

코드 검토 실행

이 자습서에서는 몇 가지 코드 검토 기능에 대해 알려줍니다. 이것은 소프트웨어 개발자용으로 작성되었습니다.

필요한 시간

자습서를 읽기만 하는 데는 약 **15 분** 정도가 필요하고 제공된 샘플 프로젝트를 사용하여 연습을 하려면 약 **30 분** 정도가 필요합니다.

전제조건

학습을 완료하려면 Java 소프트웨어 어플리케이션 개발 경험이 있어야 합니다. 또한 IBM Rational Software Development Platform 에서 Perspective 및 보기의 사용법을 이해하면 도움이 됩니다.

학습 목표

이 자습서는 순서대로 수행해야 하는 섹션들로 구분되어 있습니다. 자동 코드 검토의 장점에 대해 읽은 후, 다음의 task 수행 방법에 대해 학습하게 됩니다.

- 코드 검토 실행
- 제공된 빠른 수정을 적용하여 코드의 문제점 해결

준비된 경우 코드 검토 개요를 시작하십시오. ”

코드 검토 개요

목적

코드 검토는 소프트웨어 개발자 또는 아키텍트가 코드를 검토하는 프로세스를 자동화하는 규칙 세트입니다. 수동 코드 검토 프로세스가 시간 소모적이며 토론에 의해 이루어지는 반면 자동 코드 검토는 효율적이고 신속하며 일관성이 있습니다. 자동 코드 검토는 수동 코드 검토를 보완합니다. 자동 코드 검토가 수동 코드 검토를 대체하지는 않습니다.

장점

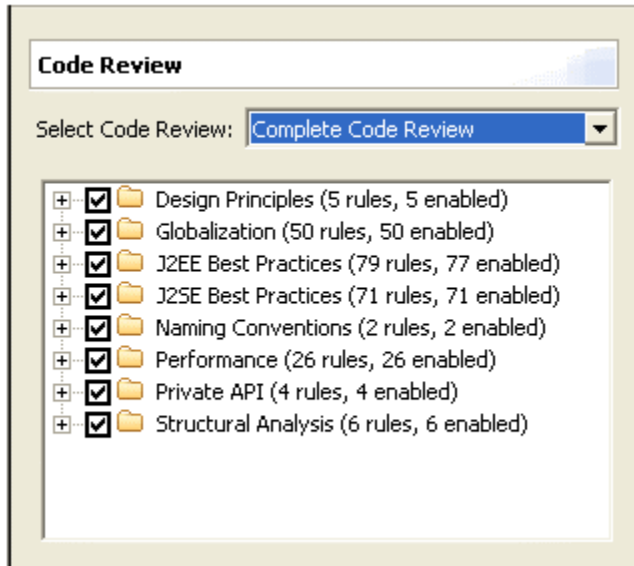
자동 코드 검토 툴은 다음과 같은 작업을 수행하여 다양한 방식으로 소프트웨어 개발 프로세스에 장점을 추가합니다.

- 코드에서 버그 찾기
- 베스트 프랙티스 준수 확인
- 각 결과 설명 및 솔루션 제공
- 일부 일반적인 결과에 대한 자동 수정 제공
- 코드 작성 시에 어플리케이션 설계 및 표준 준수 여부를 확인하는 규칙 작성

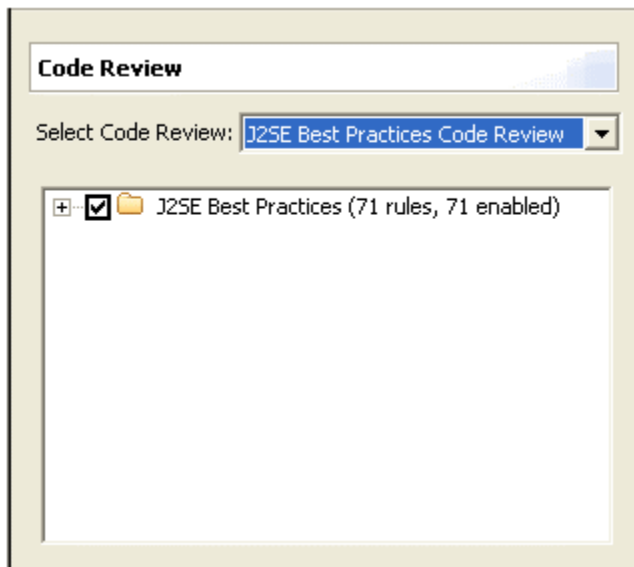
자동화된 프로세스는 빠르기 때문에 코드 검토를 자주 실행할 수 있습니다. 코드 검토 결과는 변경사항을 수행하기 쉽고, 적은 비용으로 문제점을 미리 발견하고 정정할 수 있게 해 줍니다.

코드 검토 제공

이 자습서에서는 몇 가지 코드 검토를 제공합니다. 각 코드 검토는 폴더로 구성된 여러 가지 규칙 세트를 적용합니다. 개발 프로세스 단계 및 검토 목적에 따라 사용자 요구사항에 맞는 코드 검토를 선택할 수 있습니다. 가장 자세한 검토는 다음 화면 캡처에서와 같이 모든 카테고리의 규칙에 적용되는 전체 코드 검토입니다.



일부 카테고리는 또한 연관된 코드 검토를 갖고 있습니다. 예를 들어 아래의 다음 화면 캡처에서와 같이 해당 카테고리만 규칙을 적용하려고 할 때는 J2SE 베스트 프랙티스 코드 검토를 선택할 수 있습니다. 이것은 특정 코드 부분에 집중하여 코드 검토를 실행할 수 있게 합니다.



