

运行代码复审

此教程向您展示一些代码复审特性。这是为软件开发人员编写的。

需要的时间

如果只是简单地浏览一遍此教程，需要大约 **15 分钟**。要使用所提供的样本项目做练习，则需要大约 **30 分钟**。

先决条件

要完成此教程，您应熟悉 **Java** 软件应用程序的开发。如果您了解在 **IBM Rational Software Development Platform** 中如何使用透视图和视图，也会有所帮助。

学习目标

此教程分成若干部分，您应按顺序学习。您将通过阅读而了解自动代码复审的好处，并学会如何执行以下任务：

- 运行代码复审
- 应用所提供的快速修正来解决代码中的问题

如果准备就绪，请开始“代码复审概述”。

代码复审概览

目的

代码复审是一组规则，使软件开发人员或设计人员复审代码的流程自动化。手动代码复审流程可能很耗时且涉及主观意见，而自动代码复审则有效、快速且一致。自动代码复审是手动代码复审的补充。它并不取代手动代码复审。

好处

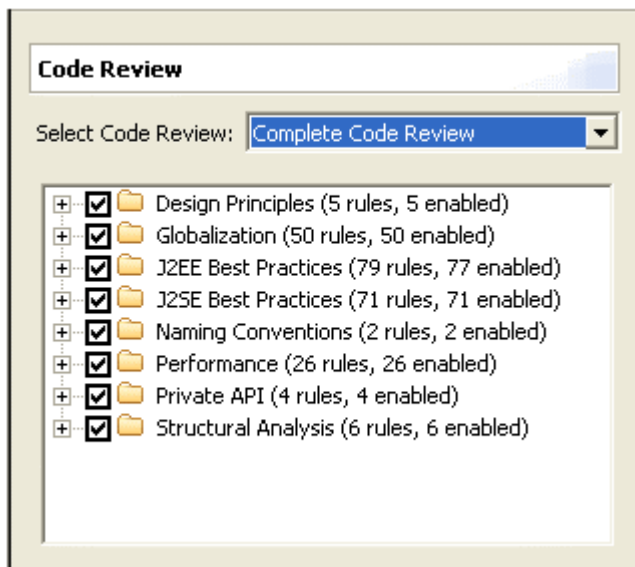
由于自动代码复审工具执行以下任务，因此它在以下几个方面对软件开发流程有价值：

- 查找代码中的错误
- 检查是否符合最佳实践
- 解释每个发现结果并提供解决方案
- 为一些典型“发现结果”提供自动修正
- 允许您创建规则，以确保您在编写代码时遵循应用程序设计标准

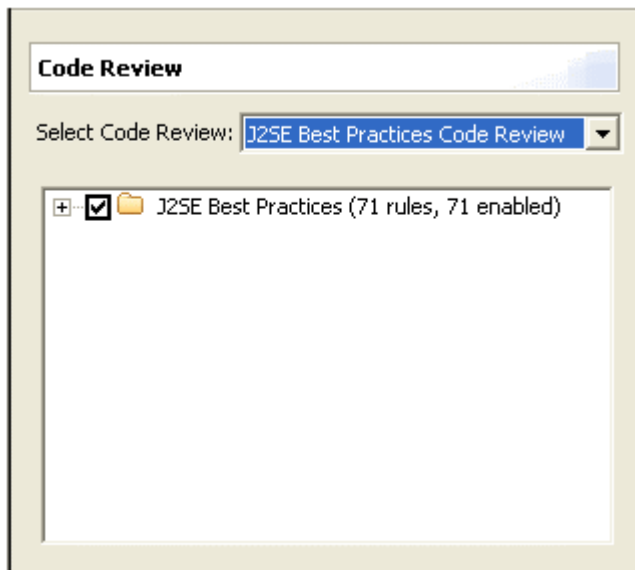
由于自动流程运行很快，因此您可以经常运行代码复审。代码复审发现使您能尽早发现并更正问题，此时进行更改最方便，成本也最低。

提供的代码复审

有若干种代码复审。每种代码复审应用一组不同的规则，这些规则分别组织在不同的文件夹中。根据所处的开发流程阶段以及复审目标，您可以选择满足您需要的代码复审。范围最广的复审是“完整代码复审”，它应用所有类别的规则，如以下屏幕捕捉所示：



一些类别也有与之关联的代码复审。例如，当您仅想应用该类别的规则时，您可以选择“J2SE 最佳实践代码复审”，以下屏幕捕捉显示了这种复审。这使您能运行仅关注特定代码方面的代码复审。






用户定义的代码复审

您可以从提供的向导创建规则。该向导使您能从两种类型的规则中进行选择：体系结构控制和通用。这些规则使软件设计人员能通过创建规则来扩展代码复审的功能，从而确保设计的完整性。

