Ausführen einer Codeprüfung

Dieses Lernprogramm informiert Sie über einige Funktionen der Codeprüfung. Es richtet sich in erster Linie an Softwareentwickler.

Erforderliche Zeit

Zum reinen Durchlesen dieses Lernprogramms benötigen Sie rund **15 Minuten**. Zum Durcharbeiten der Übung anhand des Beispielprojekts benötigen Sie rund **30 Minuten**.

Voraussetzungen

Um dieses Lernprogramm durcharbeiten zu können, müssen Sie mit der Entwicklung von Java-Softwareanwendungen vertraut sein. Außerdem ist es hilfreich, wenn Sie die Perspektiven und Ansichten der IBM Rational Softwareentwicklungsplattform verstehen.

Lernziele

Dieses Lernprogramm ist in verschiedene Abschnitte untergliedert, die Sie nacheinander durcharbeiten sollten. Sie erhalten u. a. Informationen zu den Vorteilen automatisierter Codeprüfungen und zur Vorgehensweise beim Ausführen der folgenden Aufgaben:

- Ausführen einer Codeprüfung
- Anwenden eines Quick-Fixes zur Korrektur eines Codefehlers

Beginnen Sie mit dem Abschnitt “Structural Analysis - Übersicht”.

Überblick über die Codeprüfung

Zweck

Unter Codeprüfung versteht man eine Gruppe von Regeln, die für einen Softwareentwickler oder -architekten den Prozess zur Prüfung von Code automatisiert. Im Gegensatz zur manuellen Codeprüfung, die sich zeitaufwendig gestalten und mit subjektiv gefärbten Diskussionen durchsetzt sein kann, ist die automatisierte Codeprüfung effizient, schnell und konsistent. Sie ergänzt die manuelle Codeprüfung, ersetzt diese jedoch nicht.

Vorteile

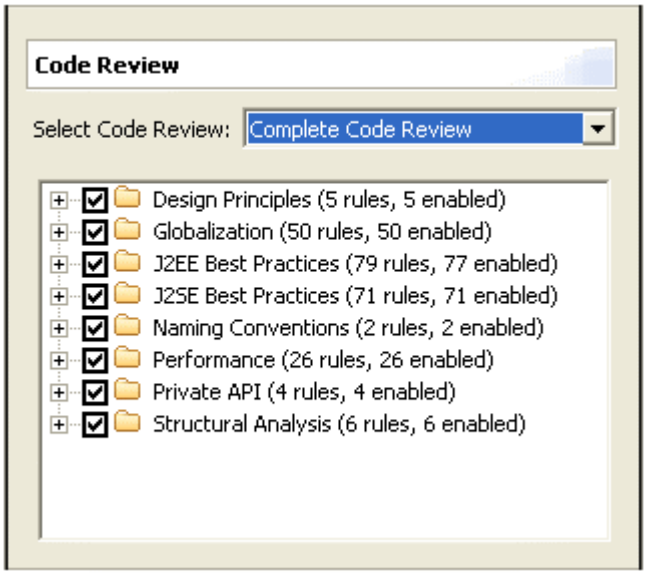
Das Tool für die automatisierte Codeprüfung wirkt sich auf verschiedene Weise vorteilhaft auf die Softwareentwicklung aus, weil es die folgenden Aufgaben übernimmt:

- Suchen von Fehlern im Code
- Überprüfung der Einhaltung von Best Practices
- Erläuterung der einzelnen Fehler und Bereitstellung einer entsprechenden Lösung
- Bereitstellung einer automatisierten Korrektur für eine Reihe typischer Fehler
- Bereithalten von Möglichkeiten zur Erstellung von Regeln, damit das Anwendungsdesign und die Standards beim Schreiben von Code eingehalten werden

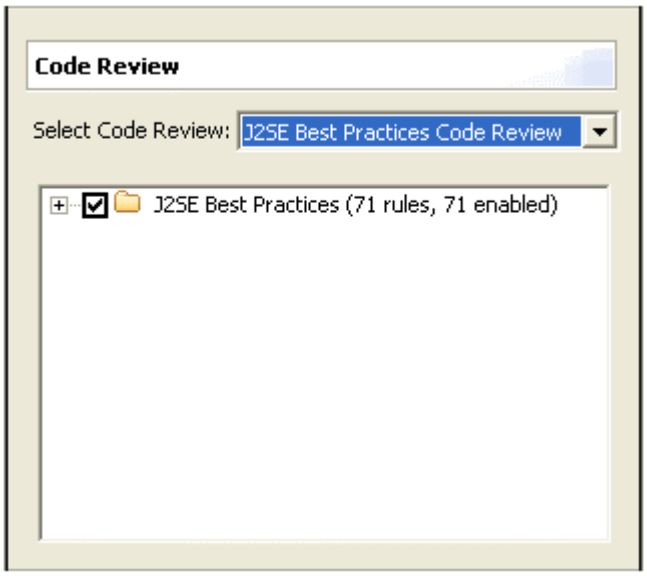
Da der automatisierte Prüfprozess sehr schnell abläuft, können Sie Codeprüfungen häufig ausführen. Mit Hilfe der bei der Codeprüfung gefundenen Fehler lassen sich Probleme bereits in einem frühen Stadium des Entwicklungsprozesses, in dem Änderungen noch einfach und kostengünstig vorgenommen werden können, erkennen und beheben.

Bereitgestellte Codeprüfungen

Eine Reihe von Codeprüfungen ist bereits Bestandteil des Lieferumfangs. Bei jeder Codeprüfung wird eine andere Gruppe von Regeln angewendet. Diese sind in Ordnern zusammengefasst. Abhängig davon, in welchem Stadium des Entwicklungsprozesses Sie sich befinden und welchen Zweck Sie mit der Prüfung verfolgen, können Sie so die jeweils passende Codeprüfung auswählen. Die "Vollständige Codeprüfung" stellt dabei die umfassendste Prüfung dar, bei der Regeln aus allen Kategorien angewendet werden (siehe folgenden Screenshot):



Einigen Kategorien ist außerdem eine Codeprüfung zugeordnet. Sie können beispielsweise die Codeprüfung "J2SE Best Practices" auswählen, wenn ausschließlich Regeln für diese Kategorie angewendet werden sollen (siehe folgenden Screenshot). Auf diese Weise können Sie Codeprüfungen ausführen, die sich auf einen speziellen Aspekt des Codes konzentrieren.