사용자 정의 코드 검토

사용자는 마법사를 사용하여 규칙을 작성할 수 있고 두 가지 유형의 규칙(구조적 제어 및 일반) 중에서 선택할 수 있습니다. 이러한 규칙으로 소프트웨어 아키텍트는 설계 무결성을 확인하는 규칙을 작성하여 코드 검토 성능을 확장할 수 있습니다.

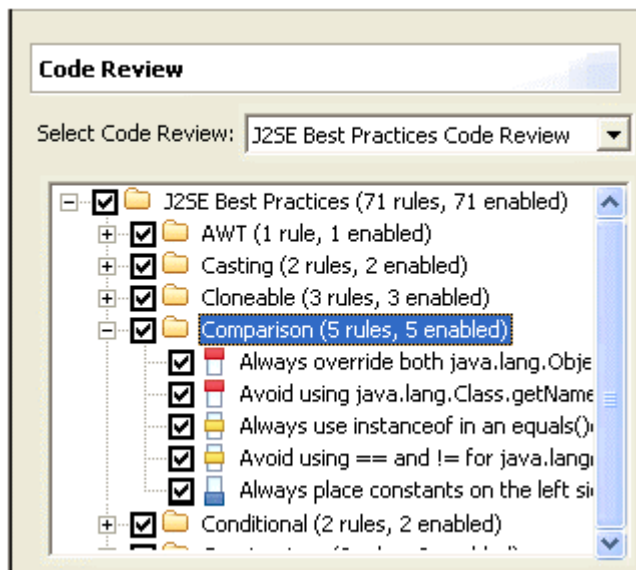
규칙의 심각도 레벨

각 규칙에는 심각도 레벨이 있습니다. 제공되는 규칙에 대해 지정된 심각도 레벨을 수정할 수 있습니다. 마법사에서 규칙을 작성할 때 심각도 레벨을 지정합니다. 세 가지 심각도 레벨은 다음과 같은 아이콘으로 표시됩니다.

- 문제점(🔴): 이 결과는 처리되어야 합니다.
- 경고(🟡): 이 결과는 처리되어야 하는 문제점과 유사합니다.
- 권장사항(📘): 이 결과는 심각하지는 않지만 바로 처리하도록 권장됩니다.

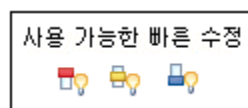
"권장사항"은 가장 낮은 심각도 레벨이지만 이러한 결과도 바로 처리하는 것이 좋습니다. 위와 같은 심각도 레벨은 엔지니어링 팀이 준수해야 하는 베스트 프랙티스 및 업계 표준 세트를 반영합니다. 이러한 결과는 긴급한 문제는 아니지만, 향후 문제가 될 수 있습니다.

다음 화면 캡처는 J2SE 베스트 프랙티스 코드 검토의 비교 폴더에 있는 규칙을 표시합니다. 폴더의 규칙은 모두 세 가지 심각도 레벨을 갖습니다.



일부 문제에 대한 자동 수정

일부 결과는 빠른 수정(제공되는 자동화 솔루션)과 함께 제공됩니다. 코드 검토의 결과에 빠른 수정이 있는 경우 다음 그림의 아이콘 중 하나로 표시됩니다.



요약

소프트웨어 개발 라이프사이클에서 코드 검토는 코드 본문의 피어 검토 프로세스를 자동화합니다. 제공되는 코드 검토를 사용하여 다음과 같은 유형의 검토를 실행할 수 있습니다.

- 모든 카테고리의 광범위한 규칙을 코드 베이스에 적용하는 광범위한 전체 코드 검토
- 자국어 지원, 설계 프린시플 또는 둘 다와 같이 하나 이상의 특정 카테고리의 규칙을 적용하는 데 중점을 둔 세부적인 코드 검토

또한 제공되는 마법사를 사용하여 사용자 어플리케이션 설계 구조의 무결성을 확인하는 사용자 자신의 특정 규칙을 작성할 수도 있습니다.

자동 코드 검토는 빠르게 실행할 수 있기 때문에 코드 베이스의 문제점과 불일치를 미리 발견할 수 있습니다. 결과적으로 이러한 문제점이 사용자 어플리케이션의 유지보수, 확장성 및 성능에 영향을 미치기 전에 수정할 수 있습니다.

이제 연습 1.1: 필요한 자원 가져오기를 시작할 준비가 되었습니다.”

연습 1.1: 필요한 자원 가져오기

이 연습은 샘플 프로젝트 CodeReview_Examples 를 가져오는 방법에 대해 알려줍니다. 샘플 프로젝트를 사용하여 연습 1.2: 코드 검토 실행 및 빠른 수정 적용을 수행하십시오. ”

샘플 프로젝트 압축 풀기

이 자습서의 샘플 프로젝트는 ZIP 파일에 포함되어 있습니다. 다음 단계를 수행하여 해당 ZIP 파일에서 사용자의 작업공간 폴더로 파일을 추출하십시오.

1. ZIP 파일 CodeReview_Examples 가 있는 `<install_dir>WradWeclipseWpluginsWcom.ibm.r2a.rad.tutorial.doc_6.0.0Wresources` 로 가십시오.
2. CodeReview_Examples 를 `<install_dir>WupdaterWeclipseWworkspace` 로 추출하십시오. 샘플 프로젝트 파일은 가져올 수 있도록 작업공간 폴더에 추출됩니다.