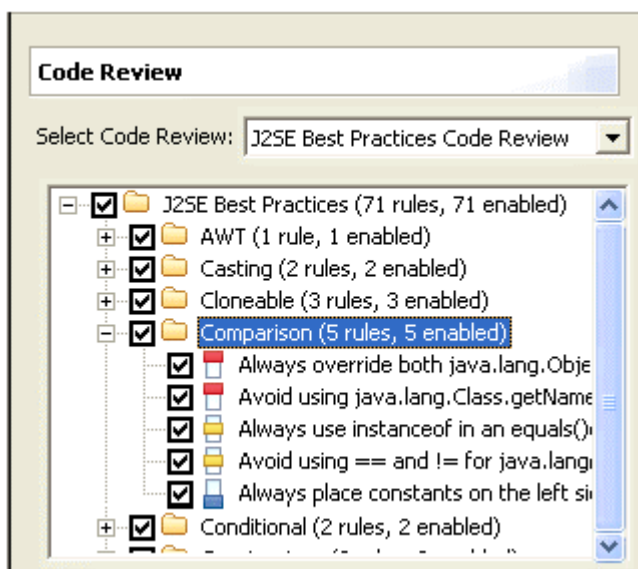
规则的严重性级别

每个规则具有一个严重性级别。对于所提供的规则，您可以修改指定给它的严重性级别。当您从向导创建规则时，要指定一个严重性级别。以下图标表示了三种严重性级别：

- 问题（）：此发现的问题必须解决。
- 警告（）：此发现的问题可能需要解决。
- 建议（）：此发现的问题尚不严重，但强烈建议您现在解决。

尽管“建议”是最低的严重性级别，但请勿低估解决这些发现结果的重要性。它们反映了设计团队应遵守的一组最佳实践和行业标准。即使这些发现的后果不会立刻体现出来，但它们可能导致将来发生问题。

以下屏幕捕捉显示了“J2SE 最佳实践代码复审”的比较文件夹中的规则。该文件夹中的规则包含所有这三个严重性级别。



一些问题的自动修正

一般常见的发现结果具有快速修正，这是工具提供的自动解决办法。如果代码复审中的某个发现结果有快速修正，则该修正会用以下插图中的一个图标指示：



摘要

在软件开发生命周期中，代码复审能自动对代码主体进行对等的复审流程。通过所提供的代码复审，您能运行以下类型的复审：

- 范围很广的完整代码复审，向代码库应用大范围的规则，这些规则来自所有类别
- 范围较窄的特定代码复审，应用一个或多个特定类别（诸如全球化和 / 或设计原理）的规则

您还可以使用所提供的向导来创建自己的规则，这些规则专门用于确保应用程序设计结构的完整性。

由于自动代码复审运行很快，因此您可以尽早检测到代码库中的问题和不一致。这样，您可以在这些问题影响应用程序的维护、可伸缩性和性能之前，尽早解决它们。

现在您准备开始“练习 1.1：导入必需的资源”。

练习 1.1：导入必需的资源

此练习教您如何导入样本项目 `CodeReview_Examples`。使用该样本项目来做“练习 1.2：运行代码复审并应用快速修正”。

解压缩样本项目

此教程的样本项目包含在一个 ZIP 文件中。以下步骤指导您将文件从该 ZIP 文件抽取到工作区文件夹中。

1. 浏览至 ZIP 文件 `CodeReview_Examples` 所在的 `<install_dir>\rad\eclipse\plugins\com.ibm.r2a.rad.tutorial.doc_6.0.0\resources`。
2. 将 `CodeReview_Examples` 抽取到 `<install_dir>\updater\eclipse\workspace`。样本项目文件被抽取到了您的工作区文件夹中，这样您就可以导入它们。