Benutzerdefinierte Codeprüfungen

Sie können auch Regeln mit Hilfe eines bereitgestellten Assistenten erstellen. Der Assistent bietet die Möglichkeit, zwischen zwei Regelarten zu wählen: architektonische Steuerelemente und allgemeine Regeln. Mit Hilfe dieser Regeln können Softwarearchitekten die Funktionen der Codeprüfung erweitern, indem sie Regeln erstellen, die die Integrität ihres Designs gewährleisten.

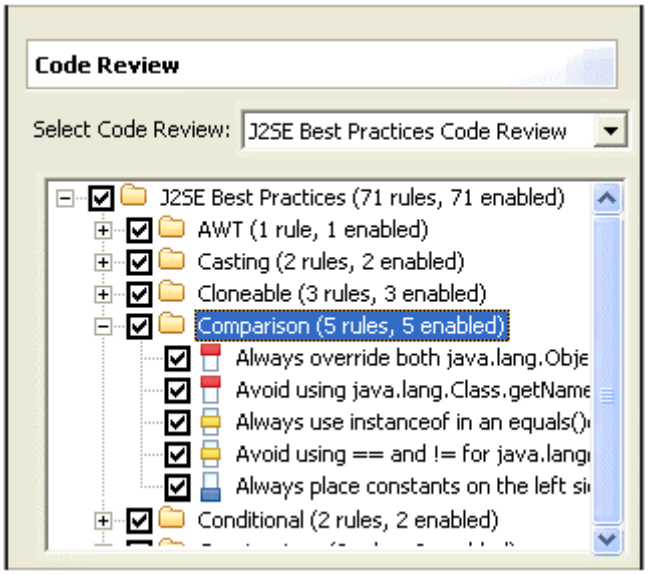
Wertigkeit von Regeln

Jede Regel besitzt eine Wertigkeit. Bei einer bereitgestellten Regel können Sie die ihr zugeordnete Wertigkeit ändern. Wenn Sie eine Regel im Assistenten erstellen, geben Sie die Wertigkeit an. Die drei Wertigkeiten werden durch die folgenden Symbole dargestellt:

- Problem ( ): Dieser Fehler muss korrigiert werden.
- Warnung ( ): Hierbei handelt es sich wahrscheinlich um einen Fehler, der korrigiert werden muss.
- Empfehlung ( ): Dieser Fehler ist noch nicht schwer wiegend, aber es wird empfohlen, ihn bereits zu diesem Zeitpunkt zu korrigieren.

Obwohl es sich bei "Empfehlung" um die niedrigste Wertigkeit handelt, können Sie daraus nicht unbedingt schließen, wie wichtig die Behebung solcher Probleme tatsächlich ist. Die Empfehlungen stellen eine Gruppe bewährter Verfahren und Branchenstandards dar, die Entwicklungsteams einhalten sollten. Auch wenn es sich hierbei nicht um dringend zu korrigierende Fehler handelt, können sie dennoch zu einem späteren Zeitpunkt zu Problemen führen.

Der folgende Screenshot zeigt Regeln im Ordner "Comparison" der Codeprüfung "J2SE Best Practices". Regeln in diesem Ordner besitzen alle drei Wertigkeiten.



Automatisierte Korrekturen für eine Reihe von Problemen

Für eine Reihe allgemeiner Fehler steht ein Quick-Fix zur Verfügung, der als automatisierte Lösung bereitgestellt wird. Steht für einen bei der Codeprüfung gefundenen Fehler ein Quick-Fix zur Verfügung, wird dies durch eines der folgenden Symbole kenntlich gemacht (siehe folgende Abbildung):

Zusammenfassung

Bei der Softwareentwicklung wird der Peer-Prüfprozess von Code durch die Codeprüfung automatisiert. Mit Hilfe der bereitgestellten Codeprüfungen können Sie die folgenden Prüfungen ausführen:

- Umfangreiche, komplette Codeprüfungen, bei denen eine große Bandbreite von Regeln aus allen Kategorien auf eine Codebasis angewendet wird
- Begrenzte, zielgerichtete Codeprüfungen, bei denen Regeln aus einer oder aus speziellen Kategorien wie Globalisierung und/oder Designprinzipien angewendet werden

Mit Hilfe des bereitgestellten Assistenten können Sie auch Ihre eigenen Regeln erstellen, die speziell die Integrität der Struktur Ihrer Anwendung gewährleisten.

Da automatisierte Codeprüfungen nur wenig Zeit in Anspruch nehmen, können Sie Probleme und Inkonsistenzen in einer Codebasis bereits in einem frühen Stadium erkennen. Entsprechend können Sie diese dann frühzeitig beheben, bevor Sie Auswirkungen auf Wartung, Skalierbarkeit oder Leistung Ihrer Anwendung haben.

Sie können nun mit “Übung 1.1: Erforderliche Ressourcen importieren” beginnen.

Übung 1.1: Erforderliche Ressourcen importieren

In dieser Übung erfahren Sie, wie Sie das Beispielprojekt CodeReview_Examples importieren. Sie benötigen dieses Projekt zur Ausführung von “Übung 1.2: Codeprüfung ausführen und Quick-Fix anwenden”.

Beispielprojekt dekomprimieren

Das Beispielprojekt für dieses Lernprogramm befindet sich in einer komprimierten Datei. Mit Hilfe der folgenden Schritte extrahieren Sie Dateien aus dieser komprimierten Datei in Ihren Arbeitsbereichsordner.

