**개발자 테스트 실행 순서**

**1. 체크리스트**

**2. 테스트케이스관리대장**

**3. 통합테스트수행**

**4. 결함관리대장**

**소프트웨어 테스트 자동화를 위한 공개SW 테스트 도구**

**1. 공개SW 테스트 도구**

소프트웨어 테스트 자동화를 위한 공개SW 테스트 도구들이 많이 소개되어 있으며, 이러한 도구들은 쉽고 효율적인 소프트웨어 테스트 수행을 지원한다. 하지만 테스트 도구를 사용하는 것은 장점 뿐 아니라 단점 역시 포함하고 있으므로 테스트 도구에 대한 이해를 통한 적절한 도구를 선택하는 것이 필요하다.

**▣ 테스트 도구의 장점**

● 테스트 데이터의 재입력 같은 반복 작업의 자동화

● 요구사항의 일관성과 반복의 가능성

● 정적인 측정값 등 객관적인 평가 기준 제공

● 성능에 대한 통계와 그래프 등 테스트 정보에 대한 쉬운 접근

**▣ 테스트 도구의 단점**

● 도입 후 프로세스 적용에 대한 시간, 비용, 노력에 대한 추가 투자

● 도구의 사용에 필요한 노력과 시간에 대한 추가 투자

● 비공개 상용SW의 경우 고가이며, 유지 관리 비용이 높음

**▣ 테스트 활동에 따른 도구 분류**

|  |  |  |
| --- | --- | --- |
| **테스트 활동** | **테스트 도구** | **내용** |
| 테스트 계획 | 요구사항 관리 | 고객 요구사항 정의 및 변경사항 관리 |
| 테스트 분석/설계 | 테스트 케이스 생성 | 테스트 기법에 따른 테스트 데이터 및 케이스 작성 |
| 커버리지 분석 | 대상시스템에 대한 테스트 완료 범위 척도 |
| 테스트 수행 | 테스트 자동화 | 기능 테스트 등 테스트 도구를 활용하여 자동화를 통한 테스트의 효율성을 높일 수 있음 |
| 정적분석 | 코딩표준, 런타임 오류 등을 검증 |
| 동적분석 | 대상시스템 시뮬레이션을 통한 오류 검출 |
| 성능 테스트 | 가상사용자를 인위적으로 생성하여 시스템 처리능력 측정 |
| 모니터링 | 시스템 자원(CPU, Memory 등) 상태 확인 및 분석 지원 도구 |
| 테스트 통제 | 형상관리 | 테스트 수행에 필요한 다양한 도구 및 데이터 관리 |
| 테스트 관리 | 전반적인 테스트 계획 및 활동에 대한 관리 |
| 결함 추적/관리 | 테스트에서 발생한 결함 관리 및 협업 지원 |

**▣ 주요 공개SW 테스트 도구 소개**

테스트 도구는 프로세스화 되었을 때 큰 효과를 볼 수 있기 때문에, 테스트 도구에 대한 도입시 조직적이고 장기적인 측면에서 고려해야 하며, 활용 가능한 주요 공개SW 테스트 도구는 아래와 같다.

