コード・レビューの実行

このチュートリアルでは、一部のコード・レビュー機能について説明します。本書は、ソフトウェア開発者を対象として書かれています。

必要な時間

このチュートリアルを単に読み終えるには、約 15 分必要です。提供されているサンプル・プロジェクトを使用してこの演習を行うには、約 30 分必要です。

前提条件

このチュートリアルを完了するためには、Java ソフトウェア・アプリケーションの開発に精通している必要があります。IBM Rational Software Development Platform でのパースペクティブおよびビューの使用方法についての知識も必要です。

学習目標

このチュートリアルは複数のセクションに分かれているので、順に学習してください。自動コード・レビューの利点を読んだ後、以下の作業を行う方法を学習します。

- コード・レビューの実行
- 提供されている即時修正の適用によるコード内の問題の解決

準備ができたら、『コード・レビューの概要』から始めてください。

コード・レビューの概要

目的

*コード・レビュー*は、ソフトウェア開発者または設計者がコードをレビューするための処理を自動化する一連のルールです。手動コード・レビュー処理は時間がかかって現実的ではありませんが、自動コード・レビューは効果的で高速であり、一貫性も確保できます。自動コード・レビューは、手動コード・レビューを補足するものです。置き換えられるわけではありません。

利点

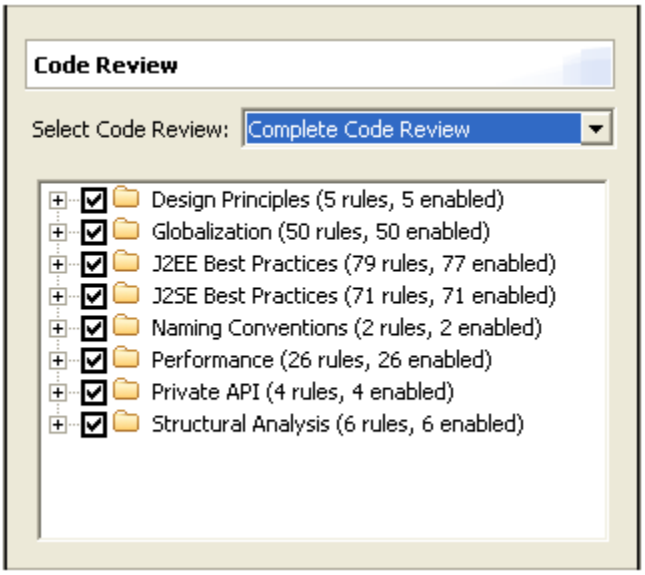
自動コード・レビュー・ツールは以下のタスクを実行するため、ソフトウェア開発処理に以下のような利点をもたらします。

- コードにあるバグの検出
- ベスト・プラクティスに準拠しているかどうかの検査
- 検出結果それぞれの説明と解決方法の提示
- 一部の典型的な検出結果に対する自動修正
- ルールの作成による、コード作成時のアプリケーション設計および標準への準拠

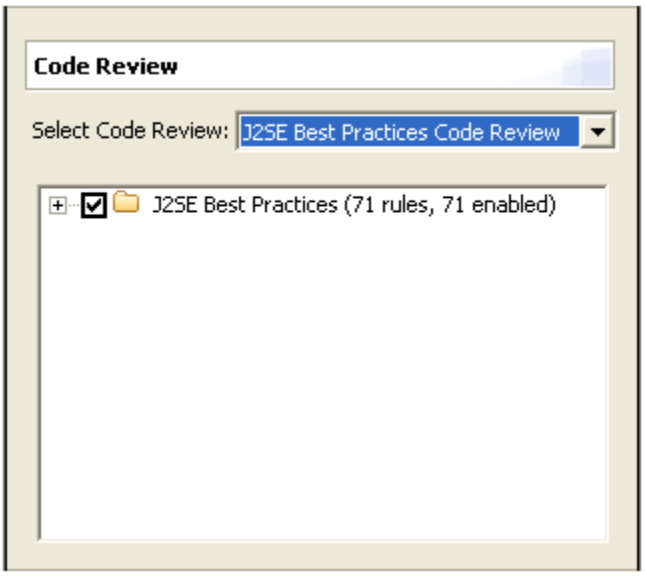
自動処理は高速であるため、コード・レビューは何回も実行できます。コード・レビューの検出結果により、問題を早期に見つけて訂正できます。変更は早期に行うのがもっとも容易であり、コストも低くなります。

提供されるコード・レビュー

いくつかのコード・レビューが提供されます。各コード・レビューはそれぞれ異なる一連のルールを適用します。これらのルールはフォルダーに編成されます。その時点で開発過程のどの段階にあるか、およびレビューの目的に応じて、ニーズに合わせてコード・レビューを選択できます。最も広範囲なレビューは完全コード・レビューであり、すべてのカテゴリーのルールを適用します（以下の画面取りを参照してください）。



一部のカテゴリーには、コード・レビューも関連付けられています。例えば、「J2SE ベスト・プラクティス」コード・レビューを選択すると、そのカテゴリーのルールのみを適用できます(以下の画面取りを参照してください)。これにより、コードの特定の側面に焦点を当ててコード・レビューを実行できます。