1. Navigieren Sie zu <*Installationsverzeichnis*>\rad\eclipse\plugins\com.ibm.r2a.rad.tutorial.doc_6.0.0\resources, wo sich die komprimierte Datei, CodeReview_Examples, befindet.
2. Extrahieren Sie CodeReview_Examples in <*Installationsverzeichnis*>\updater\eclipse\workspace. Die Dateien des Beispielprojekts werden in den Arbeitsbereichsordner extrahiert, so dass Sie diese nun importieren können.

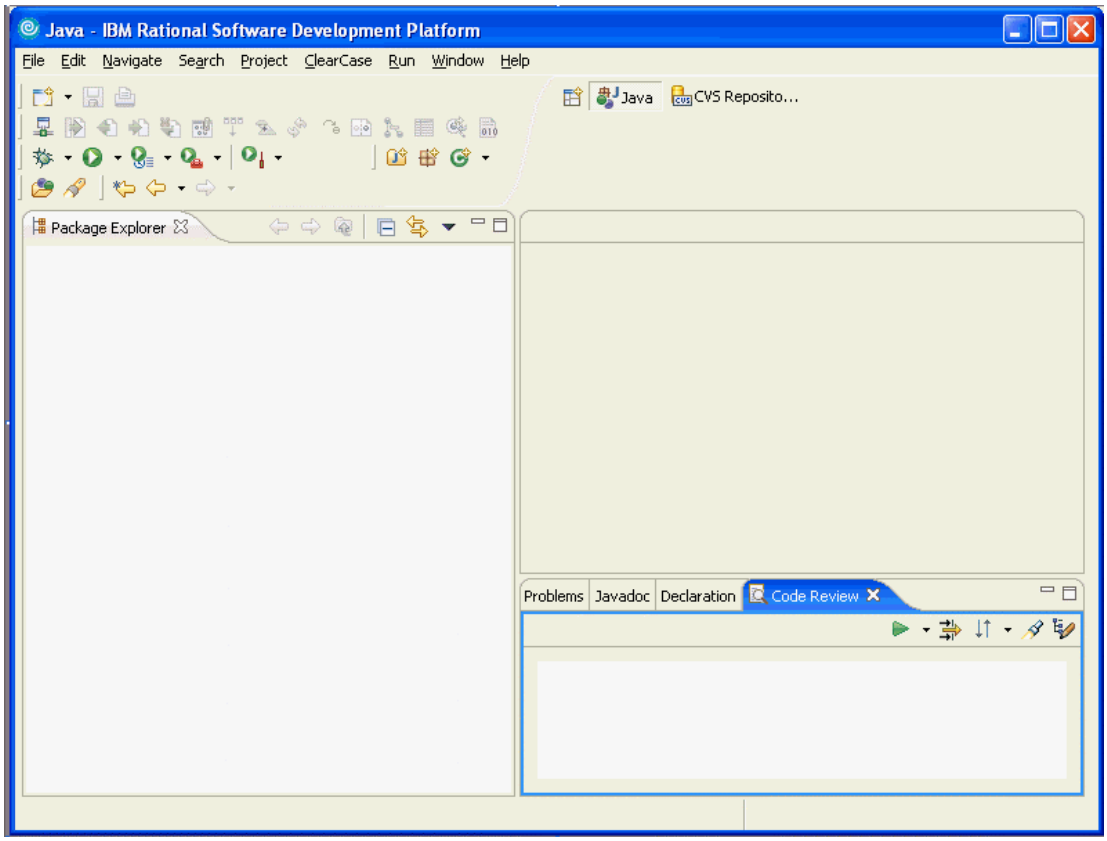
Ansicht "Code Review" öffnen

So öffnen Sie eine Perspektive, die die Ansicht "Code Review" anzeigt:

1. Starten Sie IBM Rational Software Development Platform 6.0.

2. Klicken Sie auf **Window > Preferences**.
3. Erweitern Sie **Workbench** im linken Teilfenster und klicken Sie auf **Capabilities**.
4. Klicken Sie in der Liste **Capabilities** auf **Java Developer**. Klicken Sie dann auf **OK**.
5. Klicken Sie auf **Window > Open Perspective > Java**.
6. Klicken Sie auf **Window > Show View > Other > Java > Code Review**.
7. Klicken Sie auf **Window > Show View > Other > Java > Package Explorer**.

   Nach dem Öffnen der Java-Perspektive und dem Anzeigen der Ansichten "Code Review" und "Package Explorer" werden in der Perspektive die im folgenden Screenshot dargestellten Ansichten angezeigt. Ihr Layout kann davon abweichen. Die Ansichten werden in der Perspektive u. U. an anderen Positionen anzeigt. Das Lernprogramm verwendet das im Screenshot dargestellte Layout.



Beispielprojekt importieren

So importieren Sie das Beispielprojekt in den Arbeitsbereich:

1. Klicken Sie mit der rechten Maustaste in der Ansicht "Package Explorer", um das Popup-Menü zu öffnen. Klicken Sie dann auf **Import**, um den Assistenten für den Import zu öffnen.
2. Klicken Sie in der Liste **Select** auf **Existing Project into Workspace**. Klicken Sie dann auf **Next**.
3. Klicken Sie neben dem Textfeld **Project contents** auf **Browse** und wählen Sie <*Installationsverzeichnis*>\updater\eclipse\workspace\CodeReview_Examples aus.
4. Klicken Sie auf **Finish**. Das Beispielprojekt und alle ihm zugeordneten Dateien werden in die Ansicht "Package Explorer" importiert.

Übung beginnen

Wechseln Sie zu “Übung 1.2: Codeprüfung ausführen und Quick-Fix anwenden”.

Übung 1.2: Codeprüfung ausführen und Quick-Fix anwenden

Für diese Übung wird vorausgesetzt, dass Sie bereits “Übung 1.1: Erforderliche Ressourcen importieren ” ausgeführt haben. Lesen Sie zunächst das Benutzerszenario. Versetzen Sie sich anschließend in die Rolle des im Szenario beschriebenen Softwareentwicklers.

Benutzerszenario

Eine große Gruppe geographisch verteilter Entwickler schreibt den Code für eine neue Softwareanwendung. Es ist erforderlich, dass die Entwickler routinemäßig Codeprüfungen ausführen, um ihren Code auf Fehler zu überprüfen.

