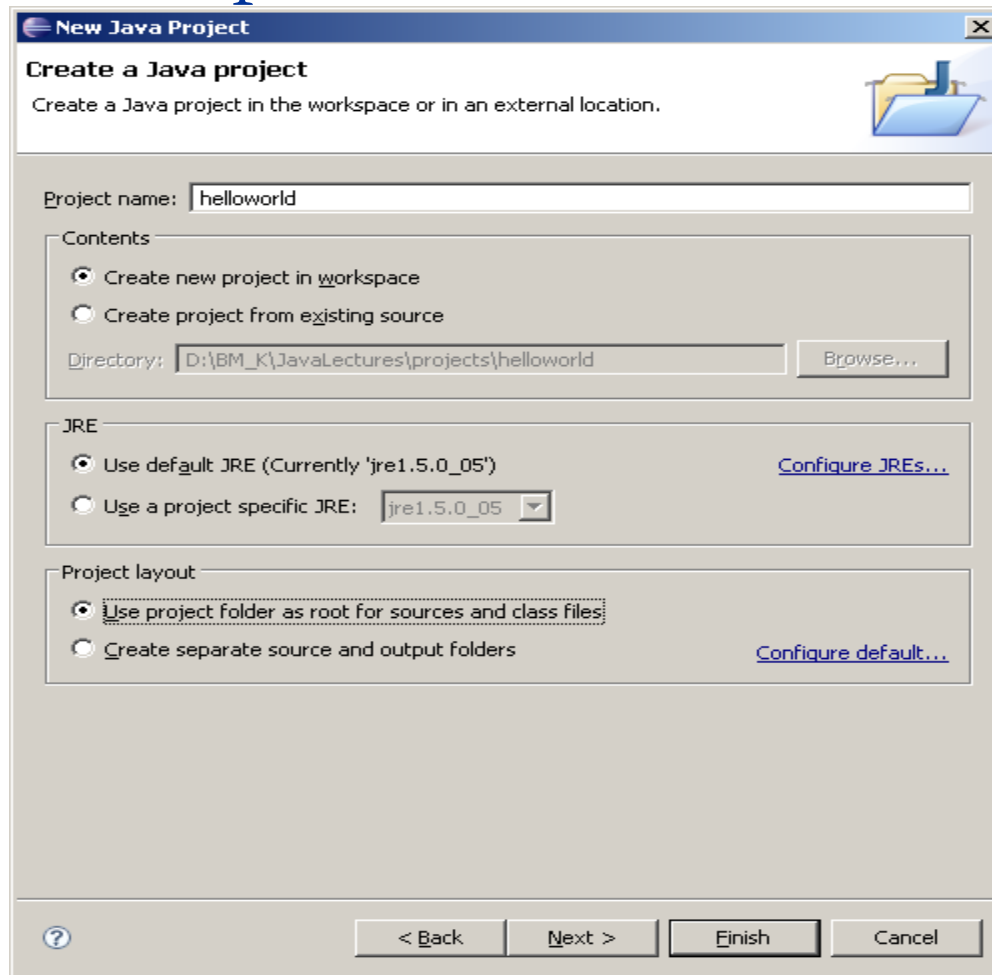
- Имя
- Идентификатор
- Имя провайдера
- Версию
- Список других требуемых подключаемых модулей
org.eclipse.ui.actionSets
- Спецификацию для периода выполнения

Разработка простейшей точки расширения.

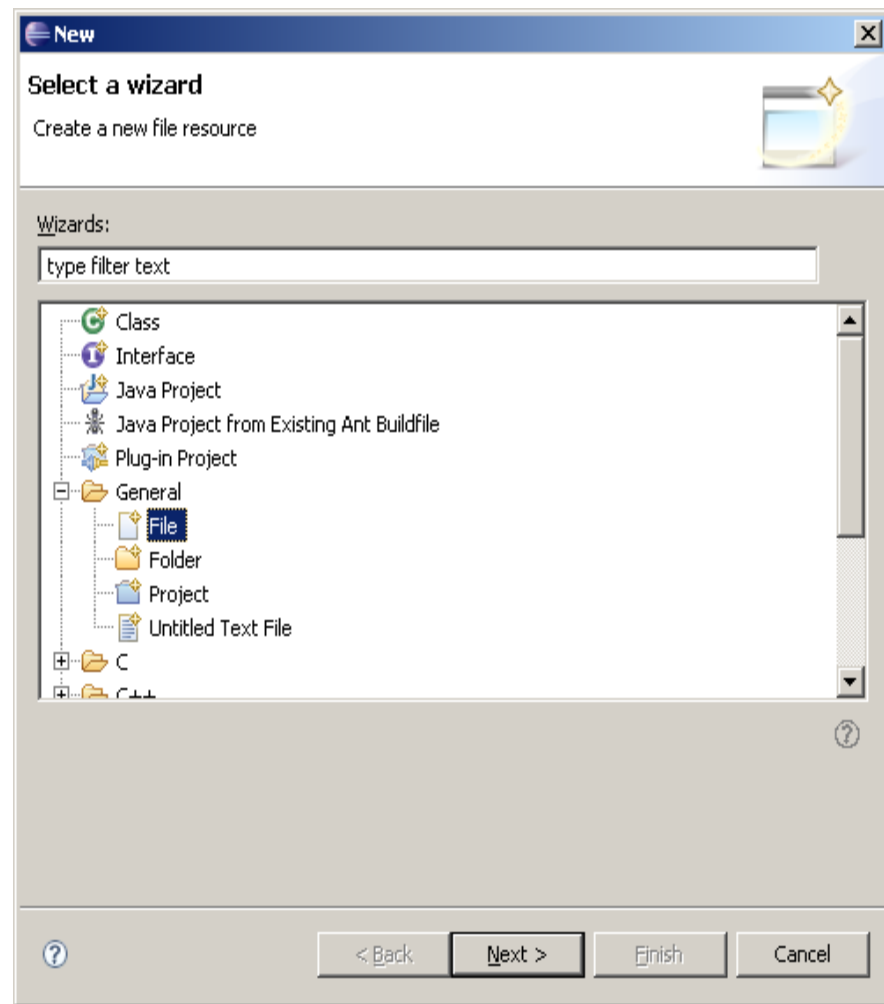
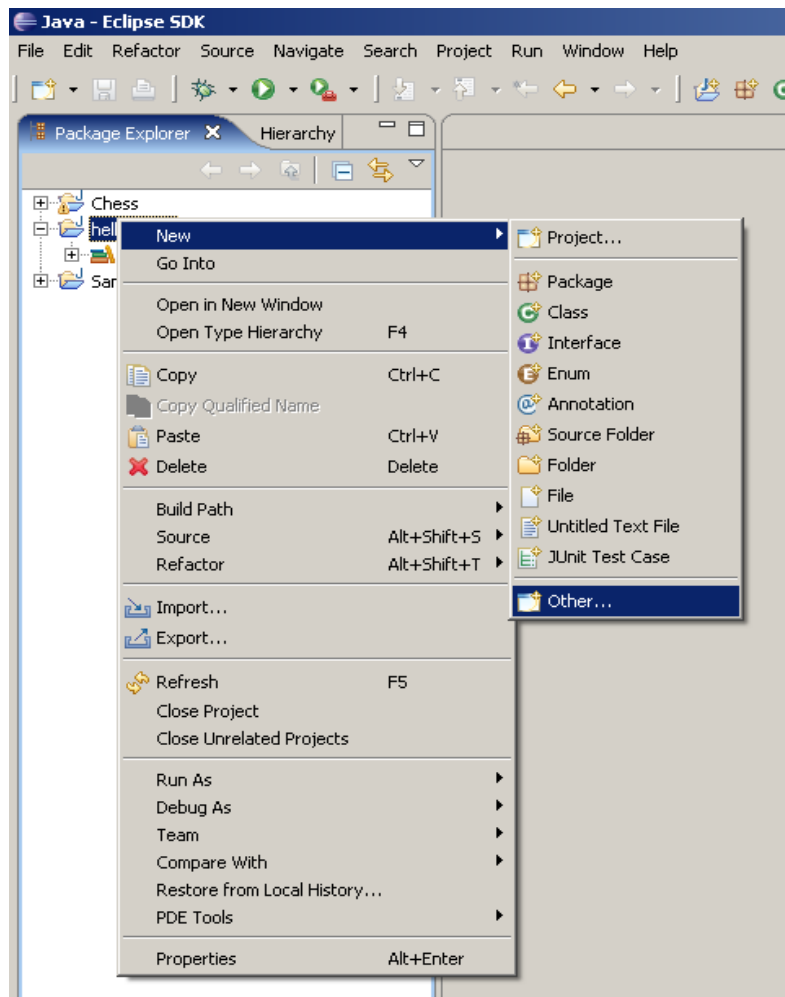
Создание Java проекта



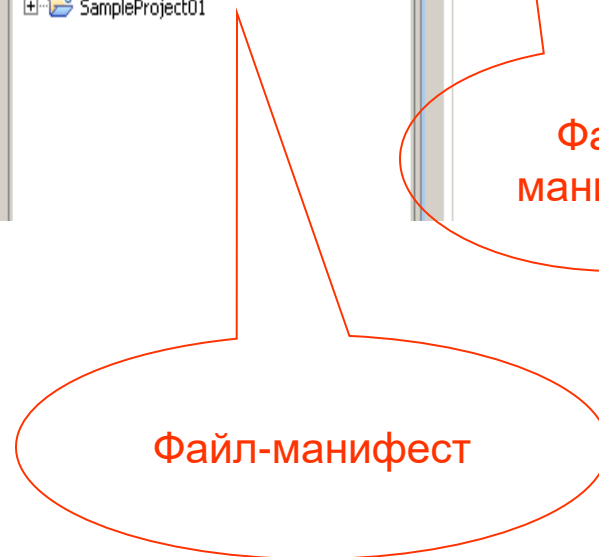
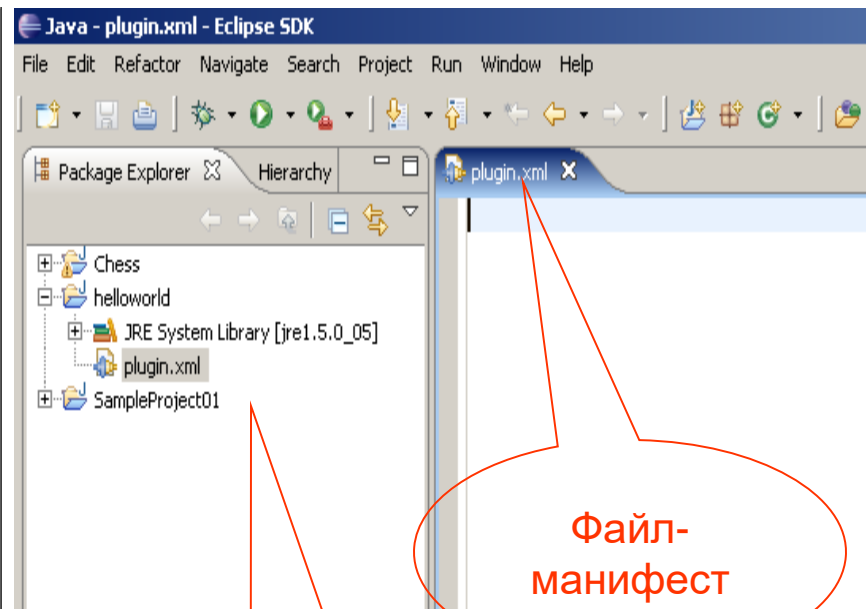
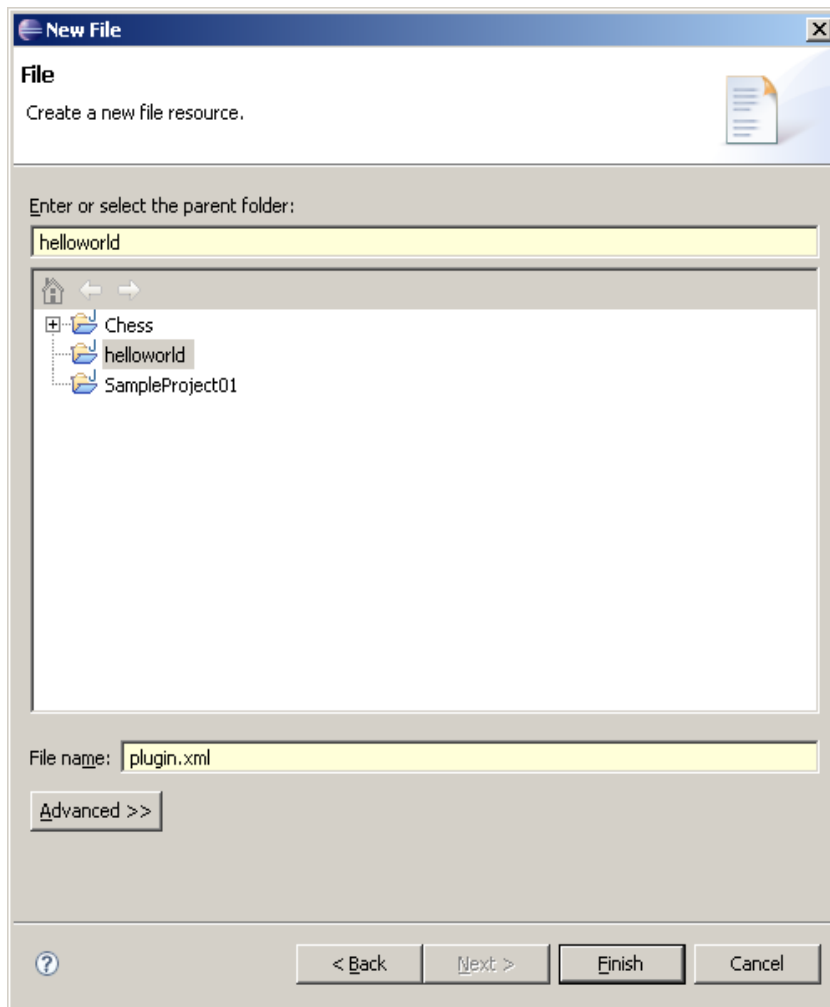
Разработка простейшей точки расширения. Именованние проекта