ユーザー定義のコード・レビュー

提供されるウィザードでルールを作成できます。ウィザードでは、アーキテクチャー制御と一般の 2 種類のルールから選択できます。これらのルールにより、ソフトウェア設計者は、ルールを作成してコード・レビューの機能を拡張し、設計の整合性を確保できます。

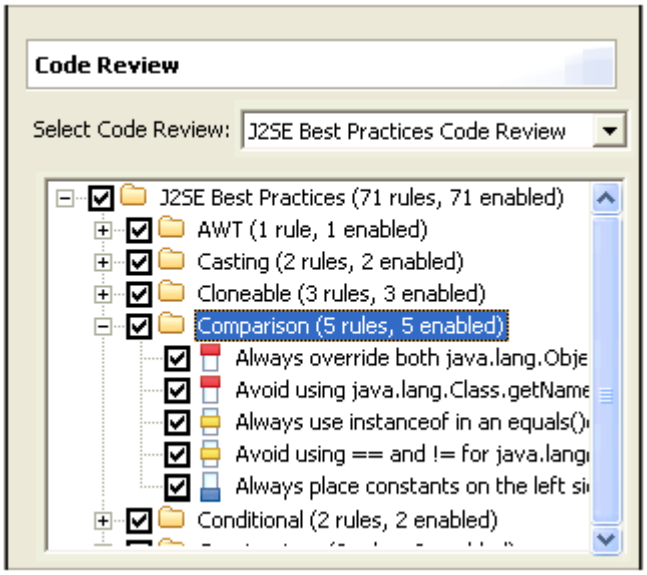
ルールの重大度レベル

各ルールには重大度レベルがあります。提供されているルールに割り当てられている重大度レベルは変更できます。ウィザードでルールを作成するときには、重大度レベルを指定します。重大度レベルは 3 つあり、それぞれ以下のアイコンで示されます。

- 問題 ( ):この検出結果には対処が必要です。
- 警告 ( ):この検出結果は、対処が必要な問題である可能性が高いことを示します。
- 推奨 ( ):この検出結果は、まだ重大な問題ではありませんが、この段階で対処することを強く推奨します。

「推奨」は最も低い重大度レベルですが、これらの検出結果への対処の重要性が低いわけではありません。これらは、技術チームが準拠すべき一連のベスト・プラクティスおよび業界標準を反映しています。これらの検出結果が即時に問題にならない場合であっても、将来問題となる可能性があります。

以下の画面取りに、「J2SE ベスト・プラクティス」コード・レビューの「比較」フォルダーのルールを示します。フォルダー内のルールには、いずれも 3 つの重大度レベルがあります。



一部の問題の自動修正

一部のよくある検出結果には、自動的な解決策として即時修正が用意されています。コード・レビューの検出結果に即時修正がある場合は、以下の図に示すアイコンのいずれかで示されます。

要約

ソフトウェア開発のライフ・サイクルにおいて、コード・レビューにより、コード本体を詳細にレビューする処理が自動化されます。提供されるコード・レビューにより、以下のタイプのレビューを実行できます。

- 広範な完全コード・レビューにより、すべてのカテゴリーからコード・ベースまでの広範なルールを適用します
- 精密かつ限定されたコード・レビューにより、グローバリゼーションや設計原理などの 1 つ以上の特定のカテゴリーのルールを適用します

提供されているウィザードを使用して、アプリケーションの設計構造の整合性を確保するための固有のルールを独自に作成できます。

自動コード・レビューは実行が高速なため、コード・ベースの問題や不整合を早期に検出できます。その結果、これらの問題を早期に修正できるため、アプリケーションの保守、スケーラビリティー、およびパフォーマンスへの影響が抑えられます。

『演習 1.1: 必要なリソースのインポート』に進んでください。

演習 1.1:必要なリソースのインポート

この演習では、サンプル・プロジェクト CodeReview_Examples のインポート方法について説明します。このサンプル・プロジェクトは、『演習 1.2: コード・レビューの実行と即時修正の適用』で使用します。

サンプル・プロジェクトの unzip

このチュートリアルのサンプル・プロジェクトは ZIP ファイルに含まれています。以下のステップにより、その ZIP ファイルから「workspace」フォルダーにファイルを解凍します。

1. *<installdir>*\rad\eclipse\plugins\com.ibm.r2a.rad.tutorial.doc_6.0.0\resources にナビゲートします（ここには ZIP ファイル CodeReview_Examples があります）。
2. CodeReview_Examples を *<installdir>*\updater\eclipse\workspace に解凍します。サンプル・プロジェクト・ファイルが「workspace」フォルダーに解凍され、インポートできるようになります。