打开“代码复审”视图

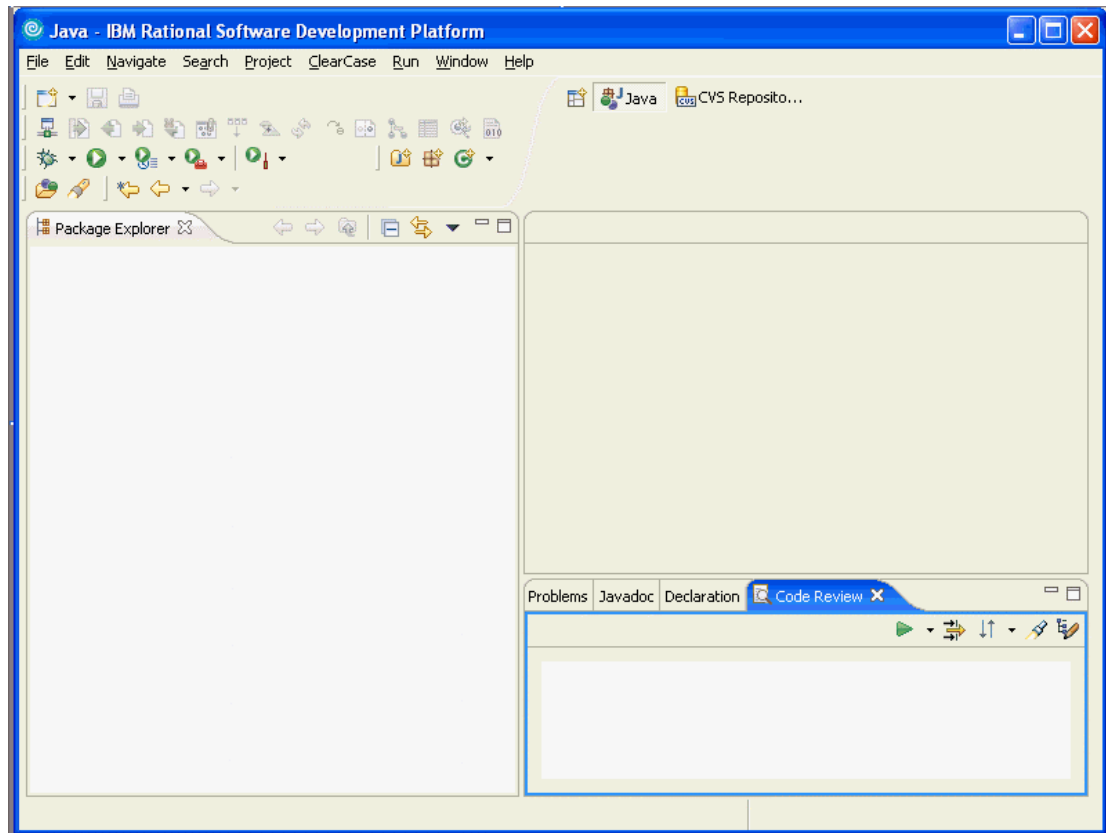
要打开显示“代码复审”视图的透视图：

1. 启动 IBM Rational Software Development Platform 6.0。



2. 单击 **窗口 > 首选项**。
3. 在左窗格中，展开工作合并单击 **功能**。
4. 在 **功能** 列表中，单击 **Java 开发人员**。然后单击 **确定**。
5. 单击 **窗口 > 打开透视图 > Java**。
6. 单击 **窗口 > 显示视图 > 其它 > Java > 代码复审**。
7. 单击 **窗口 > 显示视图 > 其它 > Java > 包资源管理器**。

打开 **Java** 透视图并显示“代码复审”和“包资源管理器”视图后，在以下屏幕捕捉中该透视图显示了两幅视图。您的布局可能不同。也就是，透视图可能在不同位置显示视图。教程使用的是屏幕捕捉中的布局。



导入样本项目

要将样本项目导入到工作区：

1. 右键单击“包资源管理器”视图，以打开弹出菜单。然后单击**导入**来打开“导入”向导。
2. 在**选择**列表中，单击**将现有项目导入到工作区**。然后单击**下一步**。
3. 接下来会看到**项目内容**文本框，单击**浏览**并选择
`<install_dir>\updater\eclipse\workspace\CodeReview_Examples`。
4. 单击**完成**。样本项目及其所有关联的文件将导入到“包资源管理器”中。

开始练习

要开始练习，请转至“练习 1.2：运行代码复审并应用快速修正”。

练习 1.2：运行代码复审并应用快速修正

此练习假设您已完成练习“1.1：导入必需的资源”。在练习中您将先读到一个用户场景。然后，您就担当用户场景中描述的软件开发人员的角色。

用户场景

一大组地理位置分散的开发人员正在为一个新的软件应用程序编码。开发人员例行地运行代码复审以在代码中查找问题，这一点很重要。

其中一个开发人员想要运行代码复审来看看自己总体的工作情况如何。为了复审新编写的代码来评估多个领域中最佳实践的遵守情况，开发人员就运行自动快速代码复审。此复审将所提供的数个类别的规则应用于代码。每个类别的规则都在特定领域（如性能）中检查代码质量。

代码复审完成后，您会看到一系列发现的结果。每个发现结果都代表不严格遵守应用规则的一串代码。这些发现结果中有一个提供了快速修正，所以开发人员应用自动解决方案，立刻更正该问题。

在练习的第一部分，执行以下任务来运行代码复审：

1. 选择要运行的代码复审。
2. 查看该代码复审中应用的规则。
3. 选择对什么代码运行复审。
4. 运行代码复审。
5. 查看代码复审的发现结果。
6. 选择一个发现结果，以查看它的以下信息：
 - 源代码。
 - 描述、示例和解决方案。