Разработка простейшей точки расширения. Создание файла-манифеста *plugin.xml* (1)



Разработка простейшей точки расширения. Создание файла-манифеста *plugin.xml* (2)



Разработка простейшей точки расширения. Структура файла-манифеста *plugin.xml* (1)

```
<?xml version="1.0" ?>
```

```
<plugin
```

```
  name="Eclipse Hello World Example"
```

```
  id="org.eclipse.examples.helloworld"
```

```
  version="0.0.1"
```

```
  provider-name="MSU" >
```

```
<requires>
```

```
  <import plugin="org.eclipse.core.resources" />
```

```
  <import plugin="org.eclipse.ui" />
```

```
</requires>
```

```
<runtime>
```

```
  <library name="helloworld.jar" />
```

```
</runtime>
```

```
<extension point = "org.eclipse.ui.actionSets" >
```

```
</extension>
```

```
</plugin>
```

Должен быть уникален

От каких модулей зависит

В каком файле искать модули

Точка расширения **actionSets** из модуля **org.eclipse.ui**.

Разработка простейшей точки расширения.

Структура файла-манифеста *plugin.xml* (2)

```
<extension point = "org.eclipse.ui.actionSets" >  
<actionSet  
  id="org.eclipse.examples.helloworld.HelloWorldActionSet"  
  label="Hello World"  
  visible="true"  
  description="The action set for the Eclipse Hello World example">  
  <menu  
    id="org.eclipse.examples.helloworld.HelloWorldMenu"  
    label="Samples" >  
    <separator name="samples"/>  
  </menu>  
  <action  
    id="org.eclipse.examples.helloworld.actions.HelloWorldAction"  
    menubarPath="org.eclipse.examples.helloworld.HelloWorldMenu/samples"  
    toolbarPath="Normal"  
    label="Hello World"  
    tooltip="Press to see a message"  
    icon="icons/helloworld.gif"  
    class="org.eclipse.examples.helloworld.HelloWorldAction"/>  
</extension>
```

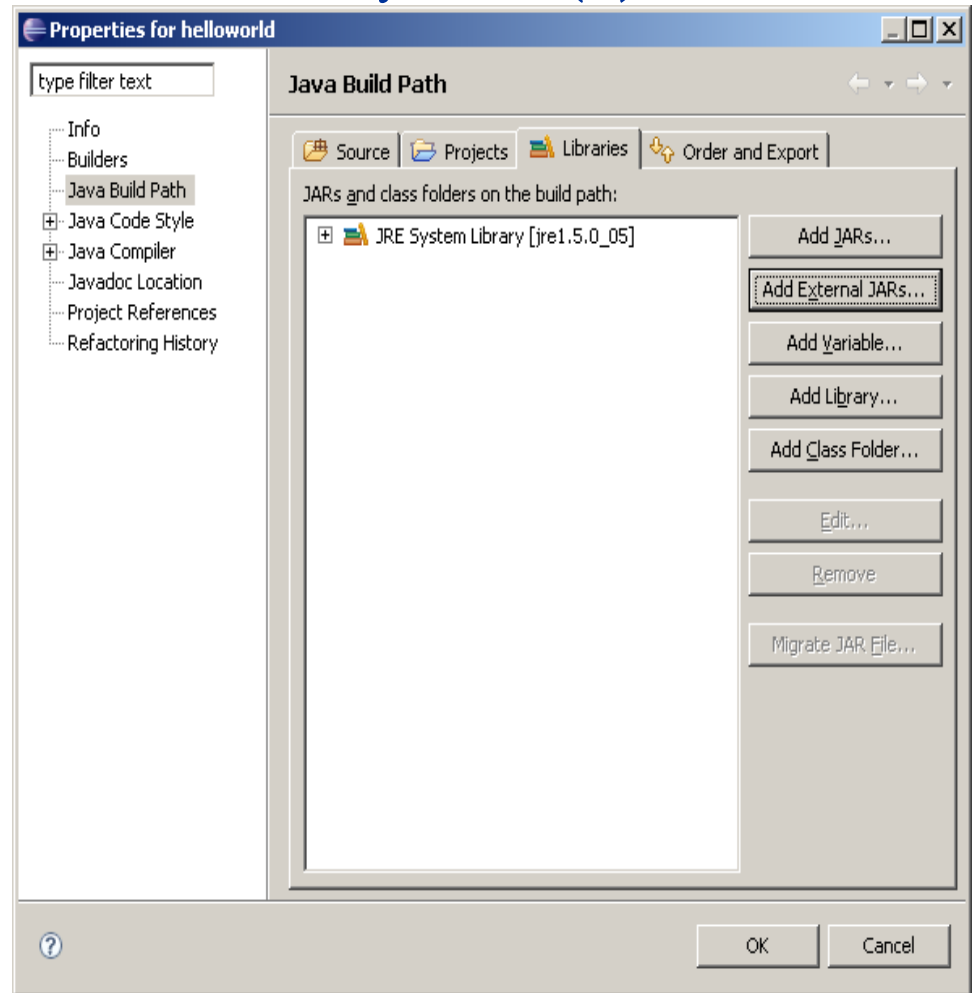
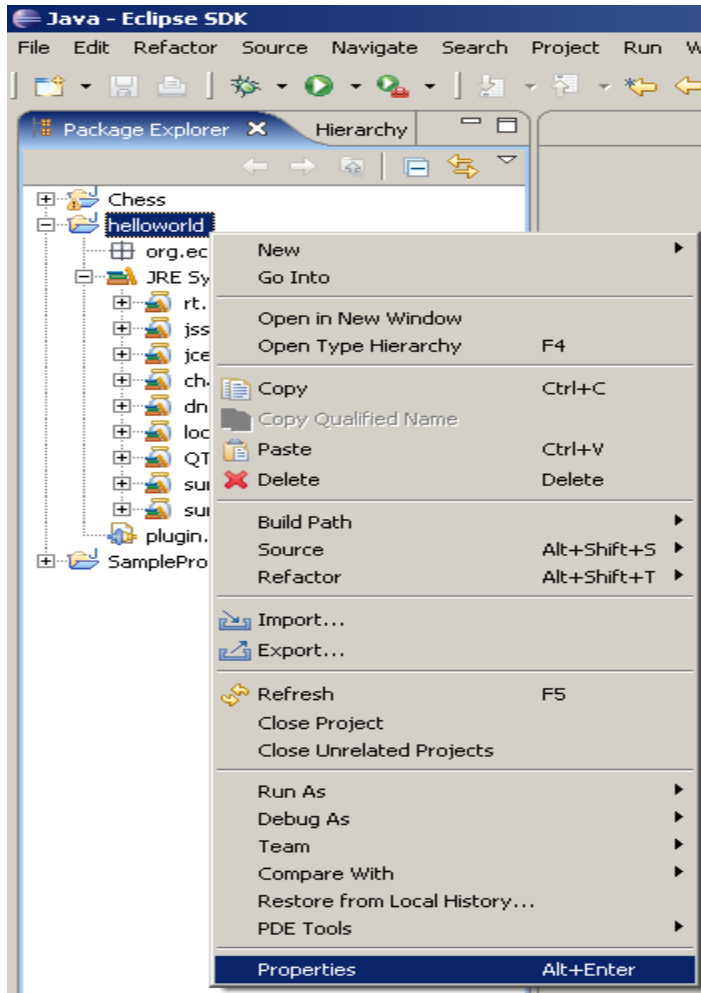
Имя расширения для системы