● **정적분석 도구**

소스코드에 대한 코딩 표준/스타일, 복잡도 및 잔존 결함을 발견하기 위하여 사용하는 도구

|  |  |  |  |
| --- | --- | --- | --- |
| **분류** | **제품명** | **세부정보** | |
| 결함 예방 /발견 | pmd(cpd) | 개요 | 자바 소스코드에 대한 잠재적인 문제에 대한 분석 |
| 지원 환경 | Linux, Windows |
| 개발도구 지원 | Eclipse, NetBeans 등 |
| 홈페이지 | <http://pmd.sourceforge.net/> |
| findbugs | 개요 | 자바 소스코드에 대한 잠재적인 문제에 대한 분석 |
| 지원 환경 | Cross-Platform |
| 개발도구 지원 | Eclipse, NetBeans 등 |
| 홈페이지 | <http://findbugs.sourceforge.net/> |
| 코딩 표준 | checkstyle | 개요 | 자바 프로그램에 대한 코딩 표준 준수 검사 도구로, 다양한 개발 도구에 통합하여 활용 가능 |
| 지원 환경 | Cross-Platform |
| 개발도구 지원 | Ant, Eclipse, NetBeans 등 |
| 홈페이지 | <http://checkstyle.sourceforge.net/> |
| 코드 복잡도 | ccm | 개요 | 소스코드 복잡도 분석 도구이며, Linux, Mac환경에서는 CLI 형태로 지원됨 |
| 지원 환경 | Cross-Platform |
| 개발도구 지원 | Visual Studio |
| 홈페이지 | <http://www.blunck.info/ccm.html> |
| eclipsemetrics | 개요 | 소스코드 복잡도 분석 소스 코드 통계 정보 제공 도구 |
| 지원 환경 | Cross-Platform |
| 개발도구 지원 | Eclipse |
| 홈페이지 | <http://www.stateofflow.com/projects/16/eclipsemetrics> |
| sourcemonitor | 개요 | 윈도우 기반 소스코드 복잡도 분석 도구 |
| 지원 환경 | Windows |
| 개발도구 지원 | - |
| 홈페이지 | <http://www.campwoodsw.com/sourcemonitor.html> |
| cobertura | 개요 | 자바언어에 대한 소스코드 복잡도 분석 및 커버리지 측정 |
| 지원 환경 | Cross-Platform |
| 개발도구 지원 | Ant, Maven |
| 홈페이지 | <http://cobertura.sourceforge.net/> |
| javancss | 개요 | 자바언어에 대한 소스코드 복잡도 분석 도구, CLI 형태로 지원됨 |
| 지원 환경 | Cross-Platform |
| 개발도구 지원 | Ant, Jacob |
| 홈페이지 | <http://www.kclee.de/clemens/java/javancss/> |

● **동적분석 도구**

프로그램을 실행하여, 코드 내에 존재하는 메모리 누수, 쓰레드 결함 등을 분석하기 위하여 사용하는 도구

|  |  |  |
| --- | --- | --- |
| **제품명** | **세부정보** | |
| Avalanche | 개요 | Valgrind 프레임워크 기반으로 구현되었으며, 프로그램에 대한 결함 및 취약점 동적 분석 도구 |
| 지원 환경 | Linux, Android |
| 개발도구 지원 | - |
| 홈페이지 | <http://code.google.com/p/avalanche/> |
| Valgrind | 개요 | 프로그램 내에 존재하는 메모리 및 쓰레드의 결함을 발견하는 동적 분석 도구 |
| 지원 환경 | Cross-Platform |
| 개발도구 지원 | Eclipse, NetBeans 등 |
| 홈페이지 | [http://valgrind.org](http://valgrind.org/) |

● **테스트 자동화 프레임워크**

단위테스트, 통합테스트 등 테스트 단계별 자동화 도구를 활용한 기능 테스트 도구

|  |  |  |
| --- | --- | --- |
| **제품명** | **세부정보** | |
| xUnit | 개요 | java(Junit), C++(Cppunit), .Net(Nunit) 등 다양한 언어를 지원하는 단위테스트 프레임워크 |
| 지원 환경 | 각각의 도구별 지원 환경 상이 |
| 개발도구 지원 | eclipse 등 |
| 홈페이지 | <http://www.junit.org/> <http://sourceforge.net/apps/mediawiki/cppunit/index.php?title=Main_Page> <http://www.nunit.org/> |
| STAF | 개요 | 서비스 호출, 컴포넌트 재사용 등 다양한 환경을 지원하는 테스트 프레임워크 |
| 지원 환경 | Cross-Platform |
| 개발도구 지원 | eclipse |
| 홈페이지 | <http://staf.sourceforge.net/> |
| FitNesse | 개요 | 웹기반 테스트케이스 설계/실행/결과확인 등을 지원하는 테스트 프레임워크 |
| 지원 환경 | Cross-Platform |
| 개발도구 지원 | eclipse |
| 홈페이지 | <http://fitnesse.org/> |
| NTAF | 개요 | NHN 테스트 자동화 프레임워크이며, STAF와 FitNesse를 통합 |
| 지원 환경 | Cross-Platform |
| 개발도구 지원 | eclipse, Maven 등 |
| 홈페이지 | <http://dev.naver.com/projects/ntaf/> |
| Selenium | 개요 | 다양한 브라우저 지원 및 개발언어를 지원하는 웹애플리케이션 테스트 프레임워크 |
| 지원 환경 | Cross-Platform |
| 개발도구 지원 | - |
| 홈페이지 | <http://seleniumhq.org/> |
| watir | 개요 | Ruby 기반 웹애플리케이션 테스트 프레임워크 |
| 지원 환경 | Cross-Platform |
| 개발도구 지원 | - |
| 홈페이지 | <http://watir.com/> |