接下来，为对代码复审中的一个发现结果应用快速修正，需执行以下任务：

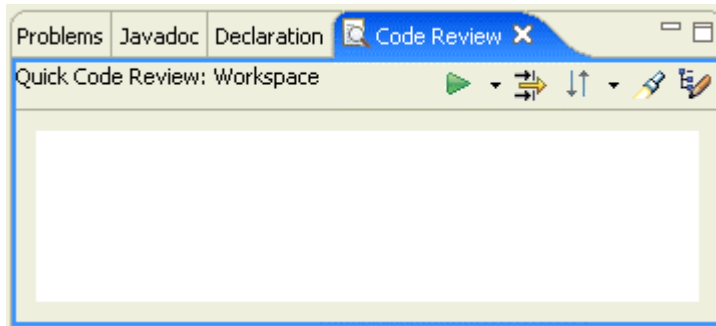
1. 找出发现结果何时快速修正。
2. 查看快速修正将对代码作出的一系列更改。
3. 在应用快速修正之前，预览原始的和重构后的代码。
4. 应用快速修正，重构代码。
5. 应用快速修正后，收到确认。

练习

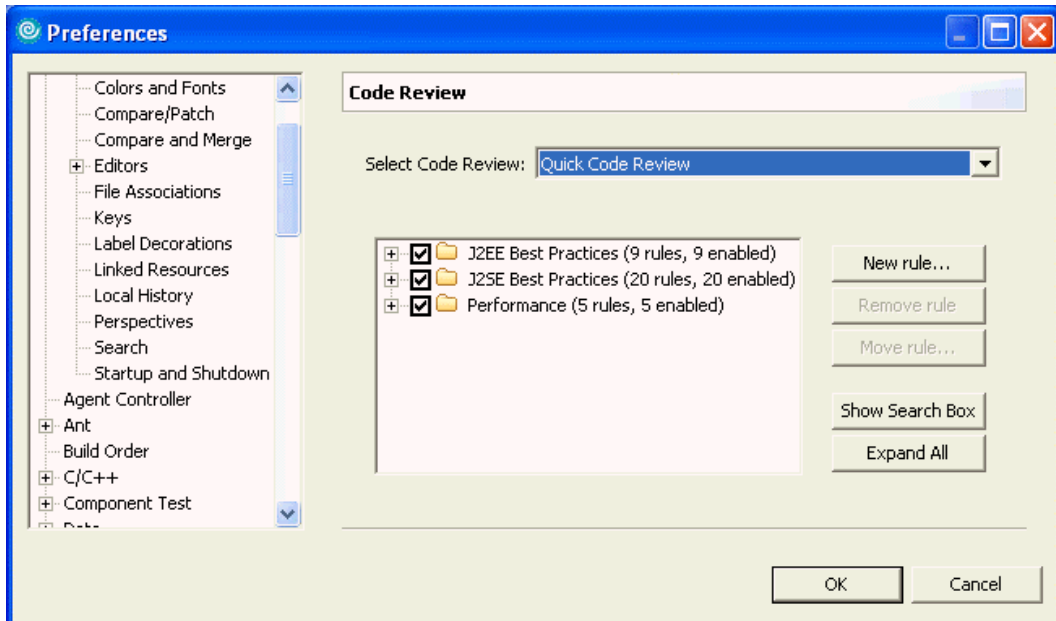
选择代码复审

要选择快速代码复审：

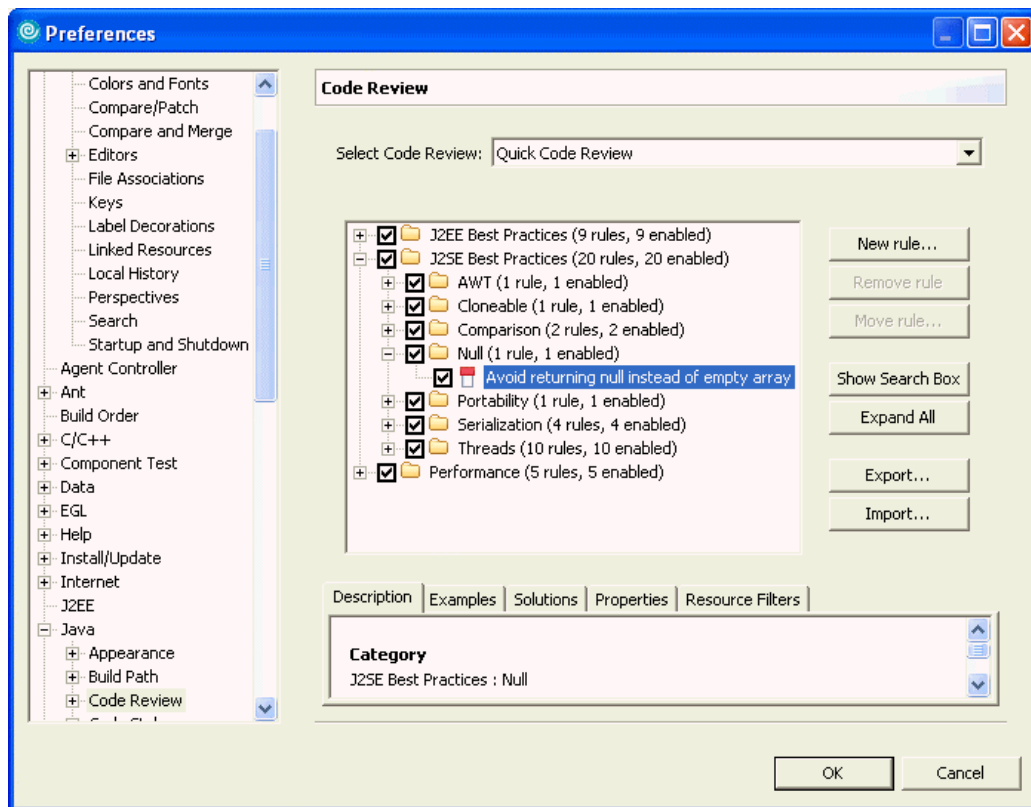
1. 在“代码复审”视图中的工具栏上，单击**管理规则**图标 。



2. 在**选择代码复审**列表中，单击**快速代码复审**。将显示您选择的代码复审的相应规则文件夹，如以下屏幕捕捉中所示：



3. 要查看某个将在代码复审中应用的规则，请展开 **J2SE 最佳实践** 文件夹，然后展开 **Null** 子文件夹。**Null** 文件夹显示一个带问题严重性级别的规则，如以下屏幕捕捉中所示：



作为复审，严重性级别图标显示在以下插图中：