Имя для отображения

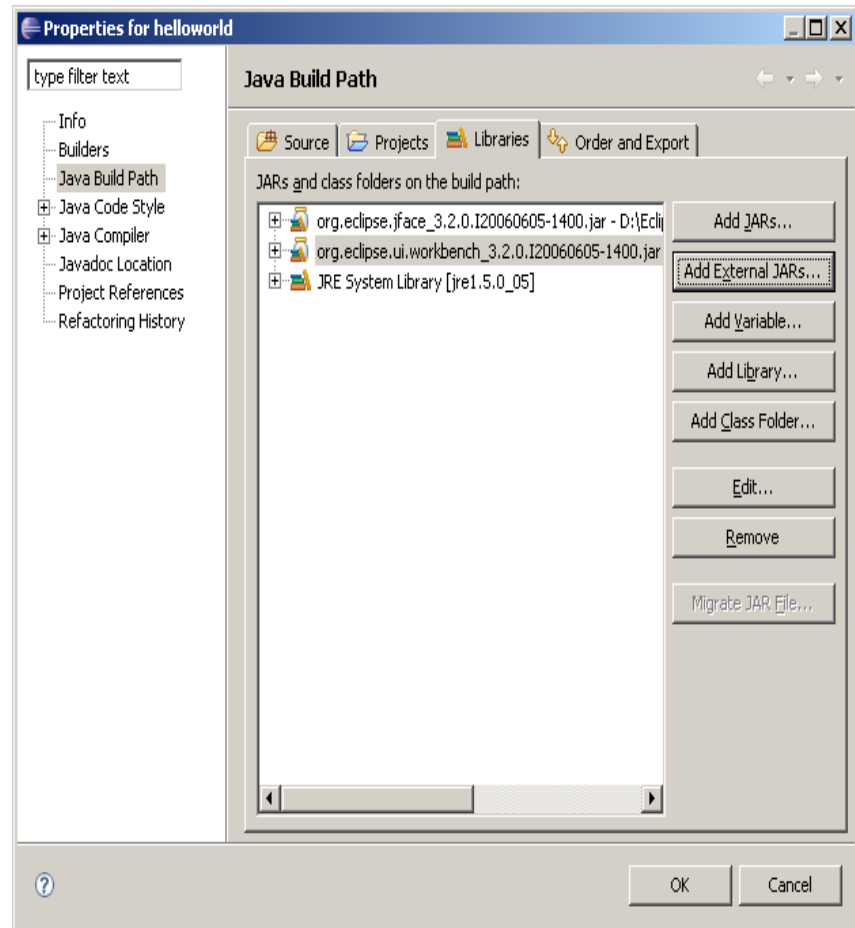
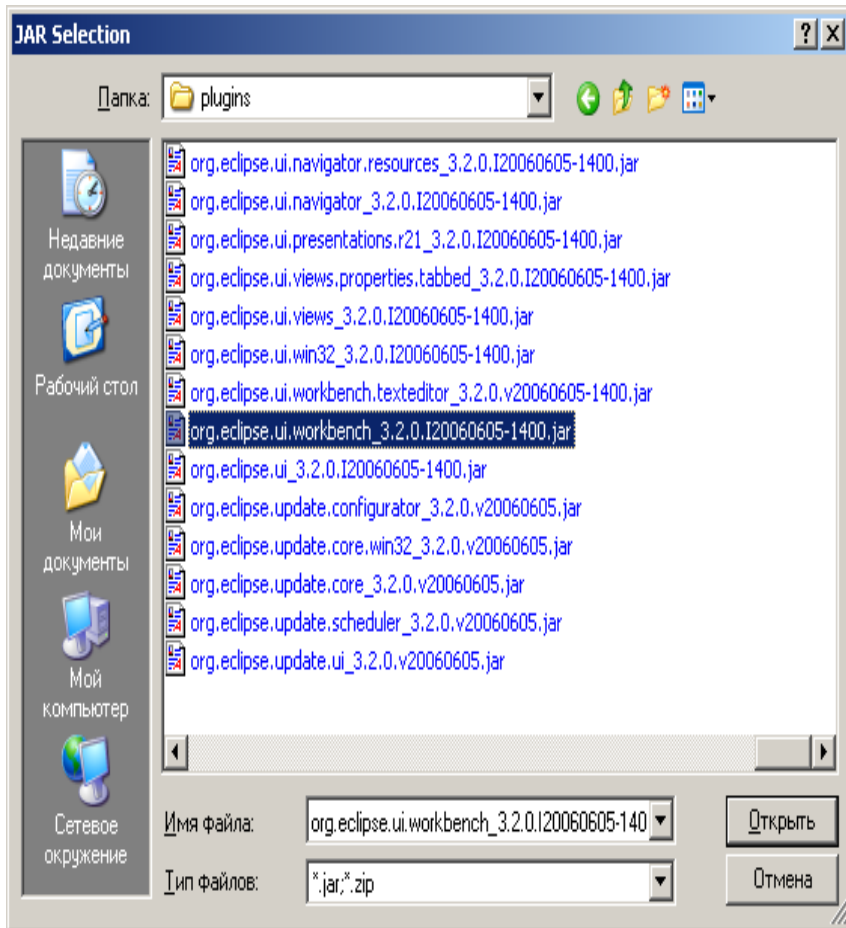
Класс реализующий интерфейс IWorkbenchWindowActionDelegate

Разработка простейшей точки расширения.

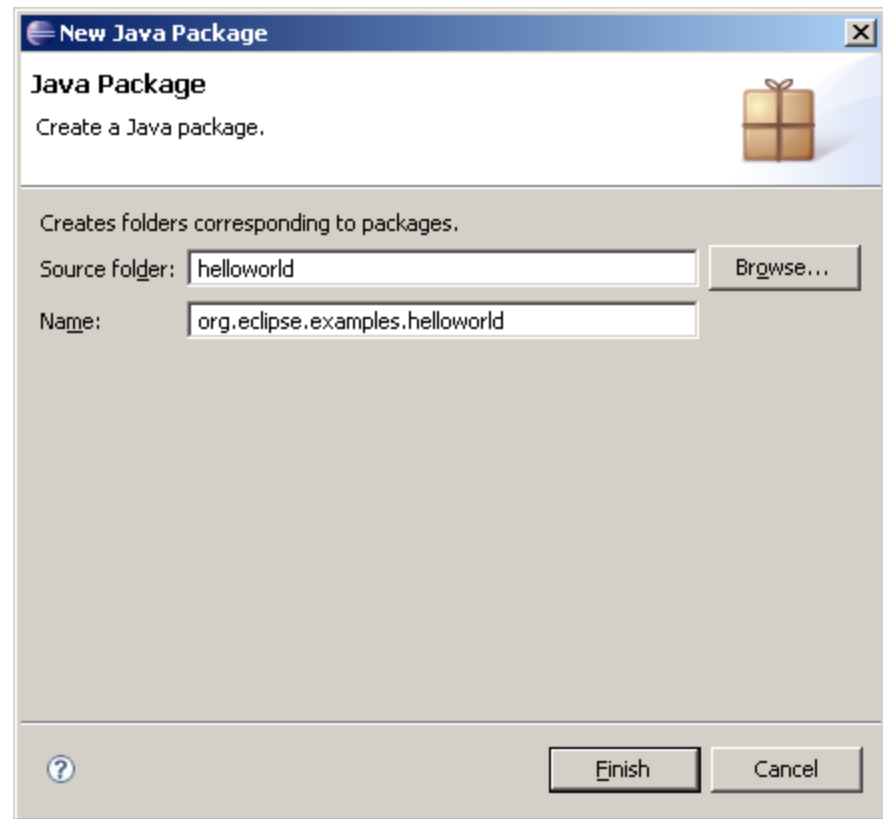
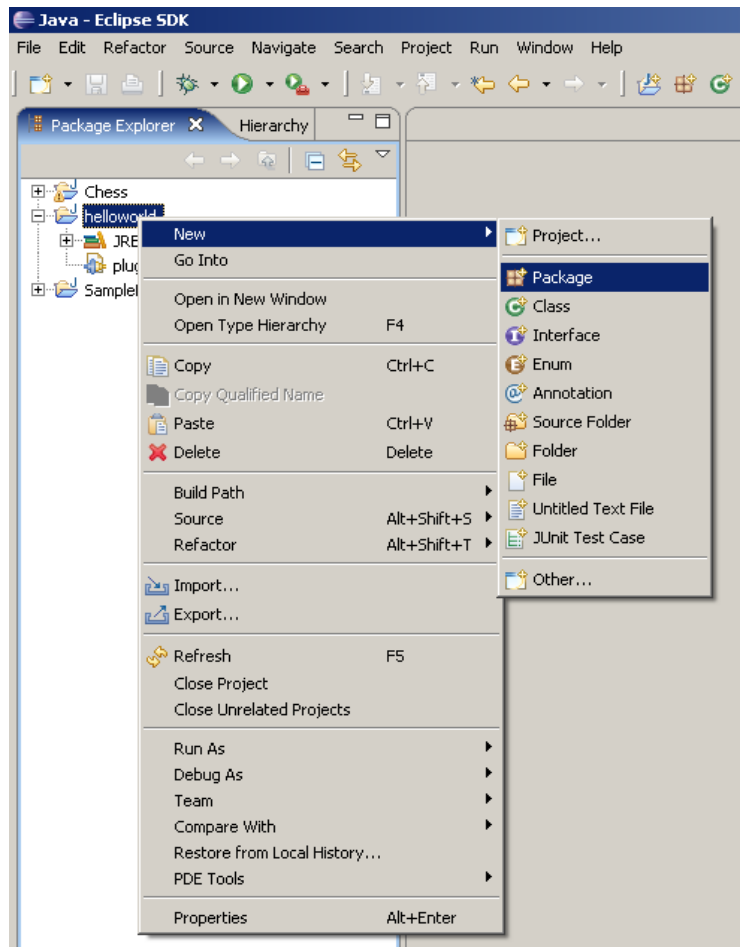
Подключение библиотеки модулей (1)



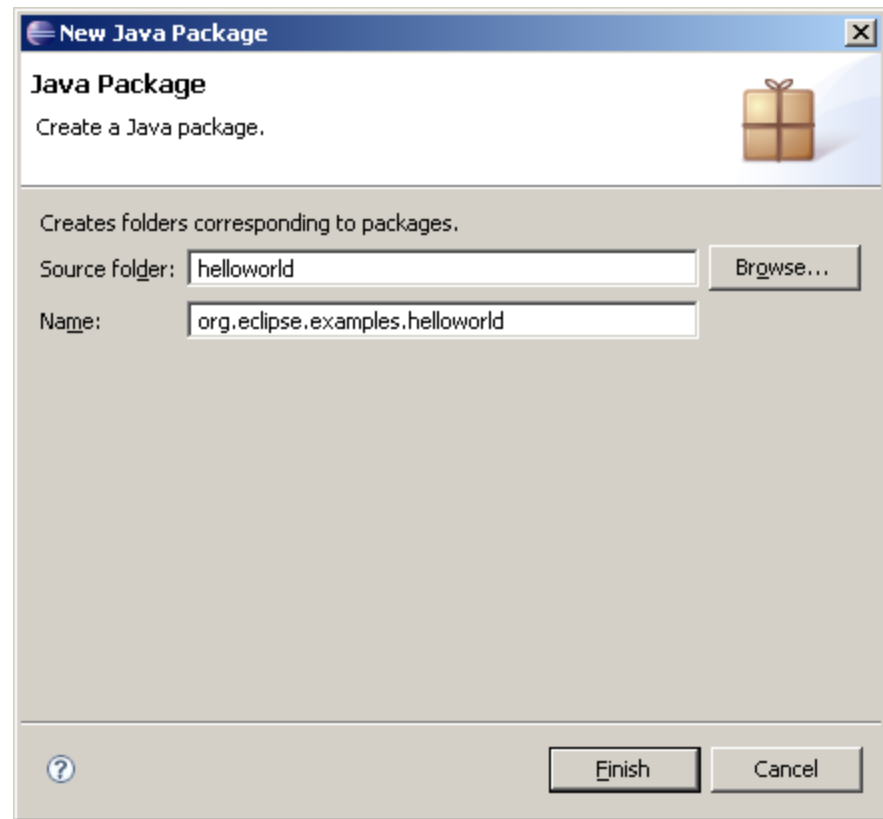
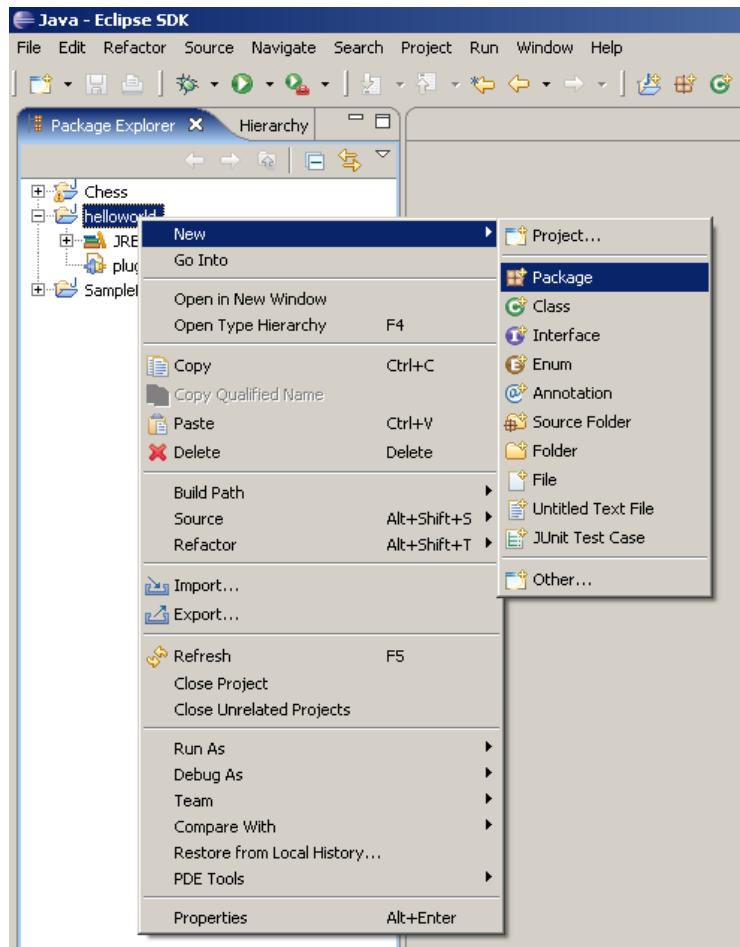
Разработка простейшей точки расширения. Подключение библиотеки модулей (2)