코드 검토 보기 열기

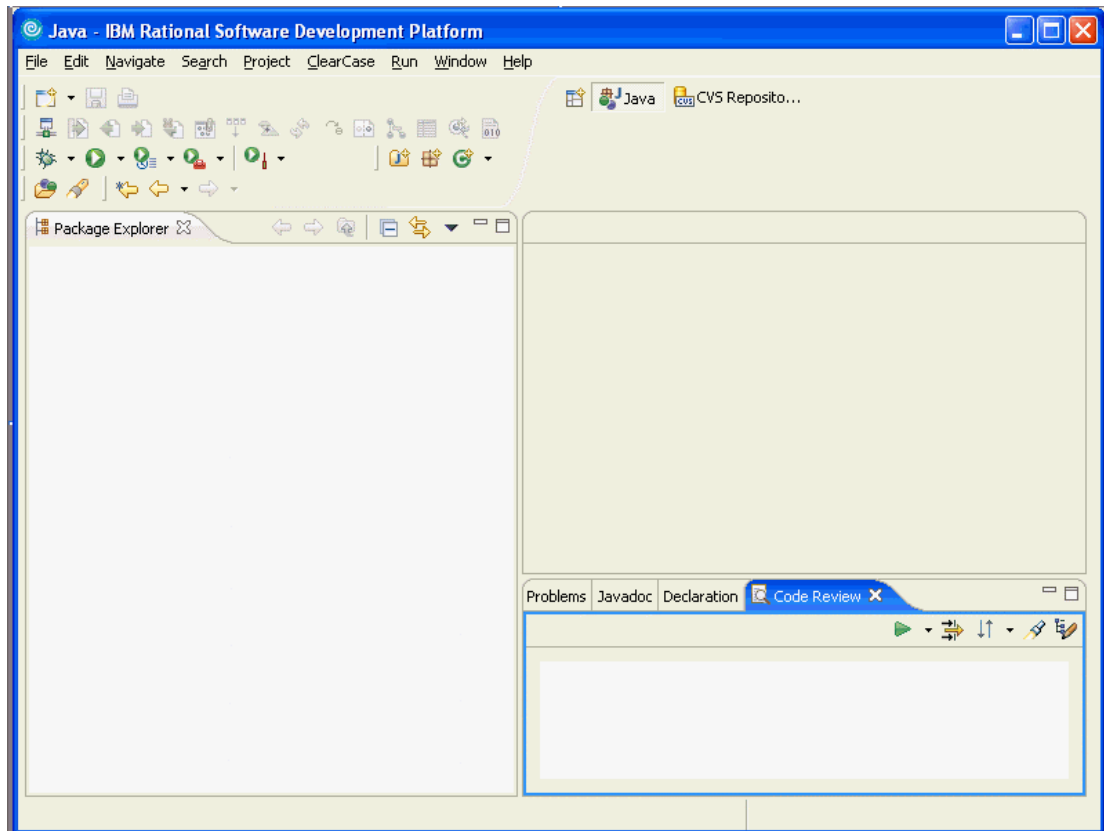
코드 검토 보기를 표시하는 Perspective 를 열려면 다음을 수행하십시오.

1. IBM Rational Software Development Platform 6.0 을 시작하십시오.



2. 창 > 환경 설정을 누르십시오.
3. 왼쪽 분할창에서 Workbench 를 펼치고 성능을 누르십시오.
4. 성능 목록에서 Java 개발자를 누르십시오. 그런 후 확인을 누르십시오.
5. 창 > Perspective 열기 > Java 를 누르십시오.
6. 창 > 보기 표시 > 기타 > Java > 코드 검토를 누르십시오.
7. 창 > 보기 표시 > 기타 > Java > 패키지 탐색기를 누르십시오.

Java Perspective 를 열고 코드 검토 및 패키지 탐색기 보기를 표시한 후에는 다음 화면 캡처에 표시된 Perspective 가 보입니다. 레이아웃은 다를 것입니다. 즉, Perspective 가 다른 위치에서 보기에 표시됩니다. 자습서는 화면 캡처에서 레이아웃을 사용합니다.



샘플 프로젝트 가져오기

샘플 프로젝트를 작업공간으로 가져오려면 다음을 수행하십시오.

1. 패키지 탐색기 보기에서 마우스 오른쪽 단추를 누르고 팝업 메뉴를 여십시오. 그런 후 **가져오기**를 눌러 가져오기 마법사를 여십시오.
2. **선택** 목록에서 **기존 프로젝트를 작업공간으로** 항목을 누르십시오. 그런 후 **다음**을 누르십시오.
3. **프로젝트** 콘텐츠 텍스트 상자 옆에 있는 **찾아보기**를 눌러 `<installDir>WupdaterWeclipseWorkspaceWCodeReview_Examples`를 선택하십시오.
4. **완료**를 누르십시오.

연습 시작

연습을 시작하려면 연습 1.2: 코드 검토 실행 및 빠른 수정 적용으로 가십시오. ”

연습 1.2: 코드 검토 실행 및 빠른 수정 적용

이 연습은 사용자가 연습 1.1: 필요한 자원 가져오기를 완료한 후에 수행해야 합니다. 연습을 시작하기 전에 먼저 사용자 시나리오를 읽어 보십시오. 사용자 시나리오에서 사용자의 역할은 소프트웨어 개발자라고 가정합니다.

사용자 시나리오

위치가 분산된 대형 개발자 그룹이 새 소프트웨어 어플리케이션을 코딩하려고 합니다. 개발자는 코드를 주기적으로 검토하여 코드의 문제를 확인하는 것이 중요합니다.