Icon	Severity Level
	Problem
	Warning
	Recommendation

- 单击**确定**来选择“快速代码复审”。

选择要复审的代码库

要选择项目作为要复审的代码库：

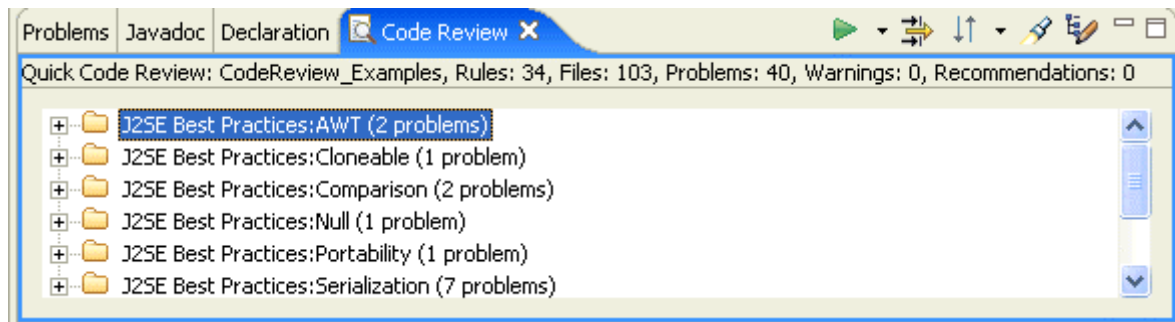
- 在“代码复审”视图中的工具栏上，单击**复审**图标 () > 项目 > **复审 CodeReview_Examples**。

运行代码复审

一旦选择了要复审的代码库，代码复审就会运行。您可以通过检查视图右下角的进度条来跟踪它的状态。

查看代码复审的发现结果

代码复审完成后，发现结果会显示在“代码复审”视图中，如以下屏幕捕捉中所示：



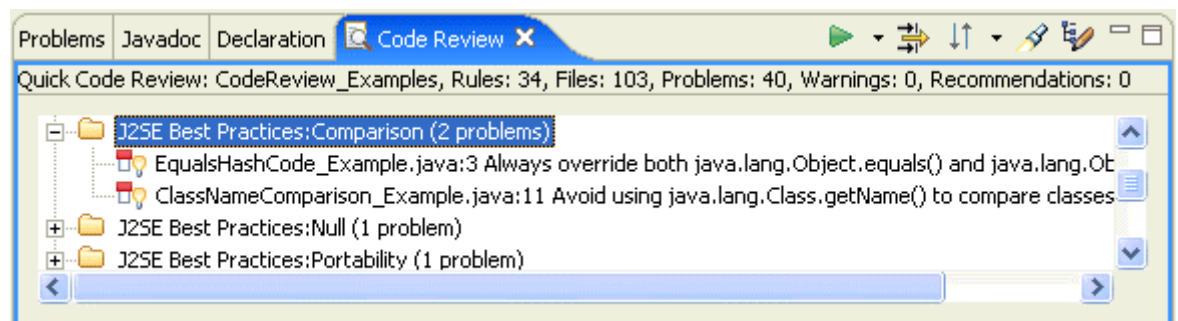
“代码复审”视图提供以下信息：

- 代码复审统计信息：发现结果上面的行显示关于最近的代码复审的信息：类型、范围、所包含的规则数和文件数以及发现结果的数目和严重性。
- 代码复审发现结果：代码复审的发现结果列出在“代码复审”视图中，位于文件夹内。每个文件夹名称都告诉您所应用的规则的类别以及发现结果的个数。