Разработка простейшей точки расширения. Создание пакета для реализации расширения



Разработка простейшей точки расширения. Создание пакета для реализации расширения



Разработка простейшей точки расширения. Создание класса для реализации расширения

```
package org.eclipse.examples.helloworld;

import org.eclipse.ui.IWorkbenchWindow;
import org.eclipse.ui.IWorkbenchWindowActionDelegate;
import org.eclipse.jface.action.IAction;
import org.eclipse.jface.dialogs.MessageDialog;
import org.eclipse.jface.viewers.ISelection;
import org.eclipse.swt.widgets.Shell;

public class HelloWorldAction implements IWorkbenchWindowActionDelegate {
    IWorkbenchWindow activeWindow = null;

    public void run(IAction proxyAction) {
        // proxyAction has UI information from manifest file (ignored)
        Shell shell = activeWindow.getShell();
        MessageDialog.openInformation(shell, "Hello Eclipse", "Hello Eclipse!");
    }

    public void selectionChanged(IAction proxyAction, ISelection selection) {}

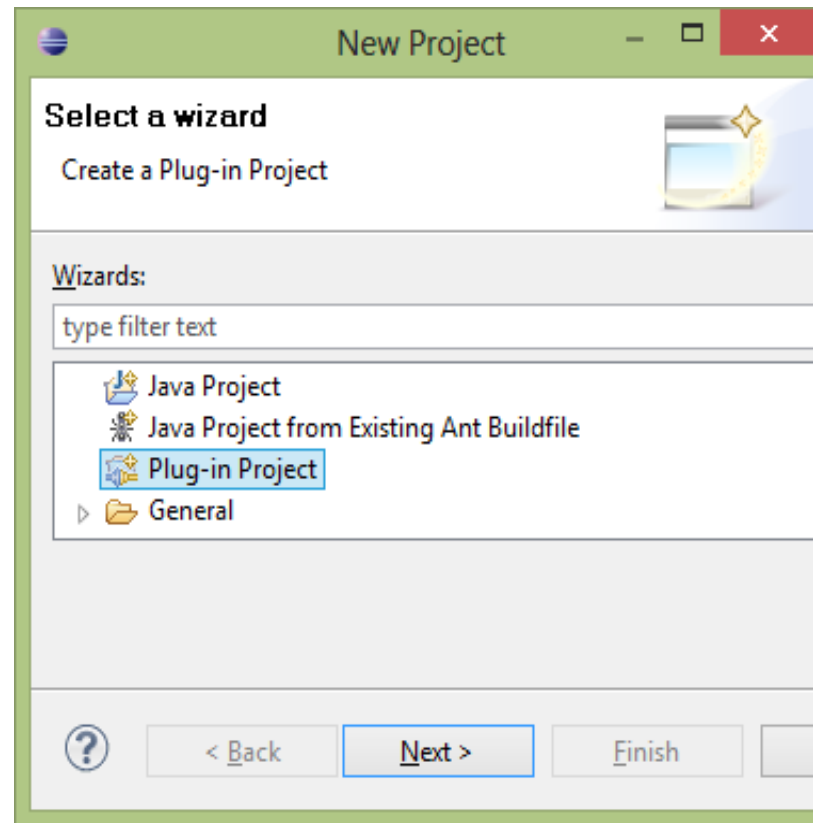
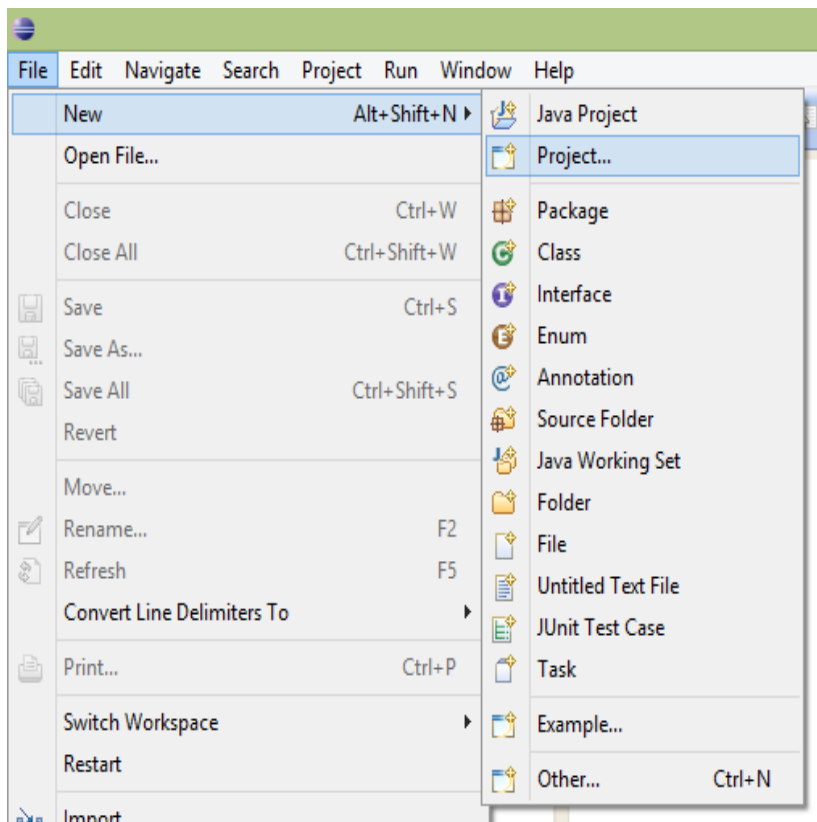
    public void init(IWorkbenchWindow window)
        { activeWindow = window; }

    public void dispose() {}
}
```

Разработка
простейшей точки расширения.

С использованием инструмента
Plugin Development Environment

Разработка простейшей точки расширения. Создание проекта с помощью PDE



Разработка простейшей точки расширения. Задаем имя и свойства проекта

Plug-in Project
Create a new plug-in project

Project name:

Use default location
Location:

Project Settings
 Create a Java project
Source folder:
Output folder:

Target Platform
This plug-in is targeted to run with:
 Eclipse version:
 an OSGi framework:

Working sets
 Add project to working sets
Working sets:

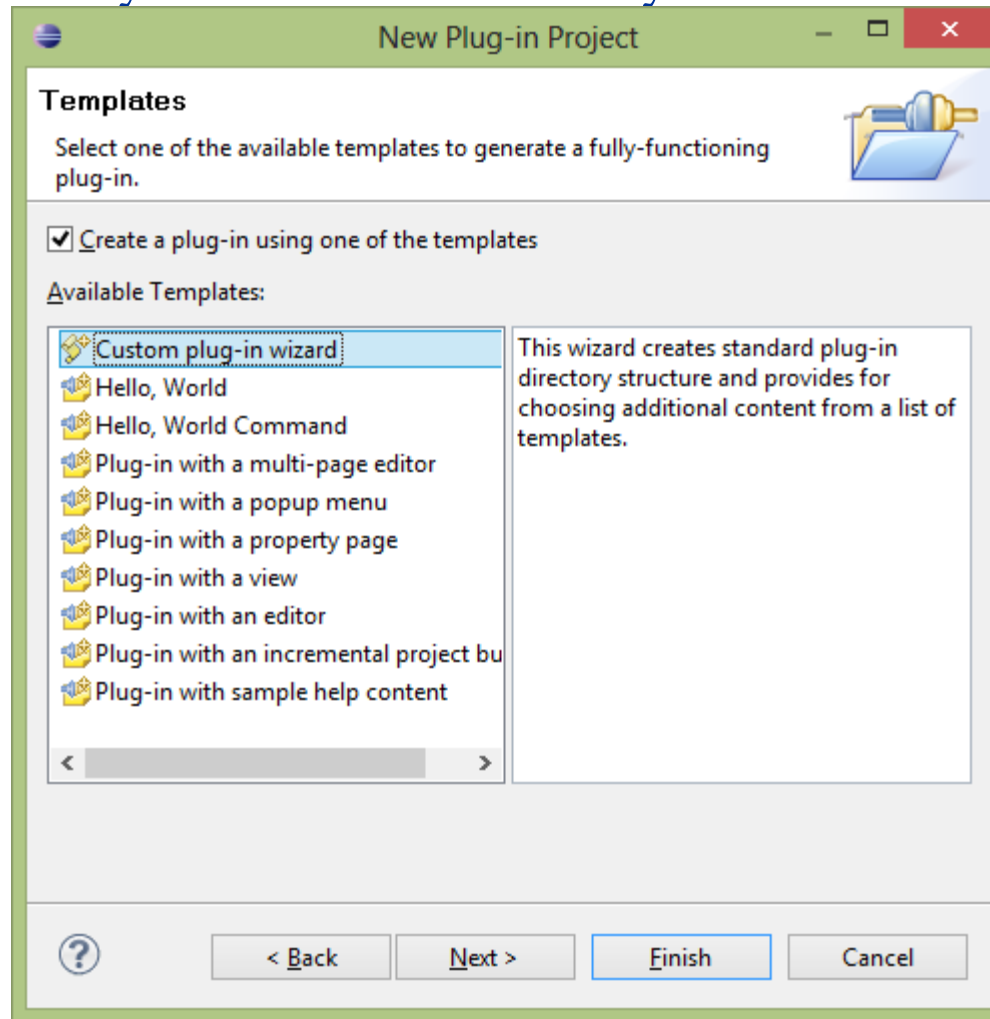
Content
Enter the data required to generate the plug-in.