개발자는 자신의 개발 방식을 확인하기 위해 코드 검토를 실행하려고 합니다. 개발자는 몇 가지 영역에서 베스트 프랙티스의 준수를 평가하기 위해 새로 작성된 코드를 검토하여 자동화된 빠른 코드 검토를 실행합니다. 이 검토는 제공되는 규칙의 몇 가지 카테고리를 코드에 적용합니다. 각 규칙 카테고리는 성능과 같은 특정 영역에서 코드 품질을 확인합니다.

코드 검토가 완료되면 결과 목록이 표시됩니다. 각 결과는 적용된 규칙을 준수하지 않은 코드를 나타냅니다. 결과 중 하나에 적용 가능한 빠른 수정이 있어서, 개발자는 자동화 솔루션을 적용하고 문제점을 즉시 정정합니다.

첫 번째 연습 부분에서 아래의 작업을 수행하여 코드 검토를 실행하십시오.

1. 실행할 코드 검토를 선택하십시오.
2. 코드 검토에 적용된 규칙을 보십시오.
3. 검토를 실행할 코드를 선택하십시오.
4. 코드 검토를 실행하십시오.
5. 코드 검토 결과를 보십시오.
6. 결과를 선택하여 다음 정보를 보십시오.
 - o 소스 코드
 - o 설명, 예 및 솔루션


다음으로 코드 검토의 결과에 빠른 수정을 적용하려면, 아래의 작업을 수행하십시오.

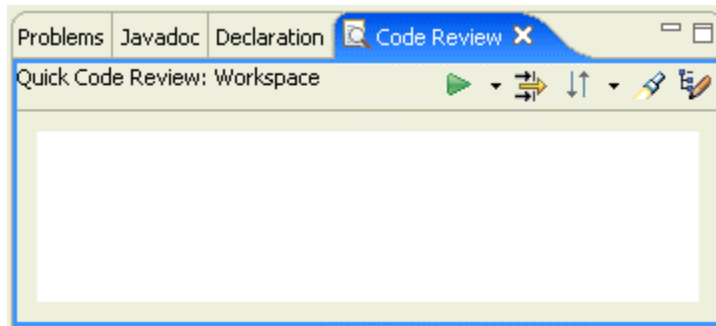
1. 결과에 대한 빠른 수정이 사용 가능한 시기를 알아보십시오.
2. 코드에 적용할 빠른 수정의 변경사항을 보십시오.
3. 빠른 수정을 적용하기 전에 원래의 리팩터 코드를 미리보십시오.
4. 빠른 수정을 적용하여 코드를 리팩터하십시오.
5. 빠른 수정을 적용한 후 확인 하십시오.

연습

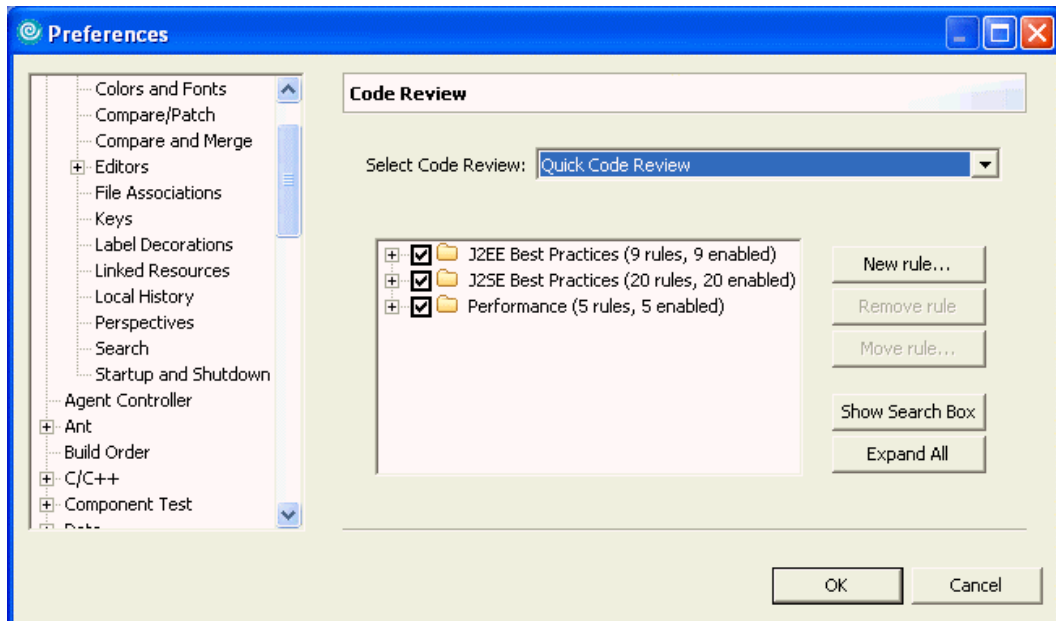
코드 검토 선택

빠른 코드 검토를 선택하려면 다음을 수행하십시오.