「コード・レビュー」ビューのオープン

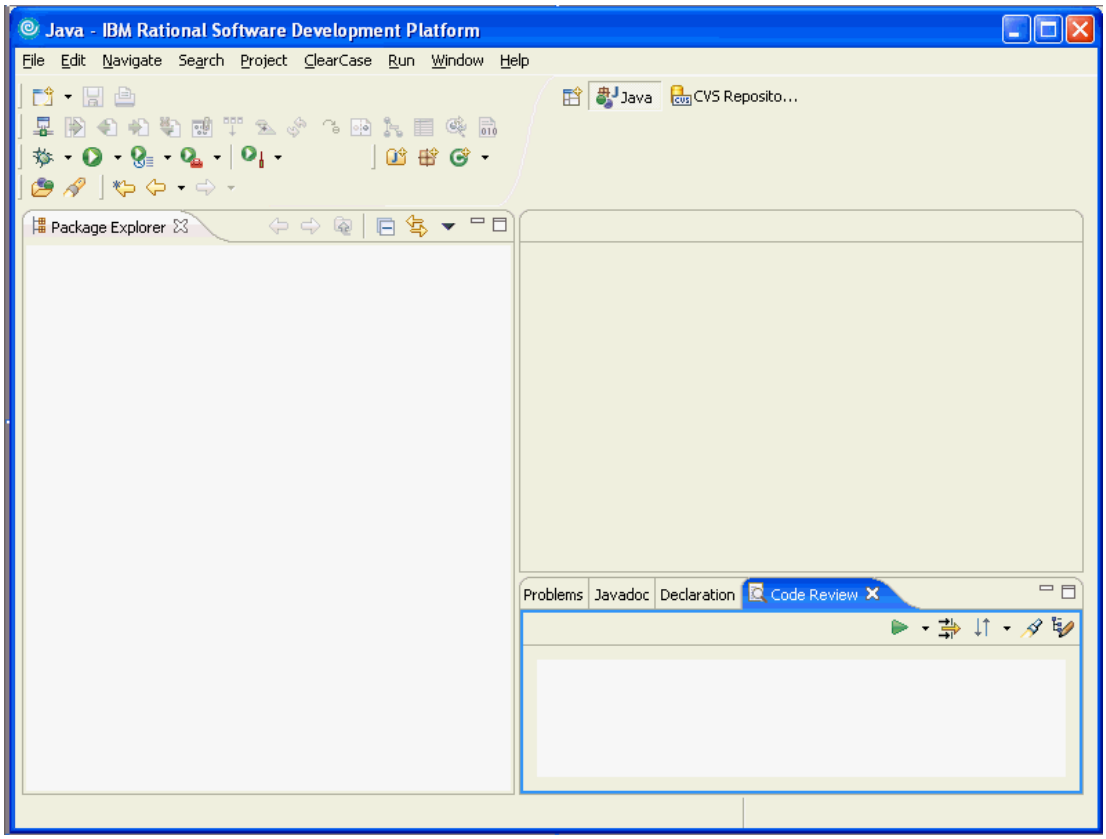
「コード・レビュー」ビューを表示するパースペクティブを開くには、以下のようにします。

1. IBM Rational Software Development Platform 6.0 を開始します。

2. **「ウィンドウ」 > 「設定」**をクリックします。
3. 左側のペインで**「ワークベンチ」**を展開し、**「機能」**をクリックします。
4. **「機能」**リストで「**Java Developer**」をクリックします。次に「**OK**」をクリックします。
5. **「ウィンドウ」 > 「パースペクティブを開く」 > 「Java」**をクリックします。
6. **「ウィンドウ」 > 「ビューの表示」 > 「その他」 > 「Java」 > 「コード・レビュー」**をクリックします。
7. **「ウィンドウ」 > 「ビューの表示」 > 「その他」 > 「Java」 > 「パッケージ・エクスプローラー」**をクリックします。

   Java パースペクティブを開き、「コード・レビュー」および「パッケージ・エクスプローラー」ビューを表示すると、以下の画面取りに示すパースペクティブが表示されます。実際に表示されるレイアウトは、以下の画面と異なることがあります。つまり、パース

ペクティブでは、ビューが異なる位置に表示されることがあります。このチュートリアルでは、以下の画面取りのレイアウトを使用します。



サンプル・プロジェクトのインポート

サンプル・プロジェクトをワークスペースにインポートするには、以下のようにします。

1. 「パッケージ・エクスプローラー」ビューを右マウス・ボタンでクリックし、ポップアップ・メニューを開きます。次に**「インポート」**をクリックして「インポート」ウィザードを開きます。
2. **「選択」**リストで**「既存プロジェクトをワークスペースへ」**をクリックします。次に**「次へ」**をクリックします。
3. **「プロジェクト・コンテンツ」**テキスト・ボックスの横にある**「ブラウズ」**をクリックし、*<installdir>*\updater\eclipse\workspace\CodeReview_Examples を選択します
4. **「終了」**をクリックします。サンプル・プロジェクトが関連ファイルとともにパッケージ・エクスプローラーにインポートされます。

演習の開始

はじめに、『演習 1.2: コード・レビューの実行と即時修正の適用』に進んでください。

演習 1.2:コード・レビューの実行と即時修正の適用

この演習では、『演習 1.1: 必要なリソースのインポート』を完了していることを想定としています。この演習では、最初にユーザー・シナリオを読みます。次に、ユーザー・シナリオで説明しているソフトウェア開発者の役割を想定します。

ユーザー・シナリオ

広い地域に分散した大規模な開発者のグループが、新しいソフトウェア・アプリケーションをコーディングしています。開発者が定期的にコード・レビューを実行してコードに問題がないことを確認することが重要です。

1 人の開発者がコード・レビューを実行し、自分の一般的なコーディング方法を確認するとします。新規に作成したコードをレビューしていくつかの領域でベスト・プラクティスに従っているかどうかを評価するには、開発者は、自動即時コード・レビューを実行します。このレビューは、提供されるルールのうち、いくつかのカテゴリーをコードに適用します。ルールの各カテゴリーでは、特定の領域（パフォーマンスなど）についてコードの品質を検査します。