● **성능 테스트 도구**

가상사용자를 인위적으로 생성하여 시스템의 처리량, 응답시간 등을 테스트하기 위한 도구

|  |  |  |
| --- | --- | --- |
| **제품명** | **세부정보** | |
| JMeter | 개요 | HTTP, FTP, LDAP 등 다양한 프로토콜을 지원하는 부하 테스트 도구 |
| 지원 환경 | Cross-Platform |
| 개발도구 지원 | eclipse, Ant |
| 홈페이지 | <http://jmeter.apache.org/> |
| AB | 개요 | 아파치 웹서버 부하 테스트 도구이며, CLI기반으로 동작 |
| 지원 환경 | Cross-Platform |
| 개발도구 지원 | - |
| 홈페이지 | <http://httpd.apache.org/docs/2.2/programs/ab.html> |
| OpenSTA | 개요 | HTTP, HTTPS 프로토콜에 대한 부한 테스트 도구 |
| 지원 환경 | Windows |
| 개발도구 지원 | - |
| 홈페이지 | <http://opensta.org/> |
| LoadUI | 개요 | HTTP, JDBC 등 다양한 프로토콜을 지원하며, 서버 모니터링, Drag&Drop 등 사용자 편리성이 강화된 부하 테스트 도구 |
| 지원 환경 | Cross-Platform |
| 개발도구 지원 | - |
| 홈페이지 | <http://www.loadui.org/> |

● **시스템 모니터링 도구**

서버 자원(CPU, Memory 등)에 대한 사용 현황을 파악할 수 있는 도구

|  |  |  |
| --- | --- | --- |
| **제품명** | **세부정보** | |
| Nagios | 개요 | 웹기반 서버, 서비스, 애플리케이션 등에 대한 모니터링 도구 |
| 지원 환경 | Cross-Platform |
| 개발도구 지원 | - |
| 홈페이지 | <http://www.nagios.org/> |
| Zenoss | 개요 | 웹기반 서버, 서비스, 애플리케이션 등에 대한 모니터링 도구 |
| 지원 환경 | Cross-Platform |
| 개발도구 지원 | - |
| 홈페이지 | <http://community.zenoss.org/index.jspa> |
| Zabbix | 개요 | 웹기반 서버, 서비스, 애플리케이션 등에 대한 모니터링 도구 |
| 지원 환경 | Cross-Platform |
| 개발도구 지원 | - |
| 홈페이지 | [http://www.zabbix.com](http://www.zabbix.com/) |
| Hyperic HQ | 개요 | 웹기반 서버, 서비스, 애플리케이션 등에 대한 모니터링 도구 |
| 지원 환경 | Cross-Platform |
| 개발도구 지원 | - |
| 홈페이지 | <http://sourceforge.net/projects/hyperic-hq/> |