获取更多关于代码复审发现结果的信息

要获取更多关于代码复审的发现结果的信息：

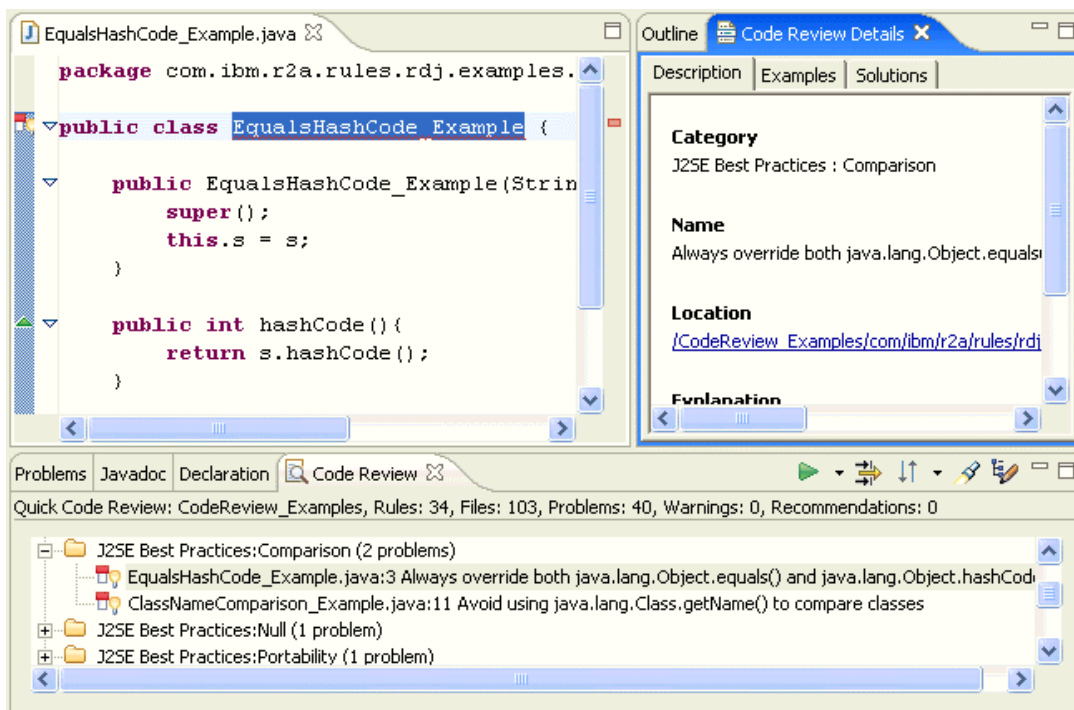
1. 在“代码复审”视图中，滚动到 **J2SE 最佳实践：比较** 文件夹。然后展开文件夹以显示其中的发现结果，如以下的屏幕捕捉中所示：



2. 第一个发现结果以 `EqualsHashCode_Example.java` 开头。后面注出了应用的规则：

始终覆盖 `java.lang.Object.equals()` 和 `java.lang.Object.hashCode()`

3. 双击第一个发现结果。关于它的详细信息出现在两个地方，如以下各点和屏幕捕捉所概述：
 - 源代码：显示发现结果发生的地方，并突出显示其精确位置。
 - “代码复审详细信息”视图：更详细地描述了发现结果，并提供更正它的示例和解决办法。