コード・レビューが完了したら、検出結果のリストを確認します。各検出結果は、適用したルールに厳密に従っていないコードのストリングを表します。検出結果には即時修正が適用可能なものもあるため、開発者は、自動化された解決策を適用して問題をただちに訂正します。

演習の最初の部分では、以下の作業を行ってコード・レビューを実行します。

1. 実行するコード・レビューを選択します。
2. コード・レビューで適用するルールを表示します。
3. レビューを実行するコードを選択します。
4. コード・レビューを実行します。
5. コード・レビューの検出結果を表示します。
6. 検出結果を選択し、以下の情報を表示します。
    - ソース・コード。
    - 説明、例、および解決策。

次に、コード・レビューの特定の検出結果に即時修正を適用するために、以下の作業を行います。

1. 検出結果に対して即時修正が使用可能かを確認します。
2. 即時修正でコードに対して行われる変更のリストを確認します。
3. 即時修正を適用する前に、元のコードとリファクタリング後のコードをプレビューします。

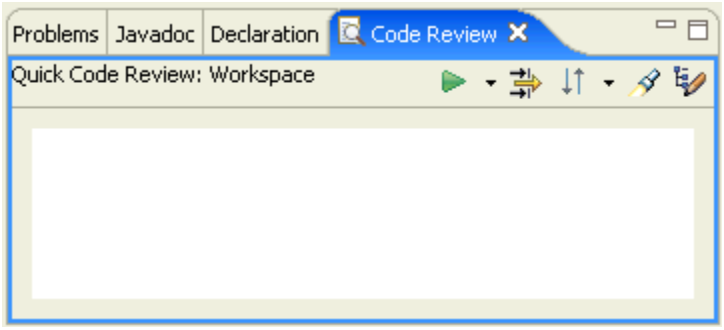
4. 即時修正を適用し、コードをリファクタリングします。
5. 即時修正を適用した後の状態を確認します。