Ein Entwickler möchte eine Codeprüfung ausführen, um seine Arbeit allgemein zu beurteilen. Um zu prüfen, ob für neu geschriebenen Code in mehreren Bereichen die Einhaltung der Best Practices gewährleistet ist, führt er eine schnelle Codeprüfung aus. Bei dieser Prüfung werden mehrere Regelkategorien auf den Code angewendet. Jede Regelkategorie überprüft die Qualität des Codes in einem bestimmten Bereich, z. B. im Bereich Leistung.

Nach Abschluss der Codeprüfung erhalten Sie eine Liste der gefundenen Fehler. Jeder Fehler steht für eine Codezeichenfolge, bei der eine angewendete Regel nicht streng eingehalten wurde. Für einen der gefundenen Fehler steht ein Quick-Fix zur Verfügung. So kann der Entwickler die automatisierte Lösung anwenden und den Fehler umgehend korrigieren.

Im ersten Teil der Übung führen Sie im Rahmen der Codeprüfung die folgenden Aufgaben aus:

1. Auswählen einer Codeprüfung
2. Anzeigen der in der Codeprüfung angewendeten Regeln
3. Wählen des Codes, für den die Prüfung ausgeführt werden soll
4. Ausführen der Codeprüfung
5. Anzeigen der bei der Codeprüfung gefundenen Fehler
6. Auswählen eines Fehlers, um die folgenden Informationen dazu anzuzeigen:
    - Quellcode
    - Beschreibung, Beispiele und Lösungen

Anschließend führen Sie die folgenden Aufgaben aus, um einen Quick-Fix auf einen der bei der Codeprüfung gefundenen Fehler anzuwenden:

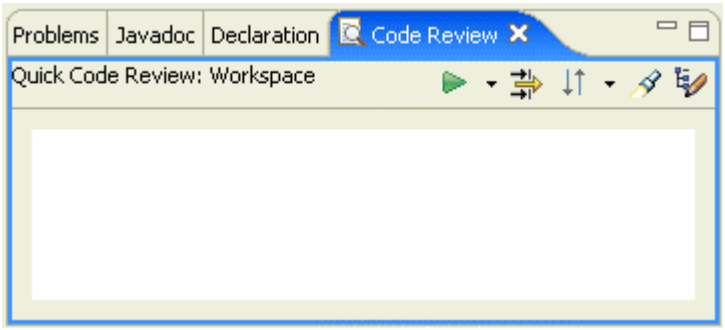
1. Erkennen, wenn ein Quick-Fix für einen gefundenen Fehler zur Verfügung steht
2. Anzeigen einer Liste mit Änderungen, die durch den Quick-Fix am Code vorgenommen werden
3. Voranzeige des ursprünglichen und des umstrukturierten Codes vor der Anwendung des Quick-Fixes
4. Anwenden des Quick-Fixes zur Umstrukturierung (Refactoring) des Codes
5. Abrufen einer Bestätigung nach der Anwendung des Quick-Fixes

Übung

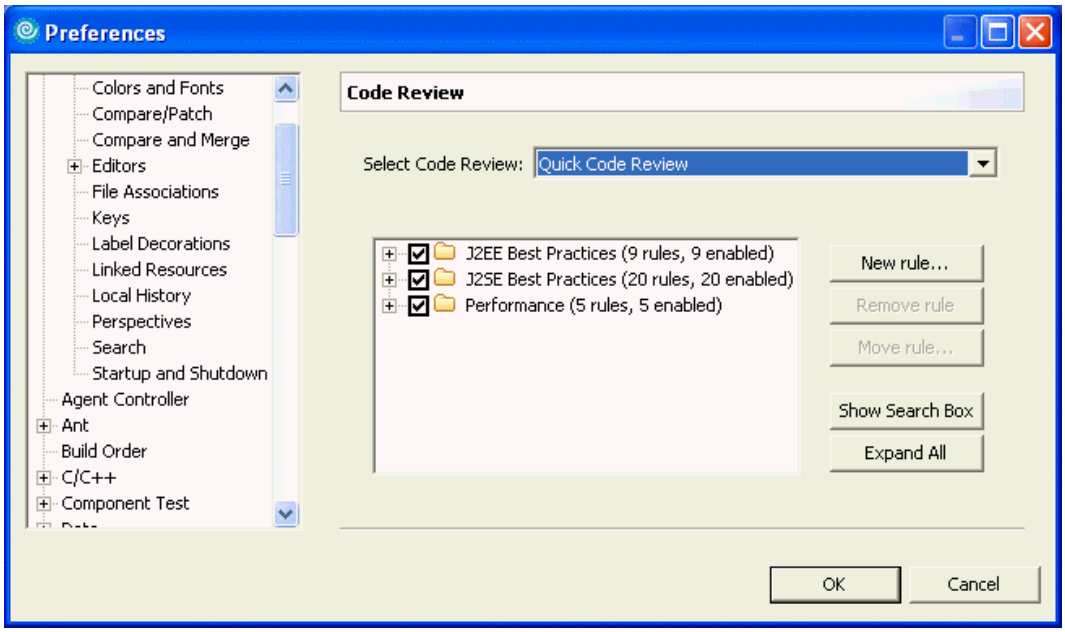
Codeprüfung auswählen

So wählen Sie eine Codeprüfung aus:

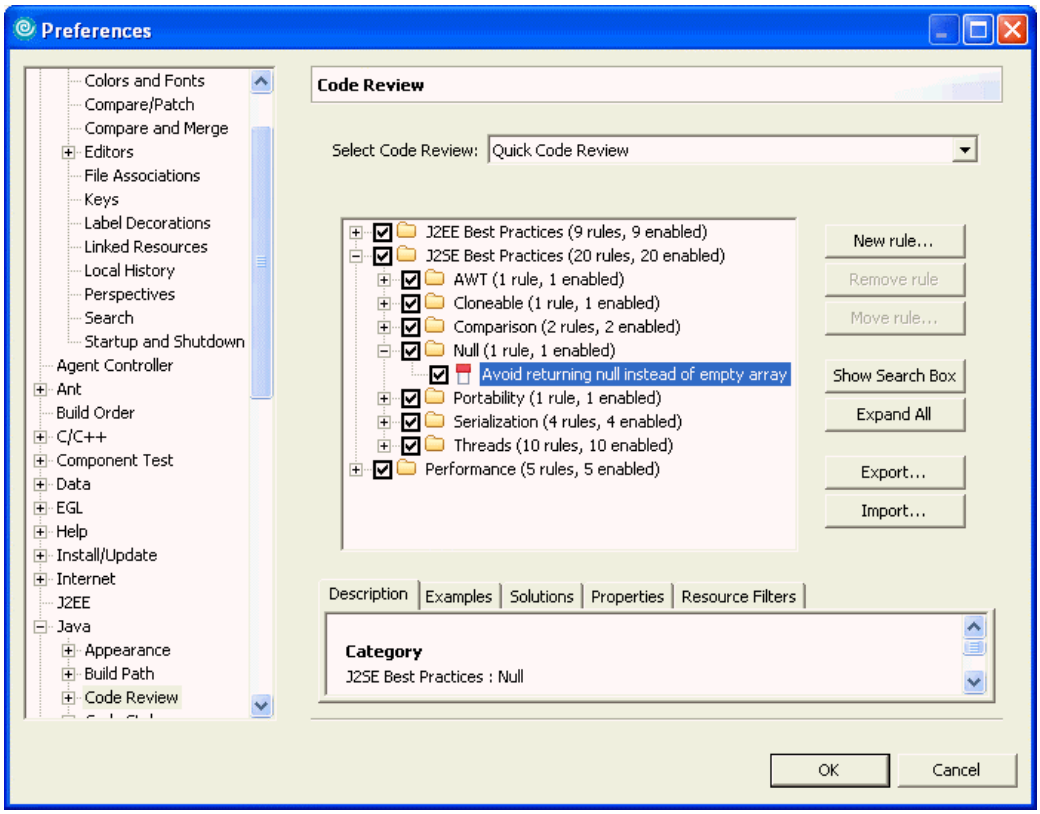
1. Klicken Sie in der Funktionsleiste in der Ansicht "Code Review" auf das Symbol für **Regeln verwalten**.



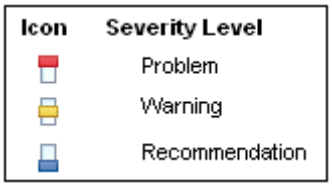
2. Klicken Sie in der Liste **Select Code Review** auf den Eintrag **Quick Code Review**. Die Ordner mit den Regeln für die von Ihnen ausgewählte Codeprüfung werden angezeigt (siehe folgenden Screenshot):



3. Zum Anzeigen einer der Regeln, die bei der Codeprüfung angewendet werden, erweitern Sie den Ordner **J2SE Best Practices** und anschließend den Unterordner **Null**. Ordner "Null" zeigt eine Regel mit einer Fehlerwertigkeit an (siehe folgenden Screenshot):



Zur Verdeutlichung enthält die folgende Abbildung eine Übersicht über die Symbole (Icon) für die Wertigkeit (Severity Level):



4. Klicken Sie auf **OK**, um die Option "Quick Code Review" auszuwählen.

Codebasis für die Prüfung auswählen

So wählen Sie das Projekt als Codebasis für die Prüfung aus:

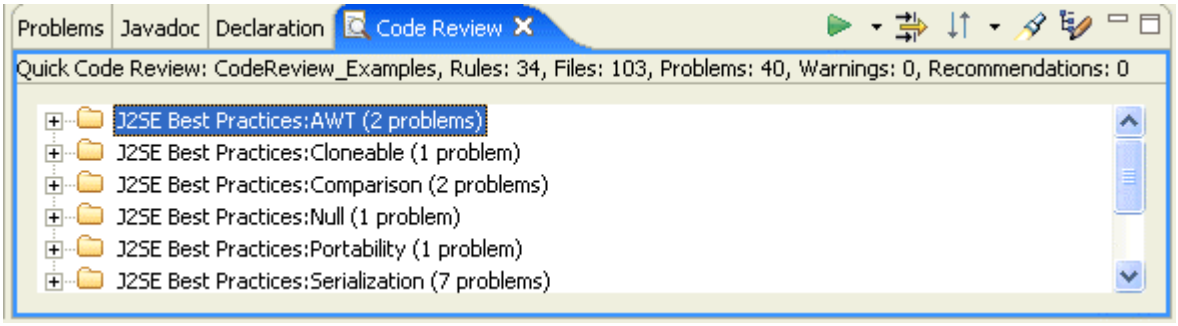
- Klicken Sie in der Funktionsleiste in der Ansicht "Code Review" auf das Symbol für **Prüfung** > **Projekte** > **CodeReview_Examples prüfen**.

Codeprüfung ausführen

Nach Auswahl der Codebasis für die Prüfung wird die Codeprüfung ausgeführt. Sie können den Status der Prüfung anhand des Fortschrittsanzeigers in der unteren rechten Ecke der Ansicht verfolgen.