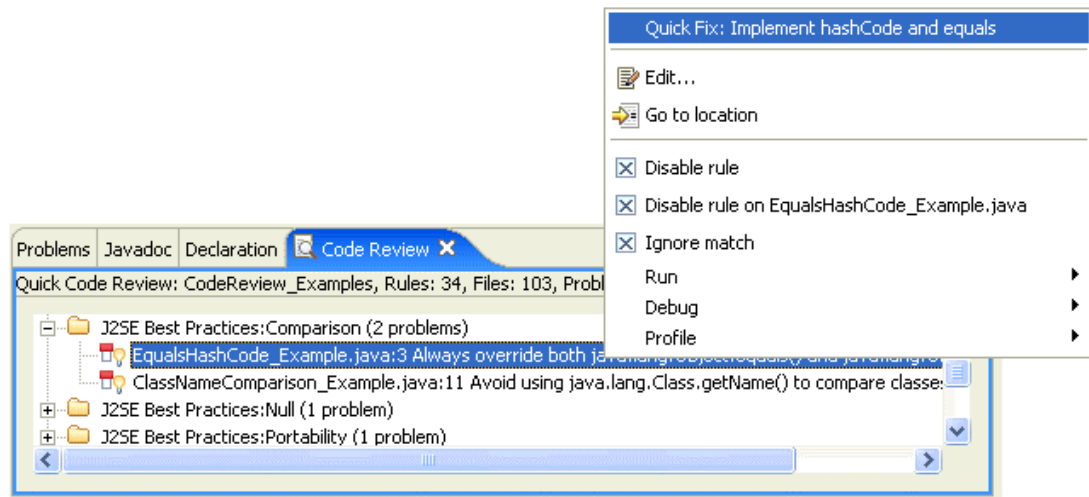


选择有快速修正的发现结果

您可以看出“最佳实践：比较”文件夹中的两个发现结果都有以图标标示的快速修正。作为复审，快速修正图标显示在以下插图中：



1. 右键单击列表中的第一个发现结果，如下一个屏幕捕捉所示。
2. 快速修正弹出菜单选项因解决方案而异。对于您选择的发现结果，修正是实施 `hashCode` 和 `equals`。