演習

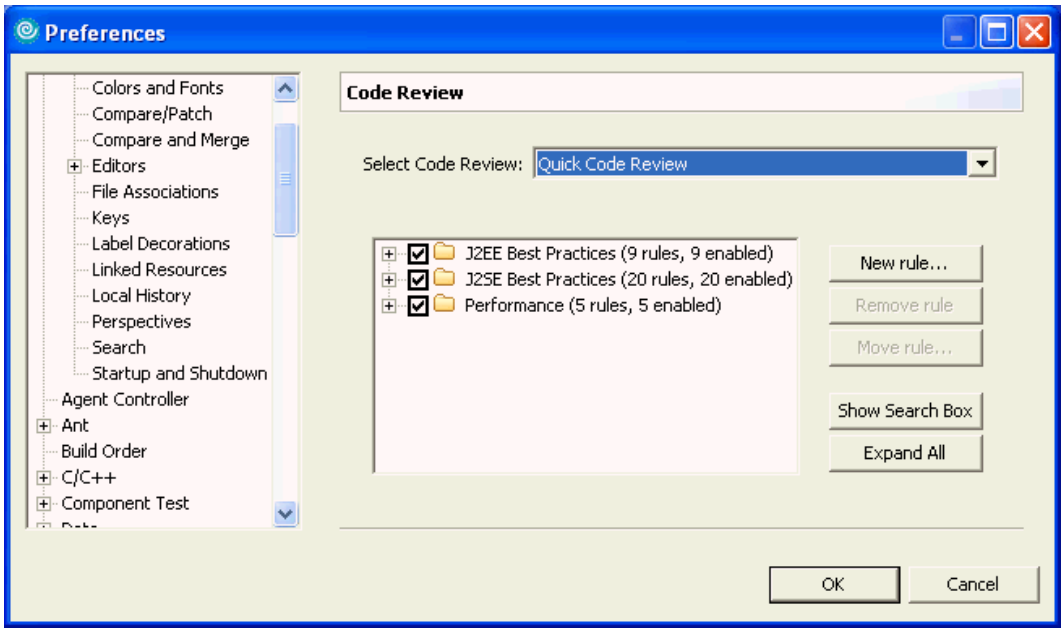
コード・レビューの選択

即時コード・レビューを選択するには、以下のようにします。

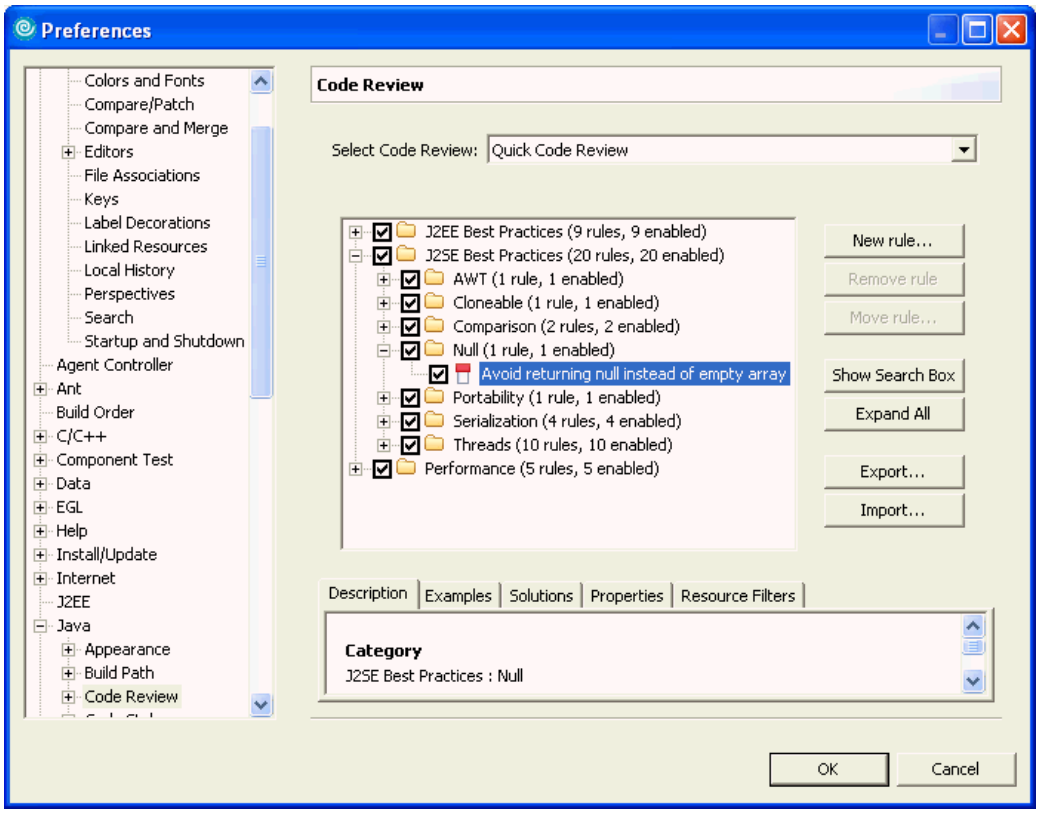
1. 「コード・レビュー」ビューのツールバーで、**「ルールの管理」**アイコン  をクリックします。



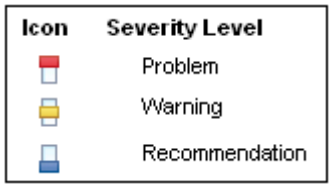
2. **「コード・レビューの選択」**リストで、**「即時コード・レビュー」**をクリックします。選択したコード・レビューのルールのフォルダーが表示されます（以下の画面取りを参照してください）。



3. コード・レビューで適用するルールを表示するために、「J2SE ベスト・プラクティス」フォルダーを展開し、「ヌル」サブフォルダーを展開します。「ヌル」フォルダーに、ルールと問題の重大度レベルが表示されます（以下の画面取りを参照してください）。



参考のために、重大度レベル・アイコンを以下の図に示します。



4. 「OK」をクリックし、「即時コード・レビュー」を選択します。

レビューするコード・ベースの選択

レビューするコード・ベースとしてプロジェクトを選択するには、以下のようにします。

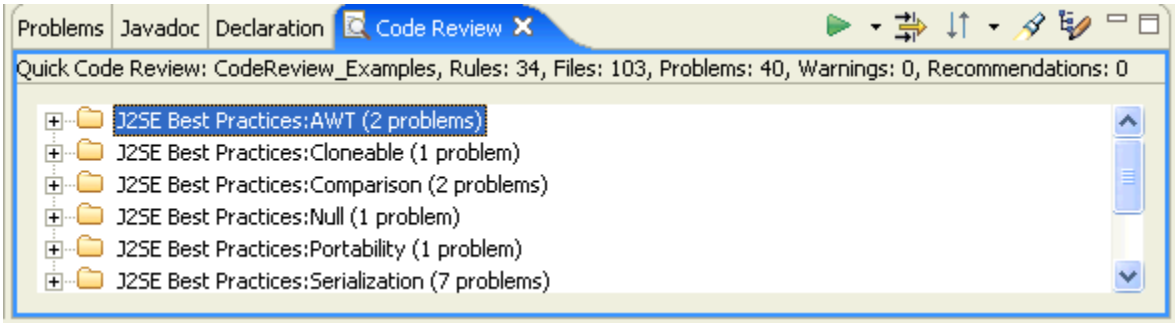
- 「コード・レビュー」ビューのツールバーで、「**レビュー**」アイコン（ ） > 「**プロジェクト**」 > 「**CodeReview_Examples のレビュー**」をクリックします