Bei der Codeprüfung gefundene Fehler anzeigen

Nach Abschluss der Codeprüfung werden die gefundenen Fehler in der Ansicht "Code Review" angezeigt (siehe folgenden Screenshot):



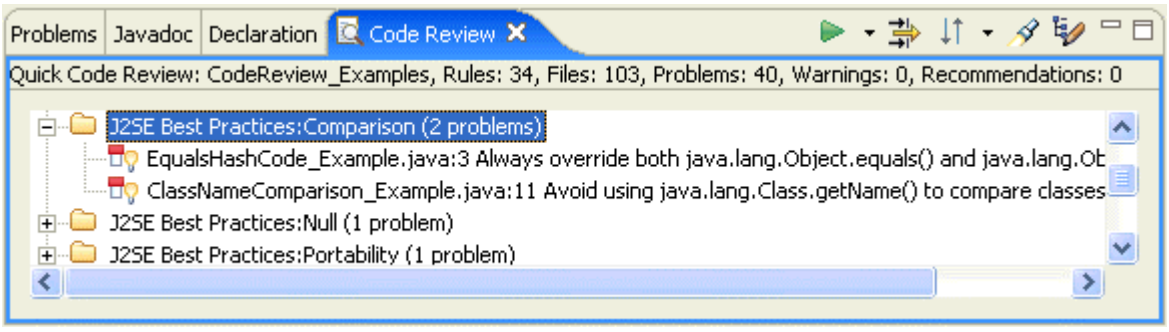
Bei der Codeprüfung erhalten Sie die folgenden Informationen:

- Statistik zur Codeprüfung: Die Zeile oberhalb der gefundenen Fehler enthält Informationen über die zuletzt ausgeführte Codeprüfung: Typ, Umfang, Anzahl der Regeln und geprüften Dateien sowie Anzahl und Wertigkeit der Fehler.
- Bei der Codeprüfung gefundene Fehler: Die bei der Codeprüfung gefundenen Fehler werden in der Ansicht "Code Review" in Ordnern aufgelistet. Jeder Ordnername gibt Auskunft über die Regelkategorien und die Anzahl der Fehler.

Weitere Informationen zu einem bei der Codeprüfung gefundenen Fehler abrufen

So rufen Sie weitere Informationen zu einem bei der Codeprüfung gefundenen Fehler ab:

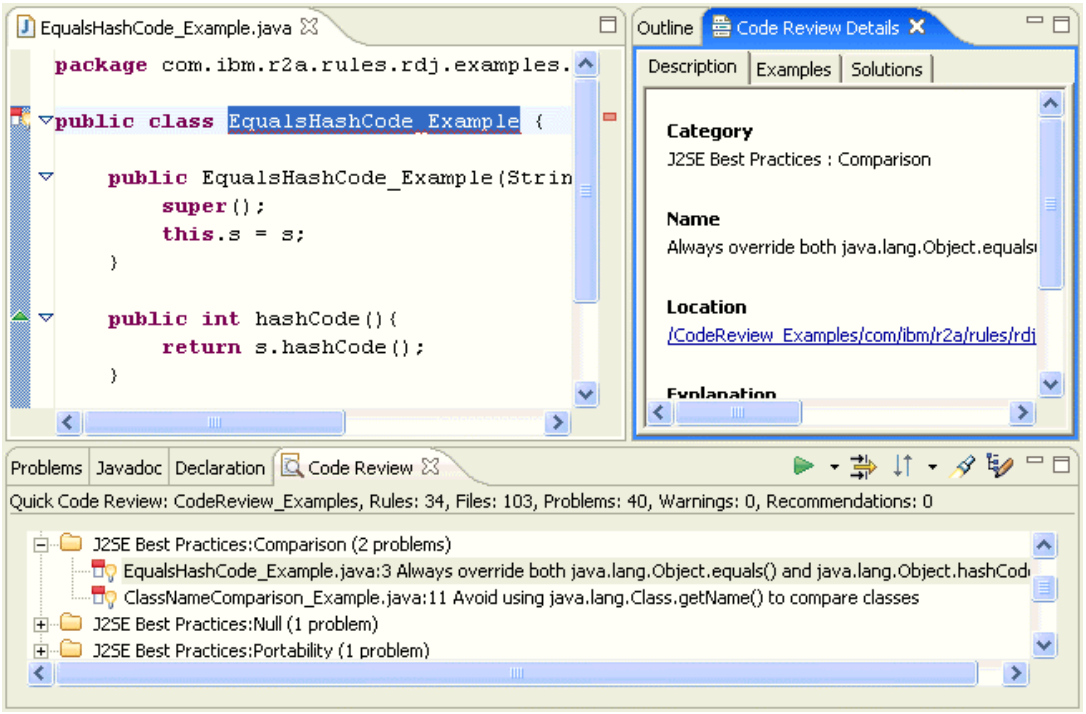
1. Blättern Sie in der Ansicht "Code Review" zum Ordner **J2SE Best Practices: Comparison**. Erweitern Sie anschließend den Ordner, um die darin enthaltenen Fehler anzuzeigen (siehe folgenden Screenshot):



2. Der erste gefundene Fehler beginnt mit "EqualsHashCode_Example.java". Anschließend wird die angewendete Regel aufgeführt:

   Überschreiben Sie immer sowohl "java.lang.Object.equals()" als auch "java.lang.Object.hashCode()".

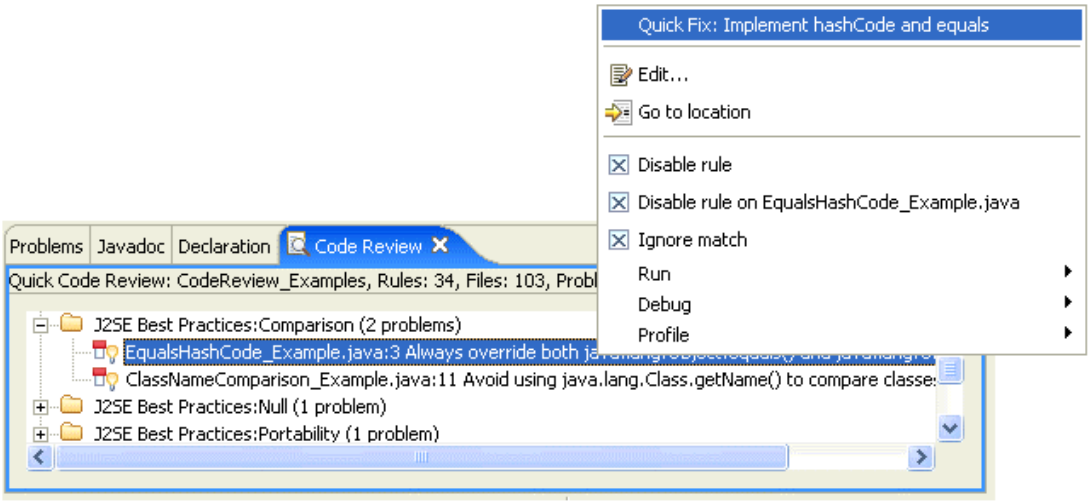
3. Klicken Sie doppelt auf den ersten Fehler. Details dazu werden an zwei Stellen angezeigt, wie in den folgenden Punkten und im Screenshot dargestellt:
    - Quellcode: Zeigt den Code an, wo der Fehler auftritt, und hebt die genaue Position hervor.
    - Ansicht "Code Review Details": Liefert eine detailliertere Fehlerbeschreibung und stellt Beispiele und Lösungen zur Korrektur bereit.