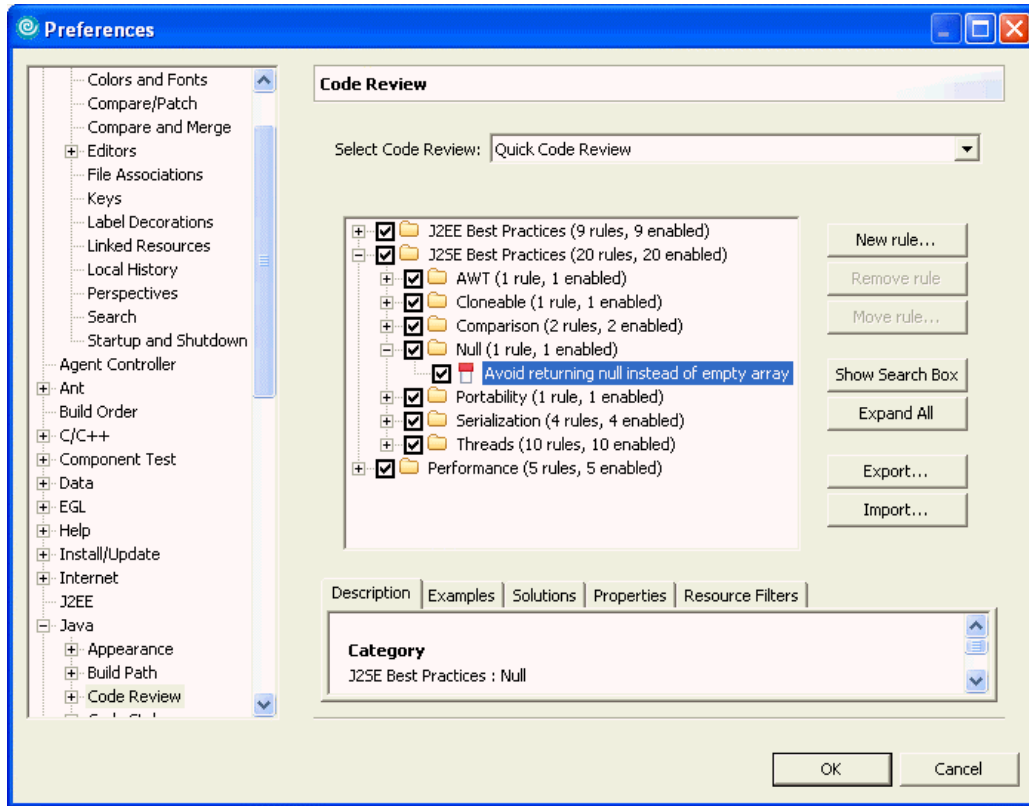
1. 코드 검토 보기의 도구 모음에서 **규칙 관리** 아이콘  을 누르십시오.



2. 코드 검토 선택 목록에서 **빠른 코드 검토**를 누르십시오. 다음 화면 캡처에서와 같이 선택한 코드 검토에 대한 규칙 폴더가 표시됩니다.



3. 코드 검토에 적용할 규칙 중 하나를 보려면 **J2SE 베스트 프랙티스** 폴더를 펼친 후 **널(Null)** 서브폴더를 펼치십시오. 널(Null) 폴더는 다음 화면 캡처에서와 같이 문제점 심각도 레벨이 있는 규칙을 표시합니다.



검토에서는 심각도 레벨 아이콘이 아래 그림같이 표시됩니다.

아이콘	심각도 레벨
	문제점
	경고
	권장사항

4. **확인**을 누르고 빠른 코드 검토를 선택하십시오.

검토할 코드 베이스 선택

검토할 코드 베이스로 프로젝트를 선택하려면 다음을 수행하십시오.

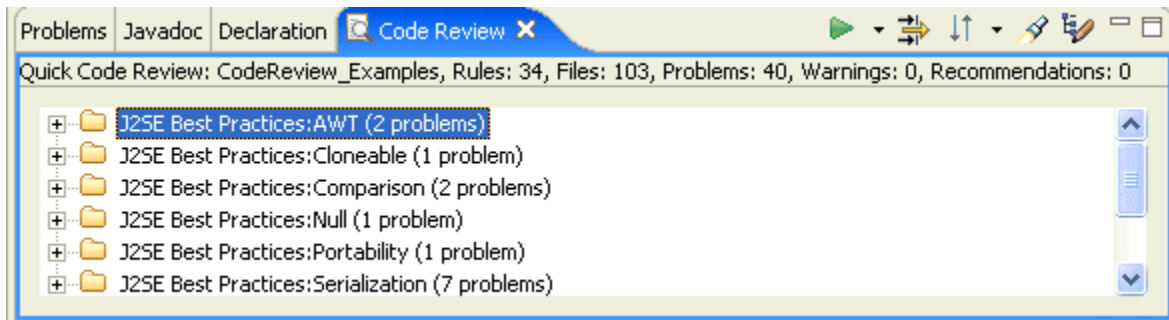
- 코드 검토 보기의 도구 모음에서 **검토** 아이콘() > **프로젝트** > **CodeReview_Examples** 검토를 누르십시오.

코드 검토 실행

검토할 코드 베이스를 선택하면 코드 검토가 실행됩니다. 보기의 오른쪽 하단 모서리의 진행 표시줄을 확인하여 해당 상태를 추적할 수 있습니다.

코드 검토 결과 보기

코드 검토가 완료되면 다음 화면 캡처에서와 같이 코드 검토 보기에 결과가 표시됩니다.



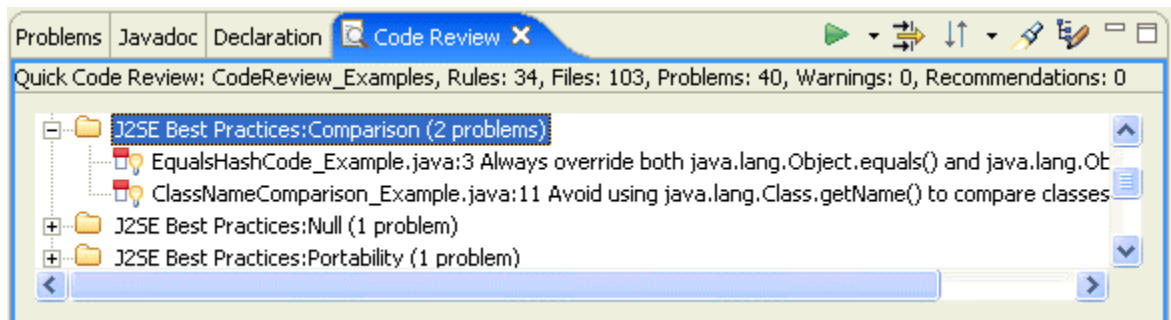
코드 검토 보기는 다음 정보를 제공합니다.