コード・レビューの実行

レビューするコード・ベースを選択したら、コード・レビューを実行します。ビューの右下隅にある進行状況表示バーを確認すると、状況が分かります。

コード・レビューでの検出結果の表示

コード・レビューが完了すると、「コード・レビュー」ビューに検出結果が表示されます（以下の画面取りを参照してください）。



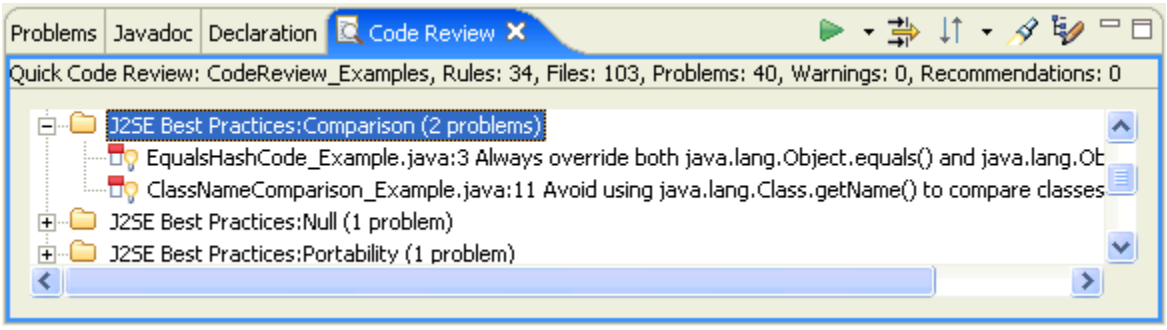
「コード・レビュー」ビューには、以下の情報が表示されます。

- コード・レビューの統計:検出結果の上の行に、最後に行ったコード・レビューに関する情報（タイプ、有効範囲、含まれるルールとファイルの数、検出結果の数と重大度）が表示されます。
- コード・レビューでの検出結果:コード・レビューでの検出結果は、「コード・レビュー」ビューのフォルダーにリストされます。各フォルダー名には、適用したルールのカテゴリーと検出結果の数が示されます。

コード・レビューでの検出結果に関する詳細情報の取得

コード・レビューでの検出結果に関する詳細情報を取得するには、以下のようにします。

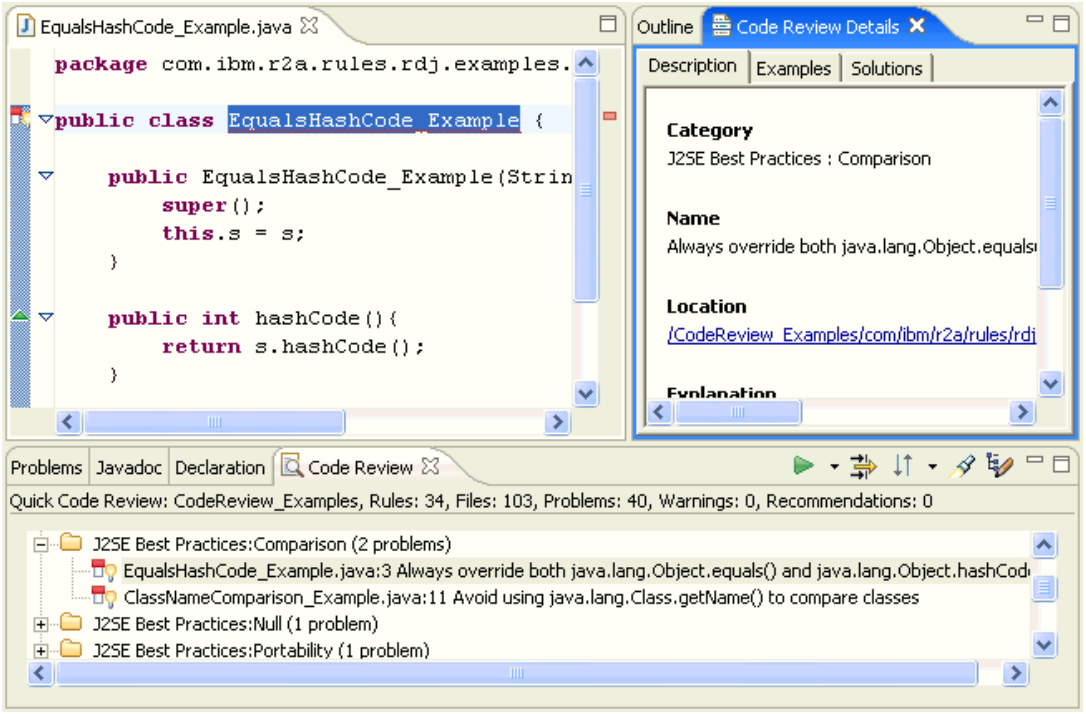
1. 「コード・レビュー」ビューで、**「J2SE ベスト・プラクティス: 比較」**フォルダーにスクロールします。次に、フォルダーを展開してフォルダー内の検出結果を表示します（以下の画面取りを参照してください）。



2. 最初の検出結果は EqualsHashCode_Example.java で始まっています。 以下のように適用されたルールがその後に示されます。

   java.lang.Object.equals() と java.lang.Object.hashCode() の両方を常に指定変更する

3. 最初の検出結果をダブルクリックします。詳細が 2 箇所に表示されます。概要および画面取りは以下のとおりです。
   - ソース・コード:検出されたコードが表示され、その正確な場所が強調表示されます。
   - 「コード・レビューの詳細」ビュー:検出結果を詳細に説明し、訂正するための例と解決策を提示します。