Fehler auswählen, für den ein Quick-Fix zur Verfügung steht

Sie können anhand des Symbols erkennen, dass für beide Fehler im Ordner "Best Practices: Comparison" ein Quick-Fix zur Verfügung steht. Zur Verdeutlichung enthält die folgende Abbildung eine Übersicht über die Symbole für die Quick-Fixes:

1. Klicken Sie mit der rechten Maustaste auf den ersten Fehler in der Liste (siehe folgenden Screenshot).
2. Die Menüauswahl im **Quick-Fix**-Popup-Menü ist abhängig von der Lösung. Für den von Ihnen ausgewählten Fehler besteht die Korrektur in der Implementierung der Methoden "hashCode" und "equals".



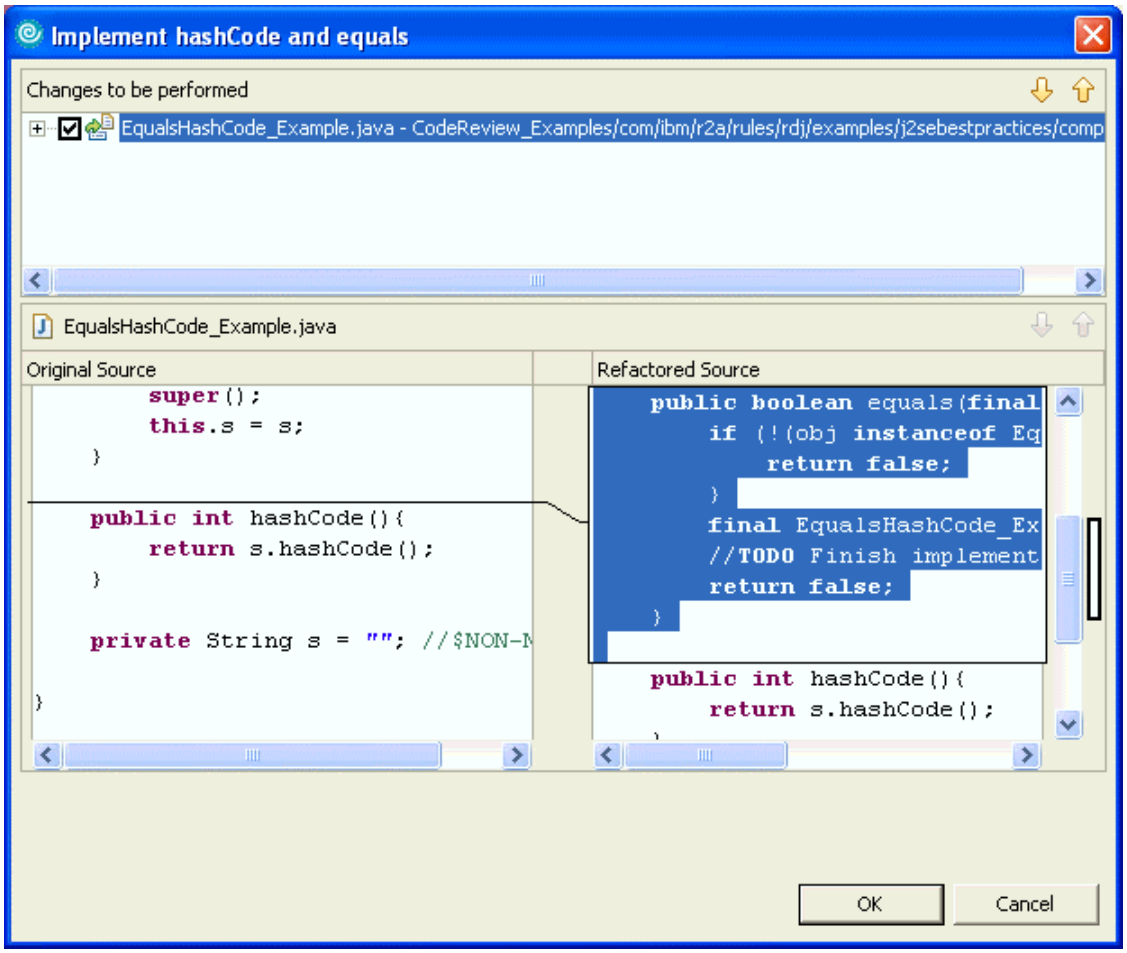
3. Klicken Sie auf **Quick Fix: Implement hashCode and equals**.

Quick-Fix anwenden

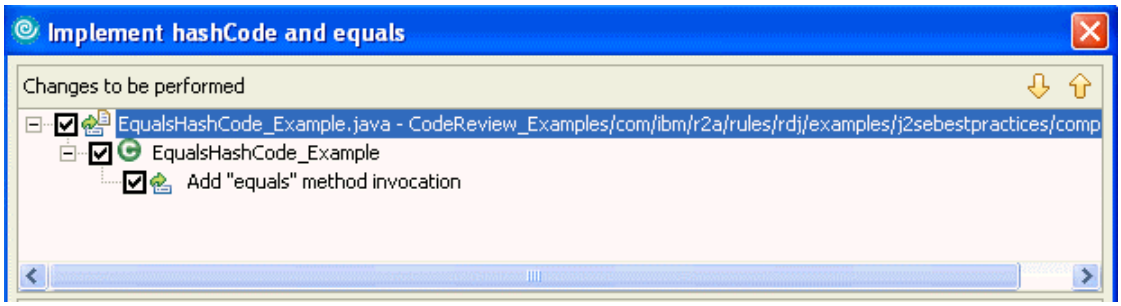
Der Quick-Fix für den von Ihnen ausgewählten Fehler besteht in der Implementierung der Methoden "hashCode" und "equals".