- 코드 검토 통계: 결과 위의 행은 가장 최근의 코드 검토에 대한 정보를 표시합니다(유형, 범위, 포함된 규칙 및 파일 수, 결과의 수 및 심각도).
- 코드 검토 결과: 코드 검토의 결과는 폴더 안의 코드 검토 보기에 나열됩니다. 각 폴더 이름은 적용되는 규칙 카테고리과 결과 수를 보여줍니다.

코드 검토 결과에 대한 자세한 정보 얻기

코드 검토 결과에 대한 자세한 정보를 얻으려면 다음을 수행하십시오.

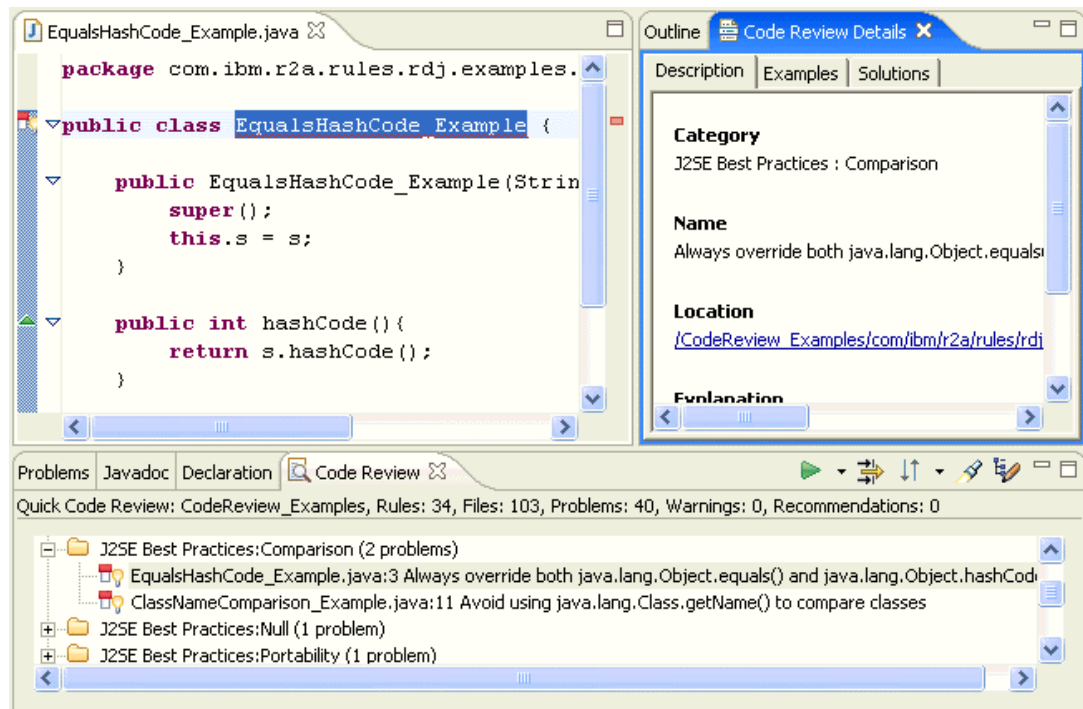
1. 코드 검토 보기에서 **J2SE 베스트 프랙티스: 비교** 폴더로 화면 이동하십시오. 그런 후 다음 화면 캡처에서와 같이 폴더를 펼쳐서 그 안의 결과를 표시하십시오.



2. 첫 번째 결과는 EqualsHashCode_Example.java 로 시작합니다. 적용되는 규칙은 그 뒤에 나옵니다.

항상 `java.lang.Object.equals()` 및 `java.lang.Object.hashCode()`를 먼저 표시하십시오.

3. 첫 번째 결과를 두 번 누르십시오. 이에 대한 세부사항이 다음 화면 캡처에서와 같이 두 위치에 나타납니다.
 - o 소스 코드: 결과가 나온 코드를 표시하고 그 정확한 위치를 강조표시합니다.
 - o 코드 검토 자세히 보기: 결과를 자세히 설명하고 문제를 해결할 수 있는 예 및 솔루션을 제공합니다.