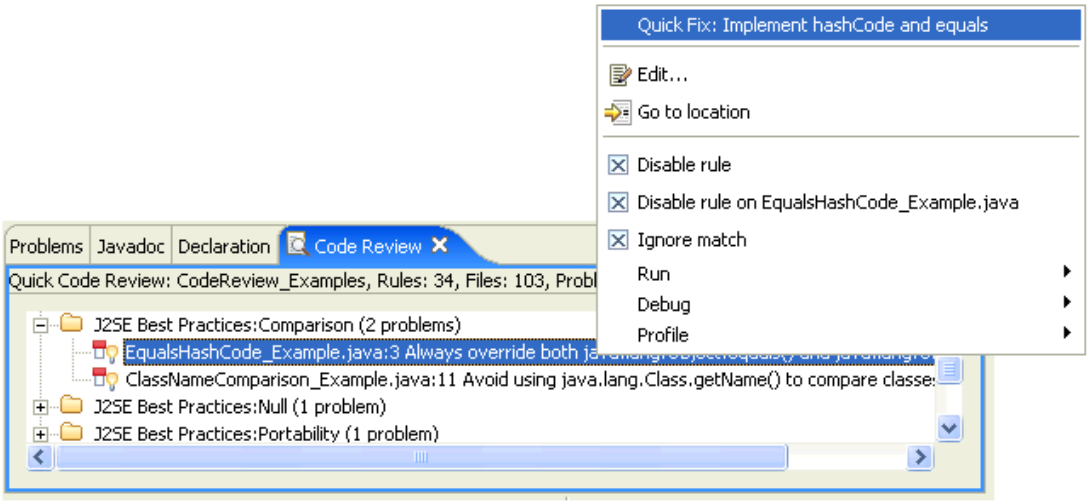


即時修正がある検出結果の選択

「ベスト・プラクティス: 比較」フォルダーの検出結果には、いずれも即時修正があることがアイコンで分かります。参考のために、即時修正アイコンを以下の図に示します。

1. リストの最初の検出結果を右マウス・ボタンでクリックします（次の画面取りを参照してください）。

2. 「**即時修正**」ポップアップ・メニューの選択項目は、解決策によって異なります。ここで選択した検出結果では、修正内容は hashCode および equals の実装です。



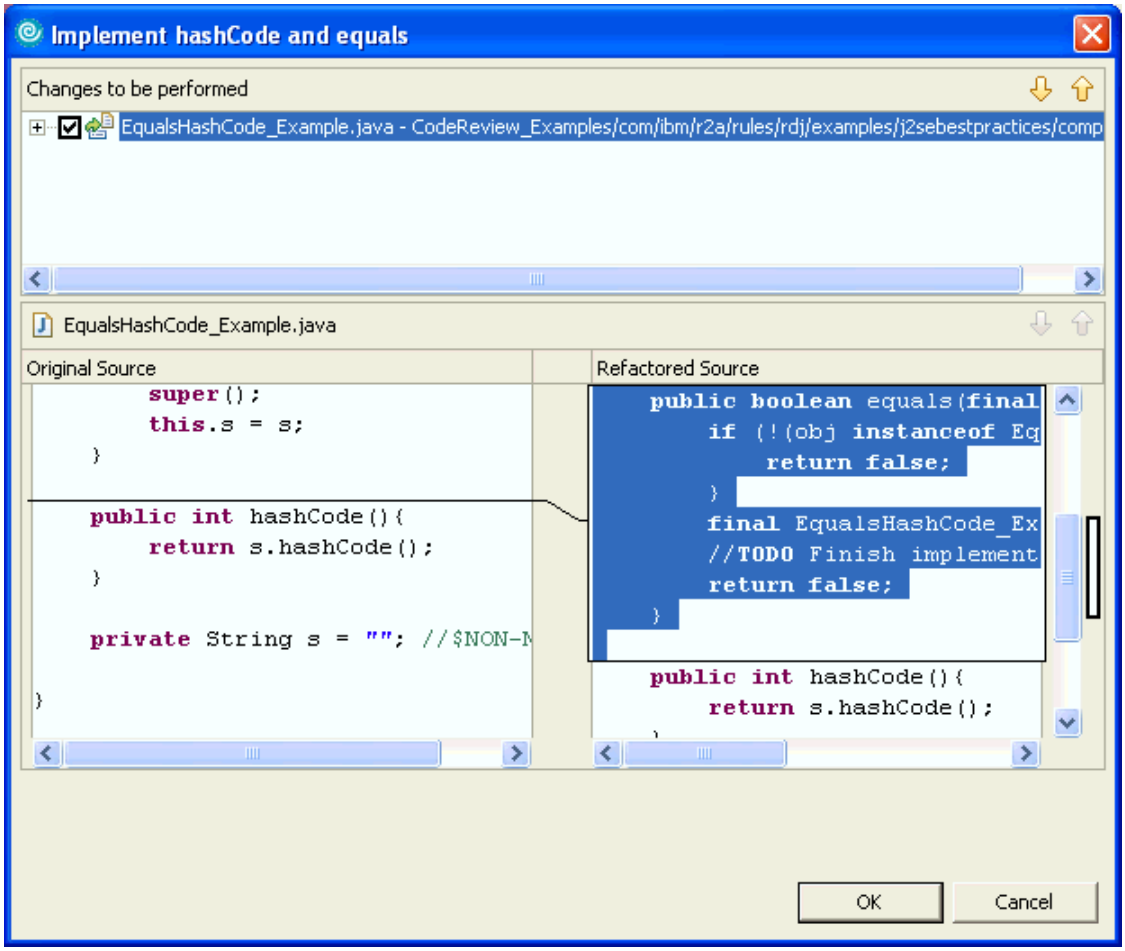
3. 「**即時修正: hashCode および equals の実装**」をクリックします。