Properties
ID:
Version:
Name:
Vendor:
Execution Environment:

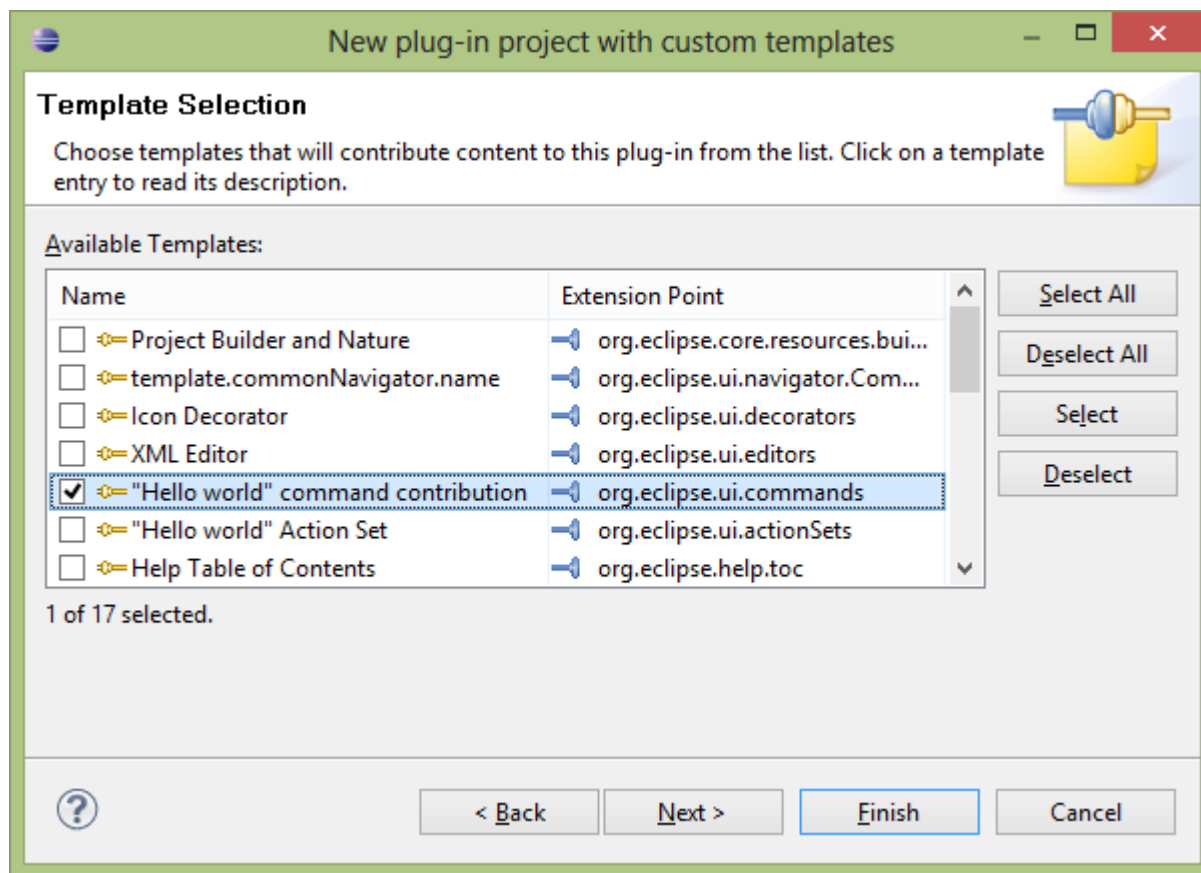
Options
 Generate an activator, a Java class that controls the plug-in's life cycle
Activator:
 This plug-in will make contributions to the UI
 Enable API analysis

Rich Client Application
Would you like to create a 3.x rich client application? Yes No

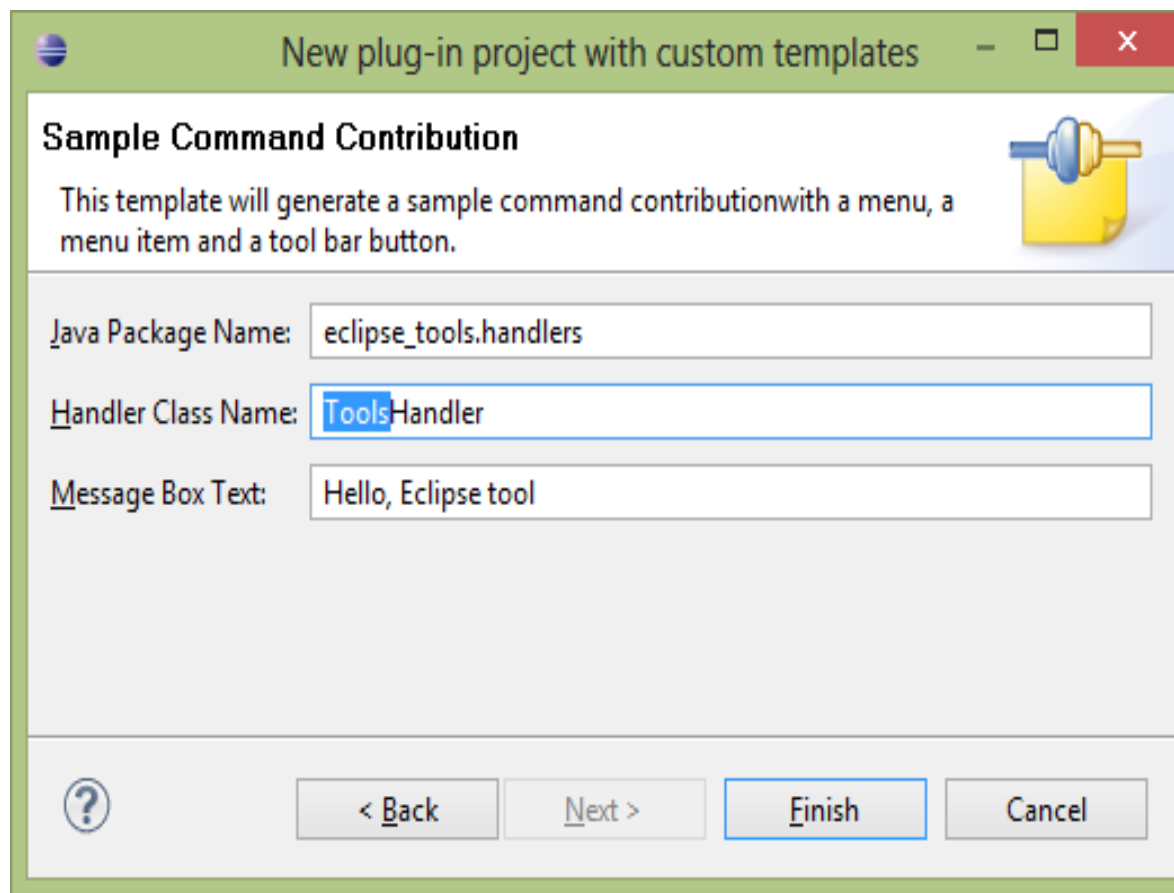
Разработка простейшей точки расширения. Создание модуля по шаблону



Разработка простейшей точки расширения. Выбор шаблонов для модуля



Разработка простейшей точки расширения. Настройка точки расширения



New plug-in project with custom templates

Sample Command Contribution

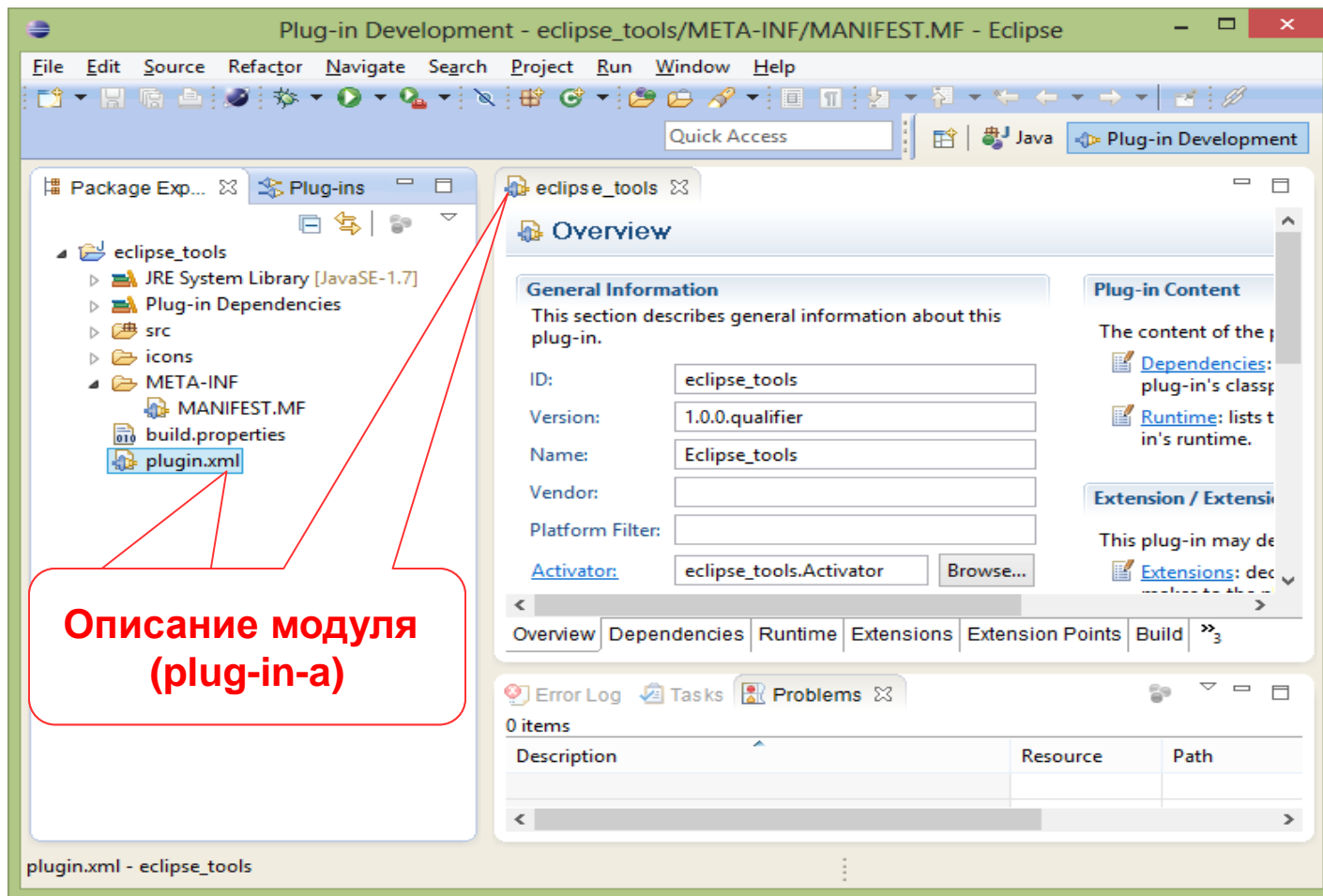
This template will generate a sample command contribution with a menu, a menu item and a tool bar button.

Java Package Name:

Handler Class Name:

Message Box Text:

Разработка простейшей точки расширения. Сгенерированный с помощью PDE проект



Разработка простейшей точки расширения. Редактор модуля PDE. Обзор

Кнопка запуска модуля на исполнение

General Information
This section describes general information about this plug-in.

ID:

Version:

Name:

Vendor:

Platform Filter:

Activator:

Activate this plug-in when one of its classes is loaded

This plug-in is a singleton

Plug-in Content
The content of the plug-in is made up of two sections:

- [Dependencies](#): lists all the plug-ins required on this plug-in's classpath to compile and run.
- [Runtime](#): lists the libraries that make up this plug-in's runtime.

Extension / Extension Point Content
This plug-in may define extensions and extension points:

- [Extensions](#): declares contributions this plug-in makes to the platform.
- [Extension Points](#): declares new function points this plug-in adds to the platform.

Overview | Dependencies | Runtime | Extensions | Extension Points | Build | MANIFEST.MF | plugin.xml | build.properties

Разработка простейшей точки расширения. Редактор модуля PDE. Обзор

Кнопка запуска модуля
на исполнении
для отладки

The screenshot shows the Eclipse IDE's Overview editor for a plug-in named 'eclipse_tools'. The interface is divided into several sections:

- General Information:** This section describes general information about the plug-in. It includes fields for ID (eclipse_tools), Version (1.0.0.qualifier), Name (Eclipse_tools), Vendor, and Platform Filter. There is also a field for the Activator (eclipse_tools.Activator) with a 'Browse...' button. Two checkboxes are present: 'Activate this plug-in when one of its classes is loaded' and 'This plug-in is a singleton', both of which are checked.
- Plug-in Content:** This section describes the content of the plug-in, which is made up of two sections: **Dependencies** (lists all the plug-ins required on this plug-in's classpath to compile and run) and **Runtime** (lists the libraries that make up this plug-in's runtime).
- Extension / Extension Point Content:** This section describes the extensions and extension points defined by the plug-in. It includes **Extensions** (declares contributions this plug-in makes to the platform) and **Extension Points** (declares new function points this plug-in adds to the platform).

At the bottom of the editor, there is a navigation bar with tabs for Overview, Dependencies, Runtime, Extensions, Extension Points, Build, MANIFEST.MF, plugin.xml, and build.properties. The Overview tab is currently selected. In the top right corner of the editor, there are several icons, including a green play button (run) and a gear icon (debug), which are highlighted by a red callout box with the text 'Кнопка запуска модуля на исполнении для отладки'.