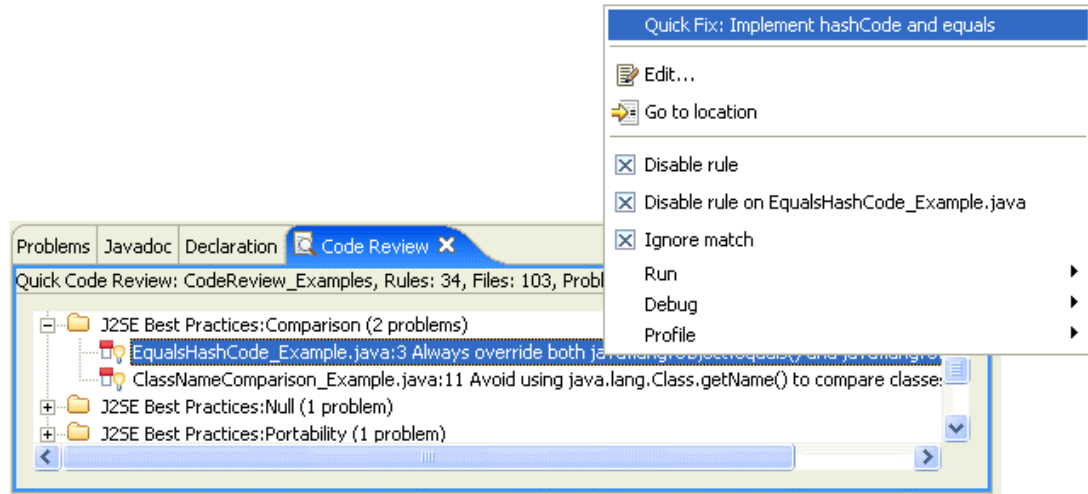


빠른 수정이 있는 결과 선택

베스트 프랙티스: 비교 폴더의 결과가 모두 아이콘에 의해 빠른 수정이 있다고 표시될 수 있습니다. 검토로서 빠른 수정 아이콘이 아래 그림같이 표시됩니다.



1. 다음 화면 캡처에서와 같이 목록에서 첫 번째 결과를 마우스 오른쪽 단추로 누르십시오.
2. **빠른 수정** 팝업 메뉴 선택항목은 솔루션에 따라 다릅니다. 선택한 결과에 따라 수정사항은 `hashCode` 및 `equals`를 구현합니다.



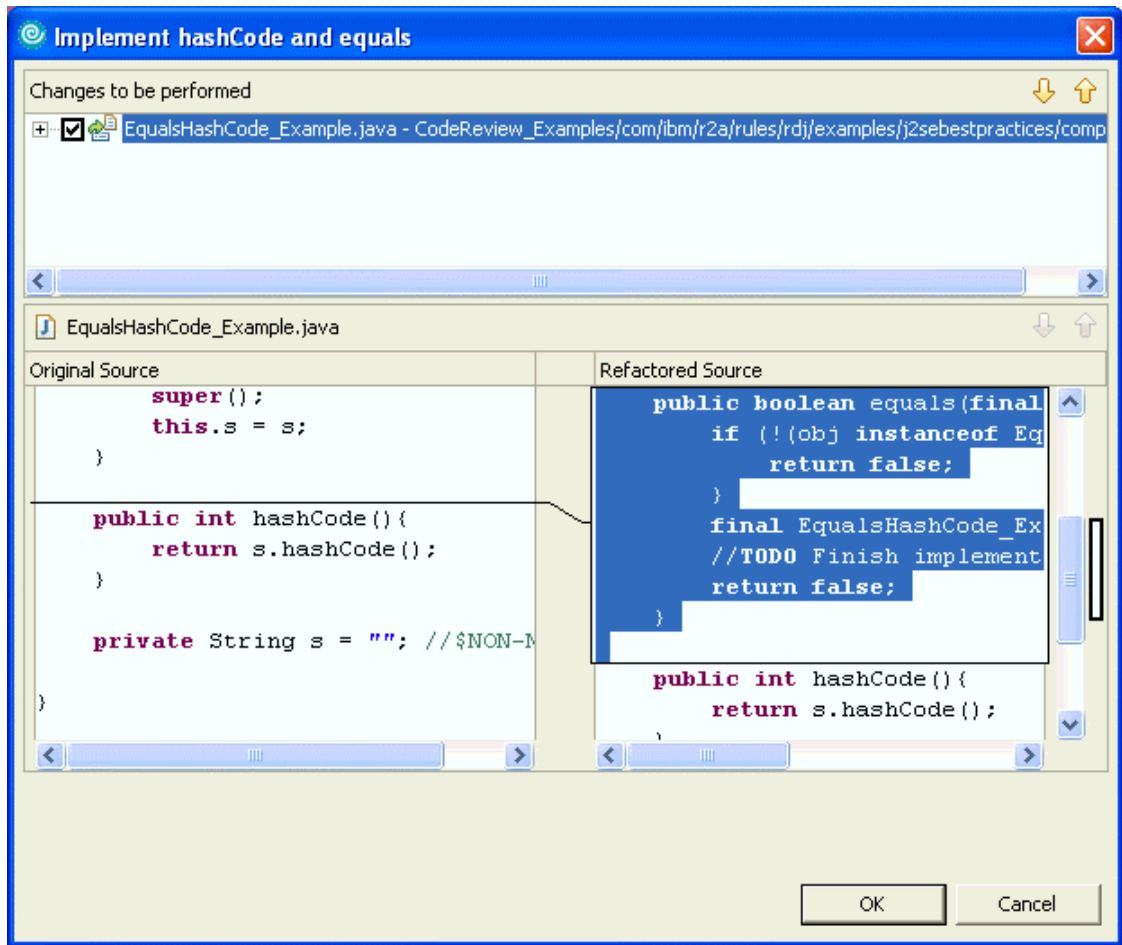
3. 빠른 수정: hashCode 및 equals 구현을 누르십시오.