So führen Sie die Prüfung aus und wenden den Quick-Fix auf den Fehler an:

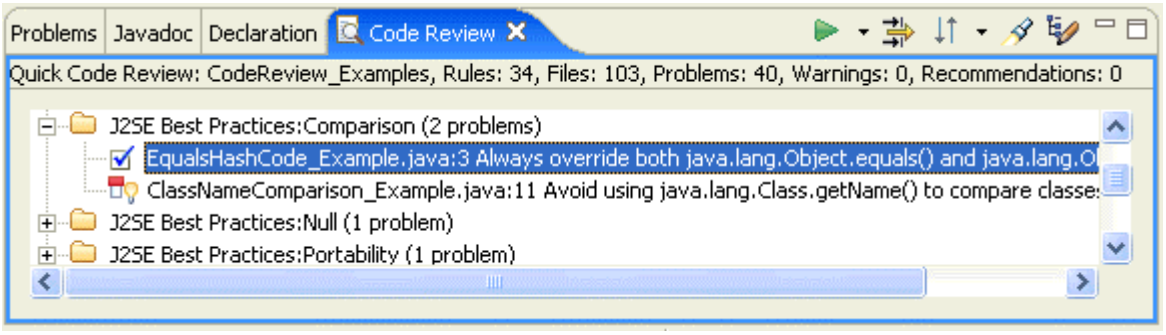
1. Zwei Codeversionen werden nebeneinander angezeigt (siehe folgenden Screenshot). Der ursprüngliche Quellcode wird links und der umstrukturierte Quellcode, der durch den Quick-Fix erstellt würde, wird rechts angezeigt. Wenn Sie den Quick-Fix anwenden, werden dadurch die hervorgehobenen fehlenden Codezeilen angefügt.



2. Erweitern Sie die Liste im Bereich **Changes to be performed**, um genau sehen zu können, welche und wie der Quick-Fix Änderungen vornimmt (siehe folgenden Screenshot):



3. Überprüfen Sie die Änderungen in der Liste. Klicken Sie dann auf **OK**, um den Quick-Fix auf alle ausgewählten Änderungen in der Liste anzuwenden.
4. Nach der Anwendung des Quick-Fixes erscheint ein Haken neben dem behobenen Fehler.



Neben dem Haken werden die folgenden Informationen bereitgestellt:

- Angewendeter Quick-Fix
- Die Nummer der Zeile im Quellcode, in der sich der gefundene Fehler befindet
- Regel in der Codeprüfung, die nicht eingehalten wurde

Sie haben damit die Übung "Codeprüfung ausführen und Quick-Fix anwenden" abgeschlossen.

Nachbearbeitung der Übung

Sie haben alle Aufgaben in der Übung "Codeprüfung ausführen und Quick-Fix anwenden" ausgeführt.

Aufgaben zum Ausführen einer Codeprüfung

Bei dieser Codeprüfung führten Sie folgende Aufgaben aus:

1. Auswählen einer Codeprüfung
2. Anzeigen der in der Codeprüfung angewendeten Regeln
3. Wählen eines Codes, auf dem die Prüfung ausgeführt werden sollte
4. Ausführen der Codeprüfung
5. Anzeigen der bei der Codeprüfung gefundenen Fehler
6. Auswählen eines Fehlers, um die folgenden Informationen dazu anzuzeigen:
    - Quellcode
    - Beschreibung, Beispiele und Lösungen

Aufgaben zur Anwendung eines Quick-Fixes

Beim Anwenden des Quick-Fixes führten Sie die nächste Gruppe von Aufgaben aus:

1. Erkennen, wenn ein Quick-Fix für einen gefundenen Fehler zur Verfügung steht
2. Anzeigen einer Liste mit Änderungen, die durch den Quick-Fix am Code vorgenommen würden
3. Voranzeigen des ursprünglichen und des umstrukturierten Codes
4. Anwenden des Quick-Fixes zur Umstrukturierung (Refactoring) des Codes
5. Abrufen einer Bestätigung über die Anwendung des Quick-Fixes

Vorteile einer Codeprüfung nutzen

Durch das proaktive Ausführen von Codeprüfungen können Sie die gefundenen Fehler nicht nur bereits in einem frühen Stadium analysieren, sondern auch korrigieren und somit die folgenden Probleme ausschließen:

- Beeinträchtigung der Leistung, Wartung oder Skalierbarkeit der Anwendung
- Verursachung von Kosten-, Zeit- und Ressourcenaufwänden für Ihr Unternehmen

Vorteile eines Quick-Fixes nutzen

Durch die Anwendung eines Quick-Fixes können Sie allgemeine Fehler automatisch korrigieren. Quick-Fixes ermöglichen Ihnen:

- Konsistente Korrektur eines Fehlers bei jedem Auftreten
- Effizientes Arbeiten und geringerer Zeitaufwand bei der Fehlerkorrektur

Zum Abschluss des Lernprogramms erhalten Sie in “Zusammenfassung: Codeprüfung ausführen” nochmals eine Übersicht über die Lernziele.

Zusammenfassung: Codeprüfung ausführen

Dieses Lernprogramm zeigte Ihnen die Vorgehensweise beim Ausführen einer Codeprüfung.

Erreichte Lernziele

Nach Abschluss der Übung können Sie nun die folgenden Aufgaben ausführen:

- Eine Codeprüfung ausführen
- Einen bereitgestellten Quick-Fix zur Lösung eines Problems anwenden

Weitere Informationen

Weitere Informationen zu den in diesem Lernprogramm behandelten Themen erhalten Sie in der Onlinehilfe zum Thema "Ausführen von Codeprüfungen".