Разработка простейшей точки расширения. Редактор модуля PDE. Обзор

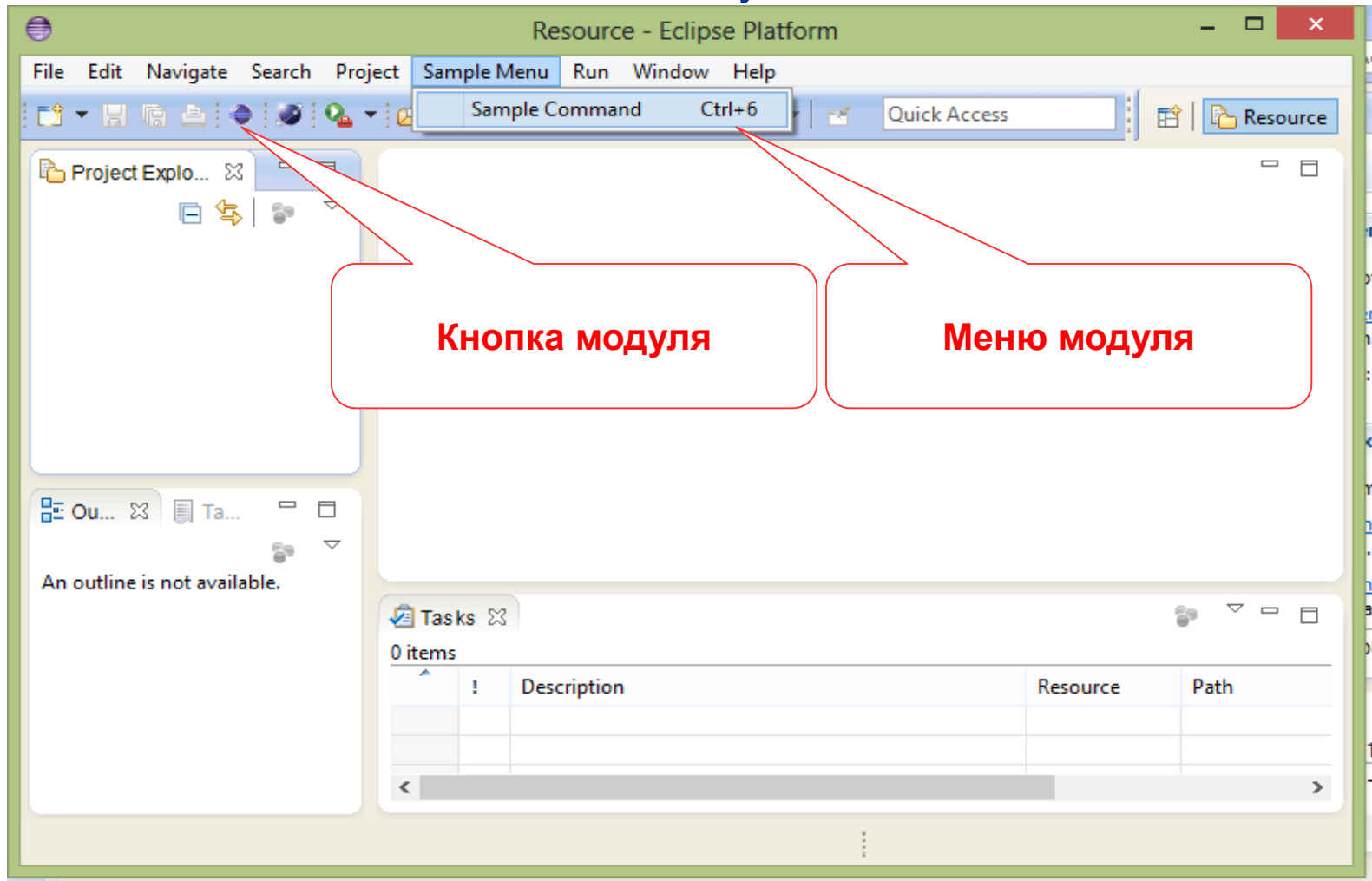
Кнопка для генерации
jar-файла модуля

The screenshot shows the Eclipse IDE's Overview editor for a plug-in named 'eclipse_tools'. The editor is divided into several sections:

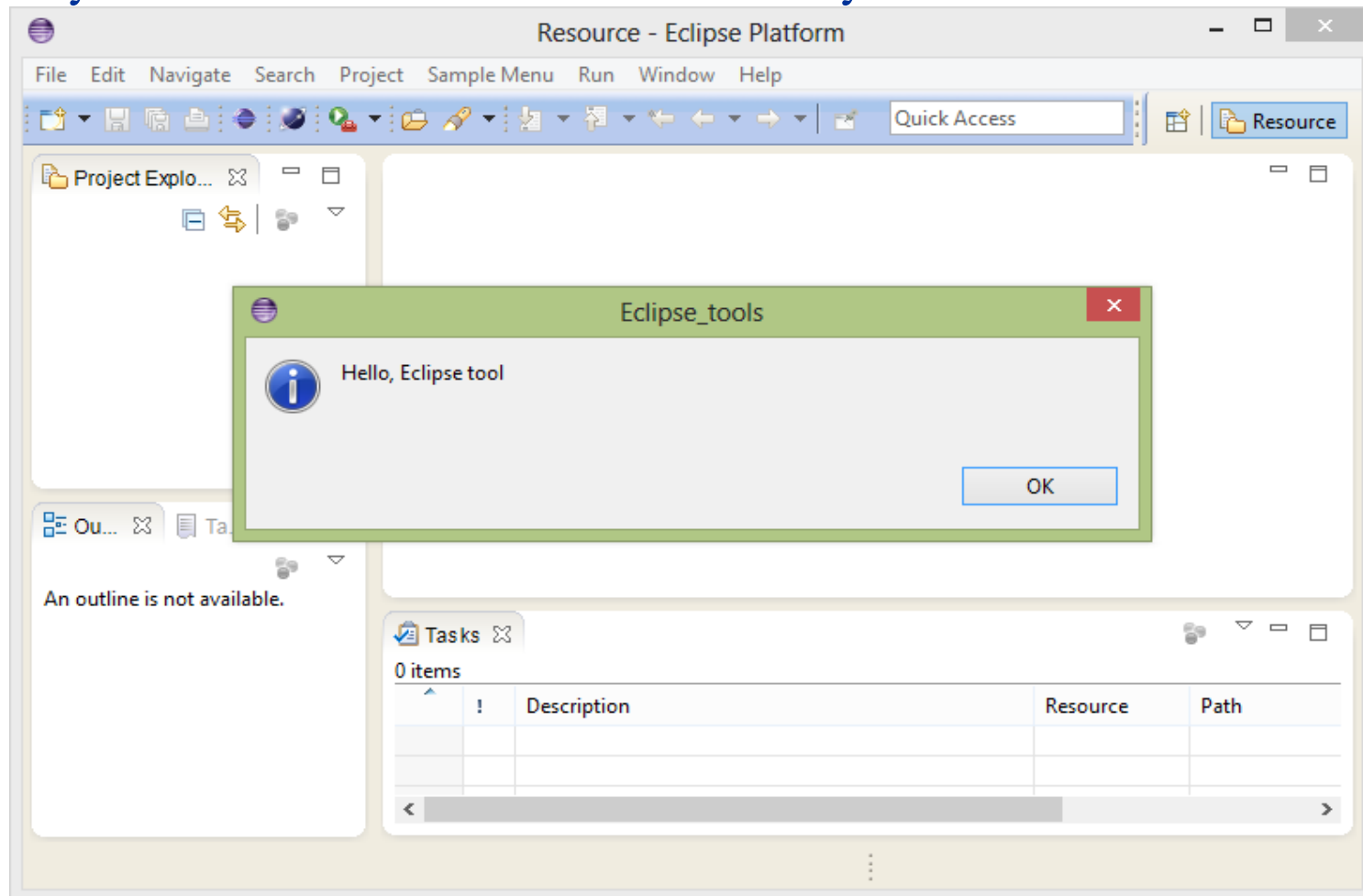
- General Information:** This section describes general information about the plug-in. It contains fields for ID (eclipse_tools), Version (1.0.0.qualifier), Name (Eclipse_tools), Vendor, and Platform Filter. There is also a field for the Activator (eclipse_tools.Activator) with a 'Browse...' button. Two checkboxes are present: 'Activate this plug-in when one of its classes is loaded' and 'This plug-in is a singleton', both of which are checked.
- Plug-in Content:** This section describes the content of the plug-in. It states that the content is made up of two sections: **Dependencies** (lists all the plug-ins required on this plug-in's classpath to compile and run) and **Runtime** (lists the libraries that make up this plug-in's runtime).
- Extension / Extension Point Content:** This section describes the extensions and extension points defined by the plug-in. It states that the plug-in may define extensions and extension points: **Extensions** (declares contributions this plug-in makes to the platform) and **Extension Points** (declares new function points this plug-in adds to the platform).

At the bottom of the editor, there is a navigation bar with tabs for Overview, Dependencies, Runtime, Extensions, Extension Points, Build, MANIFEST.MF, plugin.xml, and build.properties. The Overview tab is currently selected.

Разработка простейшей точки расширения. Меню подключаемого модуля



Разработка простейшей точки расширения. Запуск подключаемого модуля



Разработка простейшей точки расширения. Редактор модуля PDE. XML представление



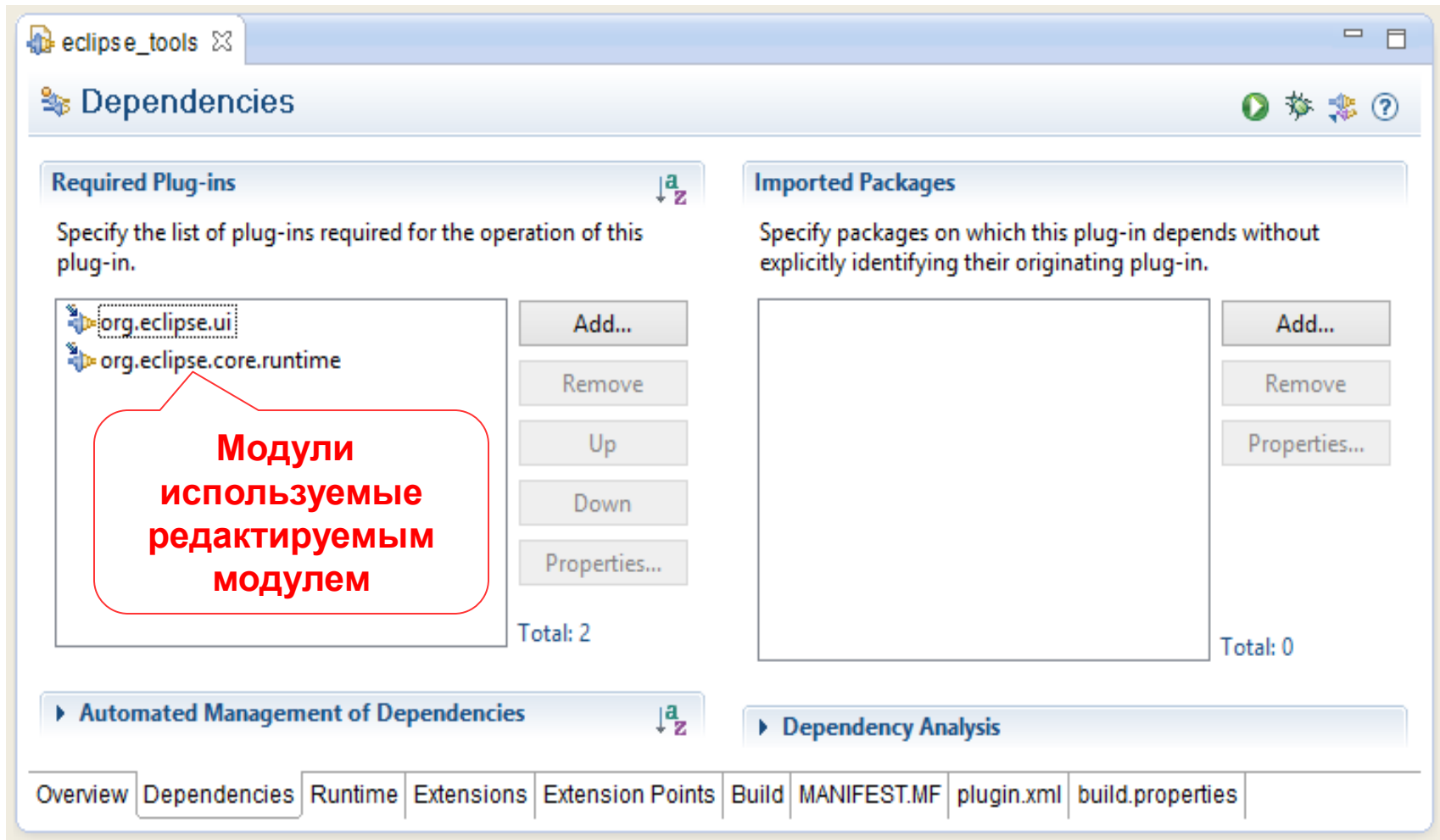
```
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.4"?>
<plugin>

  <extension
    point="org.eclipse.ui.commands">
    <category
      name="Sample Category"
      id="eclipse_tools.commands.category">
    </category>
    <command
      name="Sample Command"
      categoryId="eclipse_tools.commands.category"
      id="eclipse_tools.commands.sampleCommand">
    </command>
  </extension>
  <extension
    point="org.eclipse.ui.handlers">
    <handler
      commandId="eclipse_tools.commands.sampleCommand">
    </handler>
  </extension>
</plugin>
```