即時修正の適用

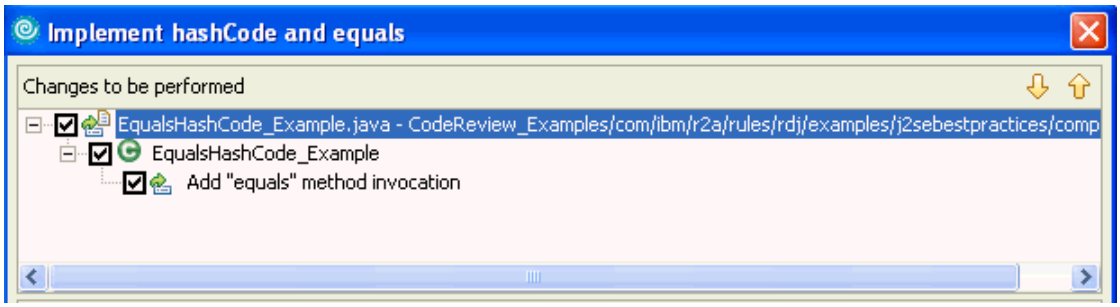
ここで選択した検出結果に対する即時修正は、hashCode および equals の実装です。

検出結果に対する即時修正を確認して適用するには、以下のようにします。

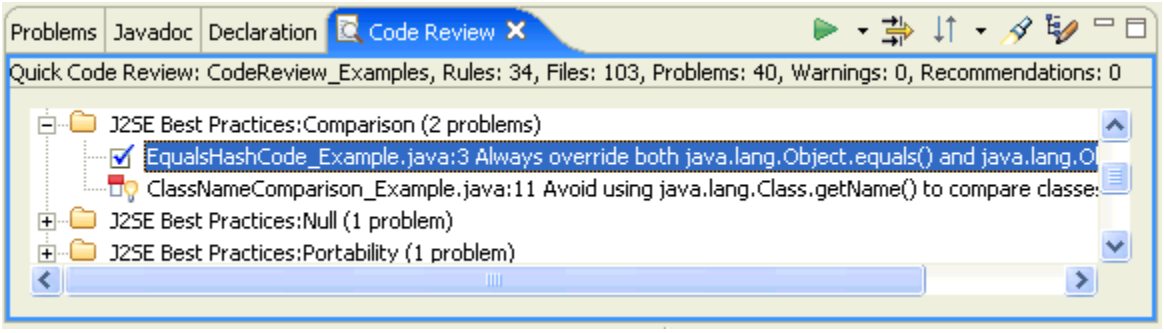
1. コードの横並びビューを表示します（以下の画面取りを参照してください）。元のソース・コードは左側に、即時修正によって作成されるリファクタリング後のソース・コードは右側に表示されます。即時修正の適用を選択すると、強調表示されているコードに欠落している行が付加されます。



2. 「実行される変更」セクションでリストを展開し、即時修正による変更内容および変更方法を確認します（以下の画面取りを参照してください）。



3. リストの変更を検討します。次に「OK」をクリックし、リストで選択したすべての変更に即時修正を適用します。
4. 即時修正を適用すると、解決した検出結果の横にチェックマークが表示されます。



チェックマークの横には以下の情報が表示されます。

- 適用した即時修正
- ソース・コード内で結果が検出された場所の行番号
- 従っていなかったコード・レビューのルール

『演習: コード・レビューの実行と即時修正の適用』は完了です。

演習のまとめ

『演習: コード・レビューの実行と即時修正の適用』の作業をすべて完了しました。

コード・レビューを実行するための作業

このコード・レビューを実行するときには、以下の作業を行いました。

1. 実行するコード・レビューを選択しました。
2. コード・レビューで適用するルールを表示しました。
3. レビューを実行するコードの本体を選択しました。
4. コード・レビューを実行しました。
5. コード・レビューの検出結果を表示しました。
6. 検出結果を選択し、以下の情報を表示しました。
   - ソース・コード。
   - 説明、例、および解決策。

即時修正を適用するための作業

即時修正を適用するときには、次の一連の作業を行いました。

1. 検出結果に対して即時修正が使用可能かを確認しました。
2. 即時修正でコードに対して行われる変更のリストを確認しました。
3. 元のコードとリファクタリング後のコードをプレビューしました。
4. 即時修正を適用し、コードをリファクタリングしました。
5. 即時修正を適用したことを確認しました。

コード・レビュー機能の活用

コード・レビューを事前に実行することにより、早期に検出結果を分析できます。つまり、以下の問題が発生する前に早期に対策をとることができます。

- アプリケーションのパフォーマンス、保守、スケーラビリティーへの影響
- 企業の資金、時間、リソースにおけるコストの上昇

即時修正機能の活用

提供されている即時修正を適用することにより、よくある検出結果を自動的に解決できます。即時修正は以下の方法で支援を行います。

- 毎回一貫した方法で問題を訂正する
- コーディングが不要になり、バグを修正するときの時間が短縮される

『要約：コード・レビューの実行』に記載されている学習目標を確認してチュートリアルを完了します。

要約：コード・レビューの実行

このチュートリアルでは、コード・レビューの実行方法について説明しました。

最終的な学習目標

演習をすべて完了すると、以下の作業を行えるようになります。

- コード・レビューを実行する。
- 提供されている即時修正を適用して問題を解決する。

詳細

このチュートリアルで扱うトピックの詳細については、コード・レビューの実行に関するオンライン・ヘルプを参照してください。