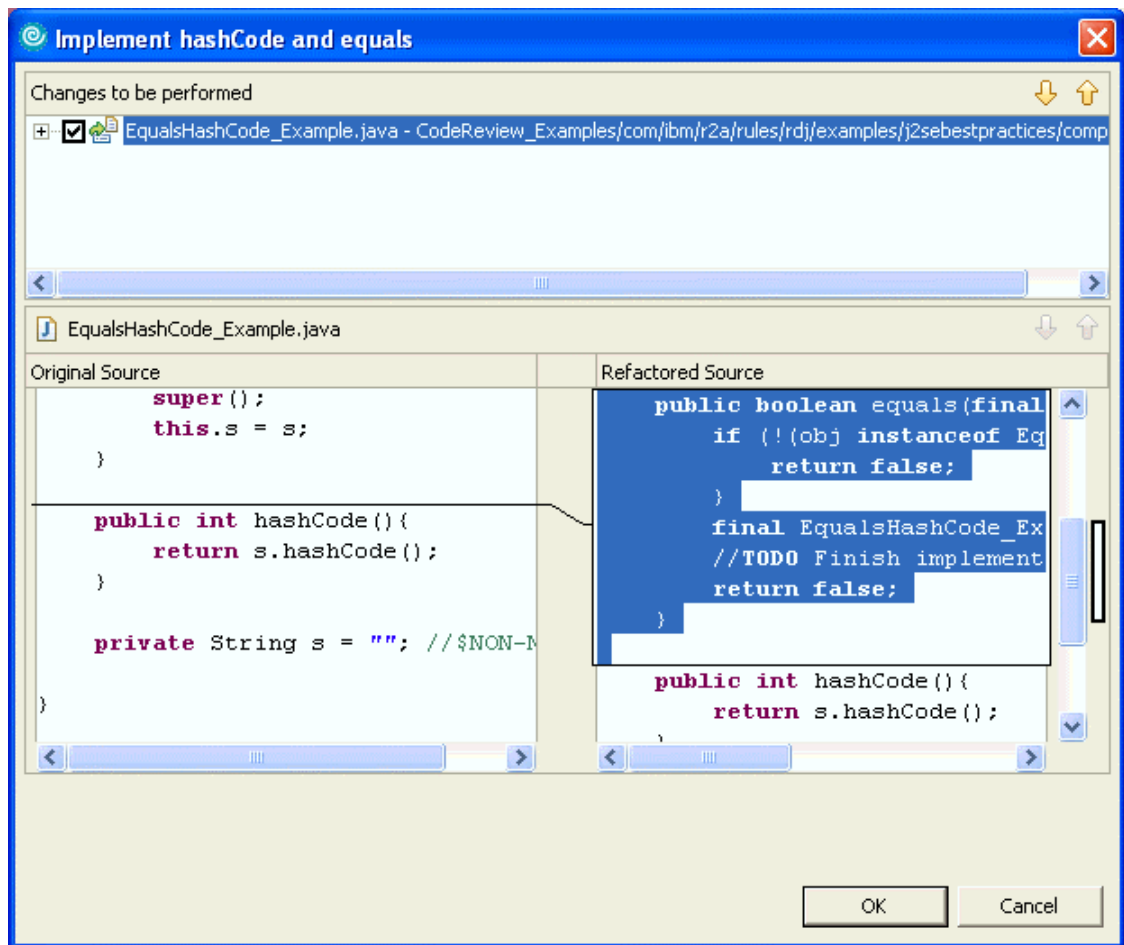
3. 单击**快速修正：实施 hashCode 和 equals**。

应用快速修正

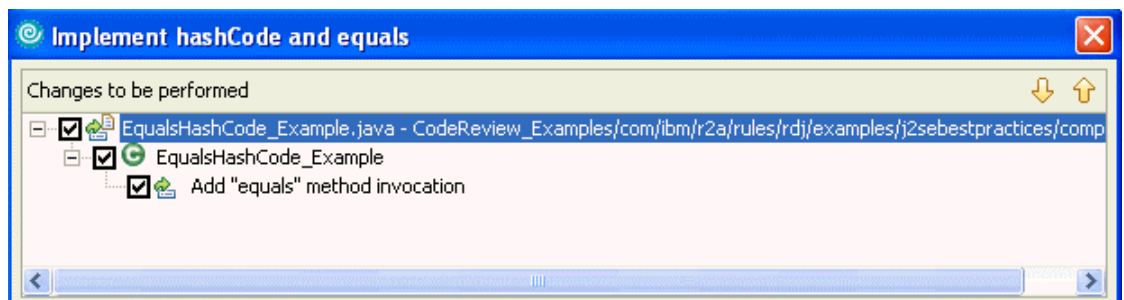
对于您选择的发现结果，适用的快速修正是实施 **hashCode** 和 **equals**。

要复审并将快速修正应用于发现结果：

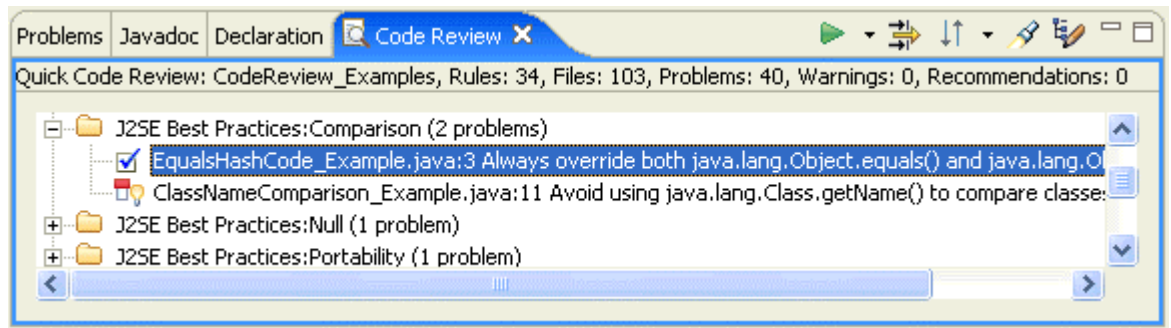
1. 您将看到代码的并排视图，如以下屏幕捕捉所示。原始的源代码在左边，将由快速修正创建的重构源代码在右边。如果您决定应用快速修正，它将追加突出显示的缺失代码行。



2. 在**要执行的更改**部分中展开列表，查看快速修正确切地将更改什么以及如何更改，如以下屏幕捕捉所示：



3. 复审列表中的更改。然后单击**确定**，将快速修正应用于列表中所有选中的更改。
4. 应用快速修正后，您将看到您解决的发现结果旁有一个选中标记。



在选中标记旁，您将看到以下信息：

- 您应用的快速修正
- 发现结果所在的源代码行号
- 代码复审中未遵守的规则

您已经完成了“练习：运行代码复审并应用快速修正”。

练习总结

您已经执行了“练习：运行代码复审并应用快速修正”中的所有任务。

运行代码复审的任务

当您运行此代码复审时，您执行了以下任务：

1. 选择了要运行的代码复审。
2. 查看了代码复审中应用的规则。
3. 选择了对什么代码主体运行复审。
4. 运行了代码复审。
5. 查看了代码复审的发现结果。
6. 选择了一个发现结果，以查看它的以下信息：
 - 源代码。
 - 描述、示例和解决方案。

应用快速修正的任务

当您应用快速修正时，您执行了接下来的一组任务：

1. 找出了发现结果何时快速修正。
2. 查看了快速修正将对代码作出的一系列更改。
3. 预览了原始的和重构后的代码。
4. 应用了快速修正，以重构代码。
5. 收到了已应用快速修正的确认。

充分利用代码复审的功能

通过主动运行代码复审，您可以在早期分析发现结果。这意味着您还可以在早期处理它们，避免它们造成以下问题：

- 影响应用程序的性能、维护或可伸缩性
- 浪费公司的金钱、时间和资源

充分利用快速修正的能力

通过应用提供的快速修正，您就有了解决常见发现结果的自动方法。快速修正在以下方面为您提供帮助：

- 每次都一致地更正问题
- 让您腾出时间研究代码，而花费较少的时间来修正错误

通过复习“摘要：运行代码复审”中的学习目标，完成教程。

摘要：运行代码复审

此教程向您展示如何运行代码复审。

完成的学习目标

如果完成了此练习，您现在就应能执行以下任务：

- 运行代码复审。
- 应用所提供的快速修正来解决问题。

更多信息

如果想要了解更多关于此教程中所涵盖主题的信息，请参阅运行代码复审的联机帮助。