Overview | Dependencies | Runtime | Extensions | Extension Points | Build | MANIFEST.MF | plugin.xml | build.properties

Редактор модуля PDE.

Зависимости от других используемых модулей



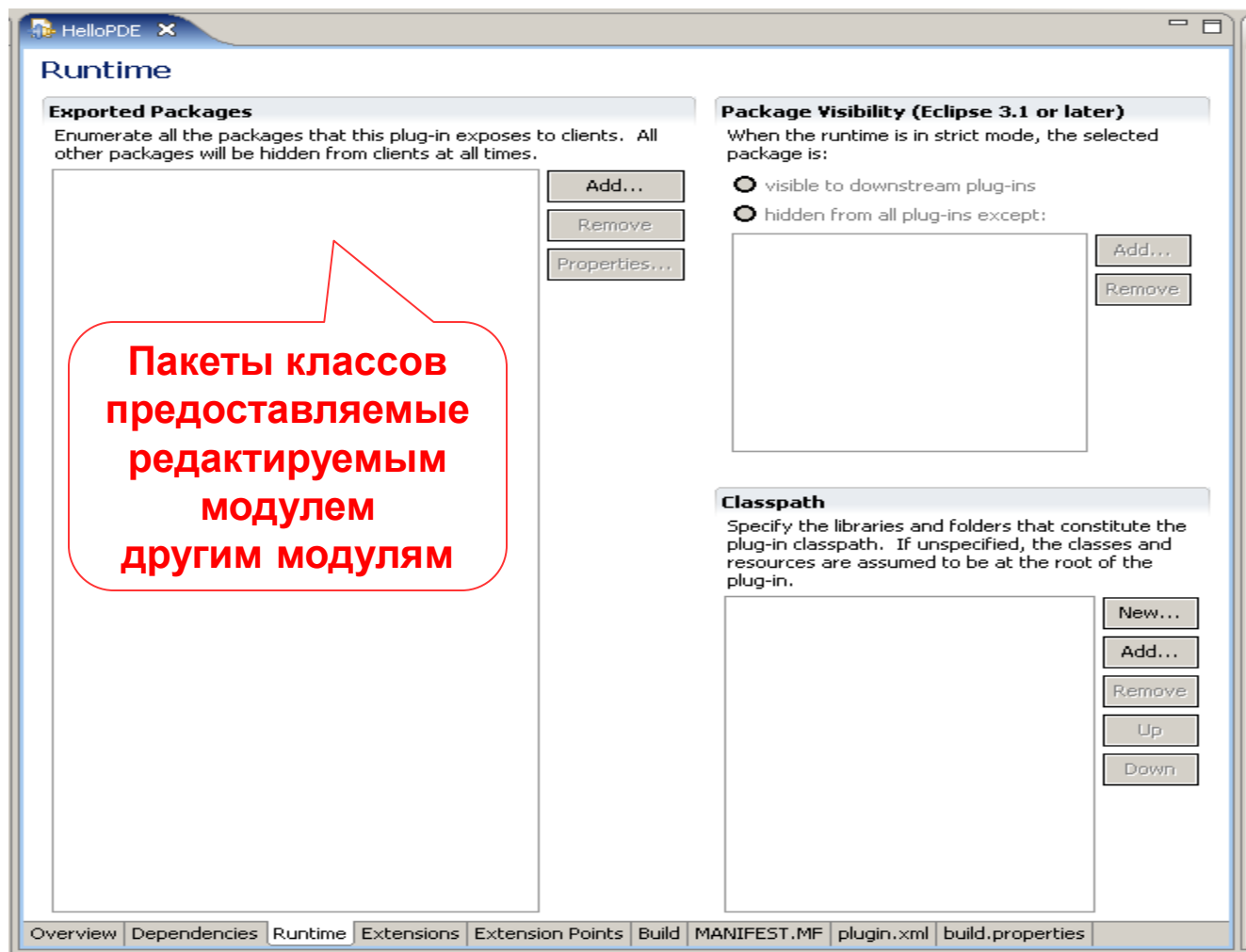
Редактор модуля PDE.

Описание зависимостей модуля

- Секция зависимостей перечисляет все модули, требуемые редактируемым модулем:
 - Все модули, которые необходимы для компиляции редактируемого модуля
 - Все модули, точки расширения которых использует редактируемый модуль при исполнении
- Эта секция позволяет автоматизировать указание месторасположения всех подключаемых модулей которые использует редактируемый модуль

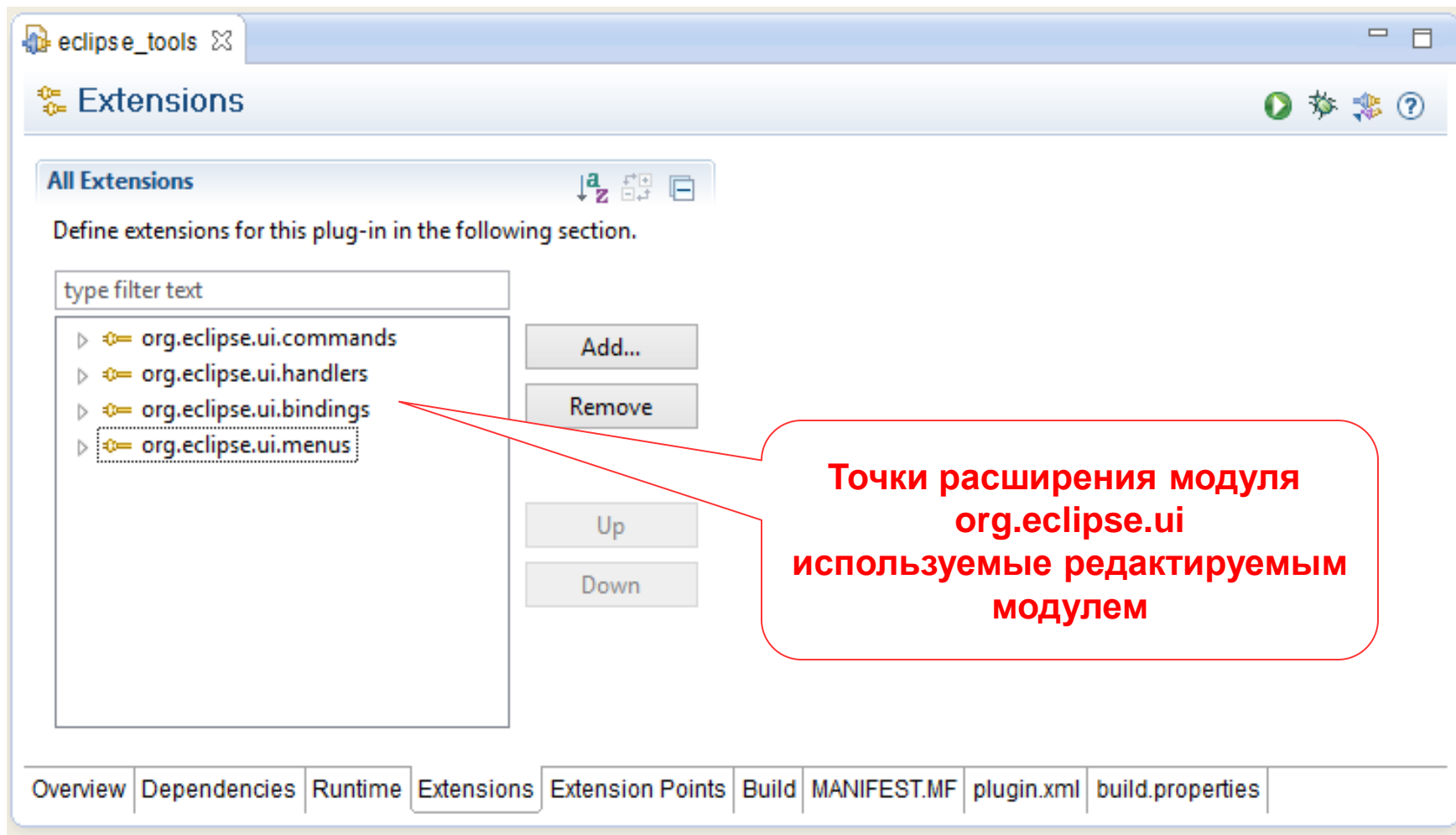
Редактор модуля PDE.

Пакеты доступные во время выполнения



Редактор модуля PDE.

Используемые точки расширения



Редактирование точек расширения. Команды модуля. Категории команд

The screenshot shows the Eclipse IDE's 'Extensions' editor. The left pane, titled 'All Extensions', displays a tree view of extension categories. The selected category is 'EclipsePracticum2016Category (category)'. The right pane, titled 'Extension Element Details', shows the configuration for this category. The 'id*' field is set to 'eclipse2016practicum.commands.category', and the 'name*' field is set to 'EclipsePracticum2016Category'. The 'description' field is empty. Red callout boxes highlight the 'id*' field as the 'Уникальный идентификатор категории' (Unique category identifier) and the 'name*' field as the 'Имя категории для пользователя' (Category name for the user).

Имя категории для пользователя

Уникальный идентификатор категории

Редактирование точек расширения. Команды модуля. Категории команд

The screenshot shows the Eclipse IDE's 'Extensions' editor for the 'eclipse2016praciticum' project. The left pane, titled 'All Extensions', displays a tree view of extension categories. The 'org.eclipse.ui.commands' category is expanded, showing 'EclipsePracticum2016Category (category)' and 'EclipsePracticum2016Command (command)'. The 'EclipsePracticum2016Command (command)' is selected. The right pane, titled 'Extension Element Details', shows the configuration for the selected command. The 'id*' field is set to 'eclipse2016praciticum.commands.sampleCommand'. The 'name*' field is set to 'EclipsePracticum2016Command'. The 'categoryId' field is set to 'eclipse2016praciticum.commands.category'. The 'defaultHandler' field is empty. The 'returnTypeId' field is empty. The 'helpContextId' field is empty. The 'Add...' and 'Remove' buttons are visible in the left pane. The 'Overview', 'Dependencies', 'Runtime', 'Extensions', 'Extension Points', 'Build', 'MANIFEST.MF', 'plugin.xml', and 'build.properties' tabs are visible at the bottom.