빠른 수정 적용

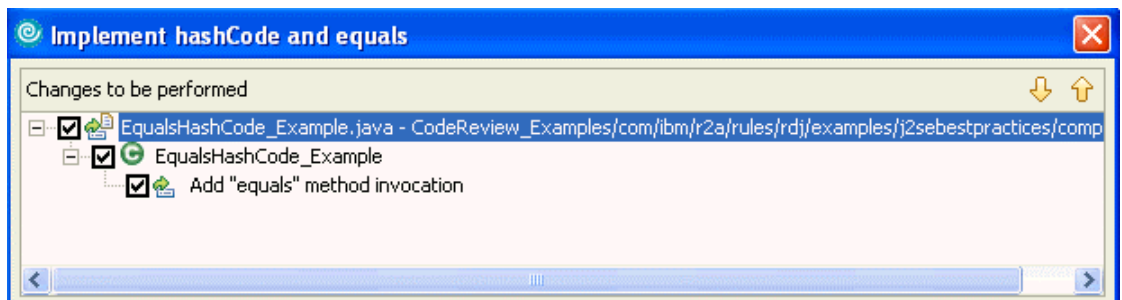
선택한 결과의 빠른 수정은 hashCode 및 equals 를 구현하는 것입니다.

빠른 수정을 검토하고 결과에 적용하려면 다음을 수행하십시오.

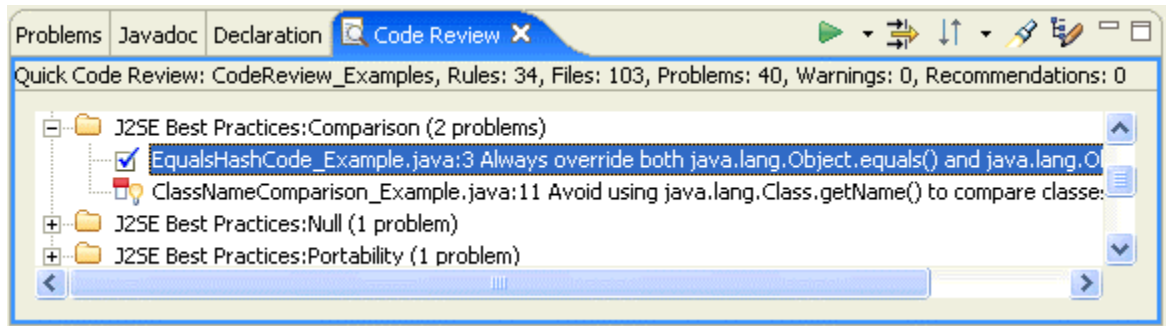
1. 다음 화면 캡처에서와 같이 코드가 나란히 표시됩니다. 최초 소스 코드는 왼쪽에 있고 빠른 수정에 의해 작성되는 리팩터 소스 코드는 오른쪽에 있습니다. 빠른 수정을 적용하기로 한 경우, 강조표시된 코드의 누락된 행이 추가됩니다.



- 수행할 변경사항 섹션에서 목록을 펼쳐서 다음 화면 캡처에서와 같이 빠른 수정이 변경할 내용과 방법이 정확히 무엇인지 확인하십시오.



- 목록에서 변경사항을 검토하십시오. 그런 후 **확인**을 눌러 목록의 모든 선택된 변경사항에 빠른 수정을 적용하십시오.
- 빠른 수정을 적용한 후에는 해결된 결과 옆에 체크표시가 표시됩니다.



체크표시 옆에는 다음과 같은 정보가 표시됩니다.

- o 적용한 빠른 수정
- o 결과가 있는 소스 코드의 행 번호
- o 준수되지 않은 코드 검토의 규칙

완료한 연습: 코드 검토 실행 및 빠른 수정 적용을 완료했습니다.

연습 요약

연습에 있는 모든 task 수행: 코드 검토 실행 및 빠른 수정 적용을 완료했습니다.

코드 검토 실행을 위한 task

이 코드 검토를 실행했다면 다음 task를 수행한 것입니다.

1. 실행할 코드 검토 선택
2. 코드 검토에 적용된 규칙 보기
3. 검토를 실행할 코드 본문 선택
4. 코드 검토 실행
5. 코드 검토 결과 보기
6. 결과를 선택하여 다음 정보 보기
 - o 소스 코드
 - o 설명, 예 및 솔루션

빠른 수정 적용을 위한 TASK

빠른 수정을 적용한 경우 다음 TASK를 수행한 것입니다.

1. 결과에 대한 빠른 수정이 사용 가능한 시기를 인식
2. 빠른 수정을 코드에 수행할 변경 목록 검토
3. 원본 소스 및 리팩터 코드를 미리 검토
4. 코드를 리팩터할 빠른 수정을 적용
5. 빠른 수정이 적용된 확인을 검토

코드 검토 기능 활용

코드 검토를 실행하면 결과를 미리 분석할 수 있습니다. 이것은 다음과 같은 문제가 발생하기 전에 미리 결과를 사전에 처리할 수도 있음을 의미합니다.

- 어플리케이션의 성능, 유지보수 또는 확장성 문제점.
- 회사 비용, 시간 및 자원 비용 문제점

빠른 수정 기능 활용

제공되는 빠른 수정을 적용하여 일반적인 결과를 해결할 수 있는 자동화된 방법이 있습니다. 빠른 수정은 다음과 같은 방식으로 사용자를 도와줍니다.

- 문제 발생 시 일관성 있게 문제점 정정
- 버그 수정 시간을 절약하여 코딩 시간을 최대한 제공

요약: 코드 검토 실행에서 학습 목표를 검토하여 학습을 완료하십시오.”

요약: 코드 검토 실행

이 자습서에서는 코드 검토 실행 방법에 대해 알려줍니다.

완료된 학습 목표

연습을 완료한 경우, 다음 타스크를 수행할 수 있습니다.

- 코드 검토 실행
- 제공된 빠른 수정을 적용하여 문제점 해결

추가 정보

이 자습서에서 다룬 주제에 대해 자세한 정보를 보려면, 코드 검토 실행에 대한 온라인 도움말을 참조하십시오.