Уникальный идентификатор команды

Имя команды для пользователя

Идентификатор категории команды

Редактирование точек расширения. Команды модуля. Обработчики команд

The screenshot displays the Eclipse IDE's 'Extensions' editor for the 'eclipse2016practicum' plug-in. The 'All Extensions' list on the left shows a tree structure with 'org.eclipse2016practicum.commands.sampleCommand (handler)' selected. The 'Extension Element Details' panel on the right shows the configuration for this handler. The 'commandId*' field is set to 'eclipse2016practicum.commands.sampleCommand', and the 'class' field is set to 'eclipse2016practicum.handlers.SampleHandler'. Two red callout boxes highlight these fields: one for 'commandId*' and one for 'class'.

Уникальный идентификатор команды

Класс в программе выполняющий команду

Разработка простейшей точки расширения. Код класса для исполнения команды.

```
import org.eclipse.core.commands.ExecutionException;
import org.eclipse.ui.IWorkbenchWindow;
import org.eclipse.ui.handlers.HandlerUtil;
import org.eclipse.jface.dialogs.MessageDialog;

/**
 * Our sample handler extends AbstractHandler, an IHandler base class.
 * @see org.eclipse.core.commands.IHandler
 * @see org.eclipse.core.commands.AbstractHandler
 */
public class SampleHandler extends AbstractHandler {
    /**
     * The constructor.
     */
    public SampleHandler() {
    }
    /**
     * the command has been executed, so extract extract the needed information
     * from the application context.
     */
    public Object execute(ExecutionEvent event) throws ExecutionException {
        IWorkbenchWindow window = HandlerUtil.getActiveWorkbenchWindowChecked(event);
        MessageDialog.openInformation(
            window.getShell(),
            "Eclipse2016practicum",
            "Hello, Eclipse world");
        return null;
    }
}
```

Редактирование точек расширения. Команды модуля. Клавиши для команд

The screenshot shows the Eclipse IDE's 'Extensions' editor. On the left, the 'All Extensions' view shows a tree structure with the following items:

- org.eclipse.ui.commands
- org.eclipse.ui.handlers
- org.eclipse.ui.bindings
 - eclipse2016practicum.commands.sampleCommand (key)**
- org.eclipse.ui.menus

Buttons for 'Add...', 'Remove', 'Up', and 'Down' are visible next to the tree. The 'eclipse2016practicum.commands.sampleCommand (key)' item is selected.

On the right, the 'Extension Element Details' view shows the configuration for the selected extension. The fields are:

- sequence*: M1+6
- schemelid*: org.eclipse.ui.defaultAcceleratorConfiguration (Browse...)
- contextId: org.eclipse.ui.contexts.window (Browse...)
- commandId: eclipse2016practicum.commands.sampleCommand (Browse...)
- platform: (empty)
- locale: (empty)

Two red callout boxes highlight specific fields:

- A callout box points to the 'sequence*' field with the text: **Горячая клавиша команды**
- A callout box points to the 'commandId' field with the text: **Уникальный идентификатор команды**

At the bottom of the IDE, the breadcrumb navigation shows: Overview | Dependencies | Runtime | Extensions | Extension Points | Build | MANIFEST.MF | plugin.xml | build.properties

Редактирование точек расширения. Команды модуля. Меню для команды

The screenshot shows the Eclipse IDE's 'Extensions' editor. On the left, the 'All Extensions' view displays a tree of extension points. The 'org.eclipse.ui.menu' extension point is expanded, and the 'menu:org.eclipse.ui.main.menu?after=additions (menuContribution)' is selected. On the right, the 'Extension Element Details' view shows the configuration for this menu contribution. The 'locationURI*' field is set to 'menu:org.eclipse.ui.main.menu?after=additions'. A red callout box points to this field with the text 'Положение меню (главное меню)'. The bottom of the editor shows a tabbed interface with 'Extensions' selected.

**Положение меню
(главное меню)**

Редактирование точек расширения. Команды модуля. Меню для команды

The screenshot displays the Eclipse IDE's 'Extensions' editor. The left pane shows a tree view of extension points, with 'EclipsePracticum2016Menu (menu)' selected. The right pane shows the configuration details for this menu item. The configuration fields are:

- label*:** EclipsePracticum2016Menu
- id:** eclipse2016practicum.menus.sampleMenu
- mnemonic:** M
- icon:** (empty field with a 'Browse...' button)
- tooltip:** EclipsePracticum2016
- commandId:** (empty field with a 'Browse...' button)

Red callout boxes highlight the following fields:

- Имя меню для пользователя** (User menu name) points to the **label*** field.
- Уникальный идентификатор меню** (Unique menu identifier) points to the **id** field.
- Подсказка для меню** (Menu tooltip) points to the **tooltip** field.

Редактирование точек расширения.

Команды модуля. Элемент меню для команды

The screenshot shows the Eclipse IDE's 'Extensions' editor. On the left, a tree view shows the extension point hierarchy: `org.eclipse.ui.commands`, `org.eclipse.ui.handlers`, `org.eclipse.ui.bindings`, and `org.eclipse.ui.menu`. Under `org.eclipse.ui.menu`, the menu item `eclipse2016practicum.menus.sampleCommand` is selected. The 'Details' panel on the right shows the configuration for this menu item. The fields are: `commandId` (set to `eclipse2016practicum.commands.sampleCommand`), `label` (set to `eclipse2016practicum.menus.sampleCommand`), `mnemonic` (set to `S`), `icon` (set to `icons/Practicum.gif`), `disabledIcon` (empty), `hoverIcon` (empty), and `tooltip` (set to `EclipsePracticum2016Tooltip`). There are also 'Add...', 'Remove', 'Up', and 'Down' buttons.

Уникальный идентификатор элемента меню

Уникальный идентификатор меню

Имя элемента меню для пользователя (по умолчанию совпадает с именем команды)

Пиктограмма для элемента меню

Подсказка для элемента меню

Редактирование точек расширения. Команды модуля. Положение инструментов

The screenshot displays the Eclipse IDE's 'Extensions' editor for the 'eclipse2016practicum' plug-in. The 'All Extensions' section on the left lists various extension points, with 'eclipse2016practicum.toolbars.sampleToolbar (toolbar)' selected. The 'Extension Element Details' section on the right shows the configuration for the selected extension, with the 'locationURI' field set to 'toolbar:org.eclipse.ui.main.toolbar?after=additions'. A red callout box highlights this field with the text 'Положение инструмента команды (Главная панель инструментов)'. The bottom of the window shows a breadcrumb trail: Overview > Dependencies > Runtime > Extensions > Extension Points > Build > MANIFEST.MF > plugin.xml > build.properties.

Редактирование точек расширения. Команды модуля. Панель инструментов

The screenshot displays the Eclipse IDE's 'Extensions' editor for the 'eclipse2016practicum' project. The interface is divided into several sections:

- All Extensions:** A list of installed extensions for the 'toolbar' category. The selected extension is 'eclipse2016practicum.toolbars.sampleToolbar (toolbar)'. Below the list are buttons for 'Add...', 'Remove', 'Up', and 'Down'.
- Extension Element Details:** A panel for configuring the properties of the selected extension. It shows the 'id*' field set to 'eclipse2016practicum.toolbars.sampleToolbar' and the 'label' field empty. A red callout box points to the 'id*' field with the text: **Уникальный идентификатор панели инструментов модуля**.
- Bottom Tabs:** A row of tabs at the bottom of the editor, including 'Overview', 'Dependencies', 'Runtime', 'Extensions' (selected), 'Extension Points', 'Build', 'MANIFEST.MF', 'plugin.xml', and 'build.properties'.

Редактирование точек расширения. Команды модуля. Инструмент для команды


The screenshot shows the Eclipse IDE's 'Extensions' editor. The left pane displays a tree view of extension points, with 'eclipse2016practicum.toolbars.sampleCommand (command)' selected. The right pane, titled 'Command Details', contains the following fields:

- commandId*:** eclipse2016practicum.commands.sampleCommand
- label:** (empty)
- id:** eclipse2016practicum.toolbars.sampleCommand
- mnemonic:** (empty)
- icon:** icons/Practicum.png
- disableOn:** (empty)
- enableOn:** (empty)
- tooltip:** EclipsePracticum2016-tool-Tooltip

Red callout boxes highlight the following elements:

- Уникальный идентификатор команды** (Unique command identifier) pointing to the **commandId*** field.
- Уникальный идентификатор панели инструментов** (Unique toolbar identifier) pointing to the **id** field.
- Пиктограмма для инструмента** (Icon for the toolbar) pointing to the **icon** field.
- Подсказка для инструмента** (Tooltip for the toolbar) pointing to the **tooltip** field.

Разработка простейшей точки расширения. Редактор модуля PDE. XML-текст для меню



```
<extension
  point="org.eclipse.ui.menus">
  <menuContribution
    locationURI="menu:org.eclipse.ui.main.menu?after=additions">
    <menu
      label="UML Menu"
      mnemonic="M"
      id="eclipse_tools.menus.sampleMenu">
      <command
        commandId="eclipse_tools.commands.sampleCommand"
        icon="icons/DDiagram.gif"
        id="eclipse_tools.menus.sampleCommand"
        label="Hello Message"
        mnemonic="S">
      </command>
    </menu>
  </menuContribution>
  <menuContribution
    locationURI="toolbar:org.eclipse.ui.main.toolbar?after=additions">
  <toolbar
```

The screenshot shows the Eclipse IDE interface with the 'eclipse_tools' project selected. The main editor displays the XML configuration for a menu extension. The configuration includes a menu contribution with a menu item labeled 'UML Menu' and a command labeled 'Hello Message'. The command is associated with the 'sampleCommand' and has an icon 'DDiagram.gif'. The menu is located in the main menu, and the command is located in the toolbar. The IDE's bottom bar shows the 'plugin.xml' file is selected, and the 'Extensions' view is active.

Разработка простейшей точки расширения. Редактор модуля PDE. Пиктограмма инструмента

The screenshot displays the Eclipse IDE interface for editing the `MANIFEST.MF` file of a plugin. The main window is titled "Java - eclipse2016practicum/META-INF/MANIFEST.MF - Eclipse". The "Extensions" editor is active, showing a list of extension points. A yellow callout bubble with the text "Пиктограмма" (Icon) points to the "icon" property in the "Extension Element Details" panel, which is set to "icons/Practicum.png". The "Package Explorer" on the left shows the project structure, including the "icons" folder and the "Practicum.png" file.

The "Extension Element Details" panel shows the following properties:

- `commandId*`: eclipse2016practicum.commands.sampleCommand
- `label`:
- `id`: eclipse2016practicum.toolbars.sampleCommand
- `mnemonic`:
- `icon`: icons/Practicum.png
- `disabledIcon`:
- `hoverIcon`:
- `tooltip`: EclipsePracticum2016-tool-Tooltip

Разработка простейшей точки расширения. Редактор модуля PDE. Пиктограмма элемента меню

The screenshot displays the Eclipse IDE interface for editing a plugin's manifest file. The main window shows the 'All Extensions' section with a list of extensions. The selected extension is 'eclipse2016practicum.toolbars.sampleCommand'. The 'Extension Element Details' panel on the right shows the configuration for this extension, including the 'icon' field set to 'icons/Practicum.png'. A yellow callout bubble with the text 'Пиктограмма' (Icon) points to the 'icon' field. The Package Explorer on the left shows the project structure, including the 'icons' folder containing 'Practicum.png'. The bottom of the image shows a snippet of the plugin's manifest file with the following configuration:

```
File Edit Navigate Search Project CocoR Run EclipsePracticum2016Menu Window Help  
EclipsePracticum2016Command
```


Простейшее расширение. Модуль Eclipse