**- PMD Plugin 설치**

http://pmd.sourceforge.net/eclipse/

1. Help -> Install New Software...

2. Click on Add..

3. Enter the following:

Name: **PMD**

URL: **https://dl.bintray.com/pmd/pmd-eclipse-plugin/updates/**

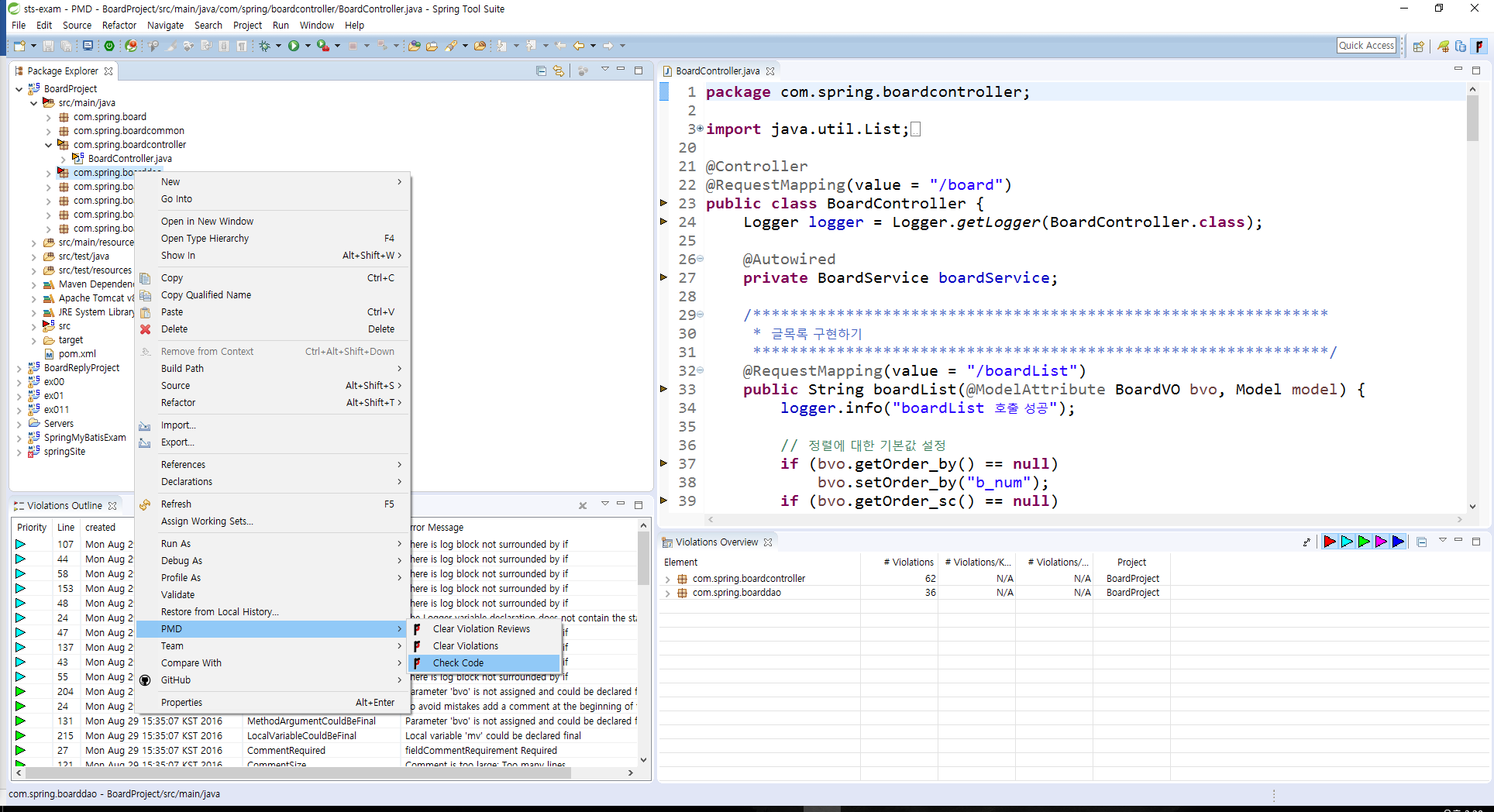
and click OK.

4. **PMD for Eclipse** 선택(버전 높은 것 선택)

5. Restart eclipse.

[PMD 실행] Open Perspective로 추가

1. 검사할 위치(패키지나 파일) 클릭 후 우클릭 PMD > Check Code 클릭



- 빨강 (High : Blocker) : 심각한 버그가 발생할 수 있는 코드이기 때문에 반드시 Rule 을 준수해야 한다.

- 하늘 (Medium : High : Critical) : 심각하지는 않지만 버그가 발생할 수 있는 코드이기 때문에 Rule 을 준수해야 한다.

- 초록 (Medium : Urgent) : 복잡한 코딩, Best Practice 및 보안, 성능 등에 관련 된 내용으로 준수 할 것을 권장한다.

- 분홍 (Medium Low : Important) : 버그가 아니며 표준, 코딩 스타일, 불필요한 코드 및 미사용 코드에 관련된 내용이다.

- 파랑 (Low : Warning) : 패키지, 클래스, 필드 등 Naming에 관련된 내용이다.

2. 검사결과 초기화

- 우클릭 PMD > Clear Violations

3. Report

- Window > Preferences 에서 PMD > Reports 에서 export 할 report type 선택

- 프로젝트 선택 후 우클릭 PMD > Generate Reports 클릭

- 프로젝트/reports 폴더 아래의 파일 확인

**- FindBugs Plugin 설치**

이 FindBugs라는 프로그램은 Java 소스에서 잠재적으로 버그가 일어날만 한 부분에 경고를 보여 준다.

http://findbugs.sourceforge.net/downloads.html

1. Help -> Install New Software...

2. Click on Add..

3. Enter the following:

Name: **FindBugs**

URL: **http://findbugs.cs.umd.edu/eclipse**

and click OK.

다시 한번 더

3.1 Enter the following:

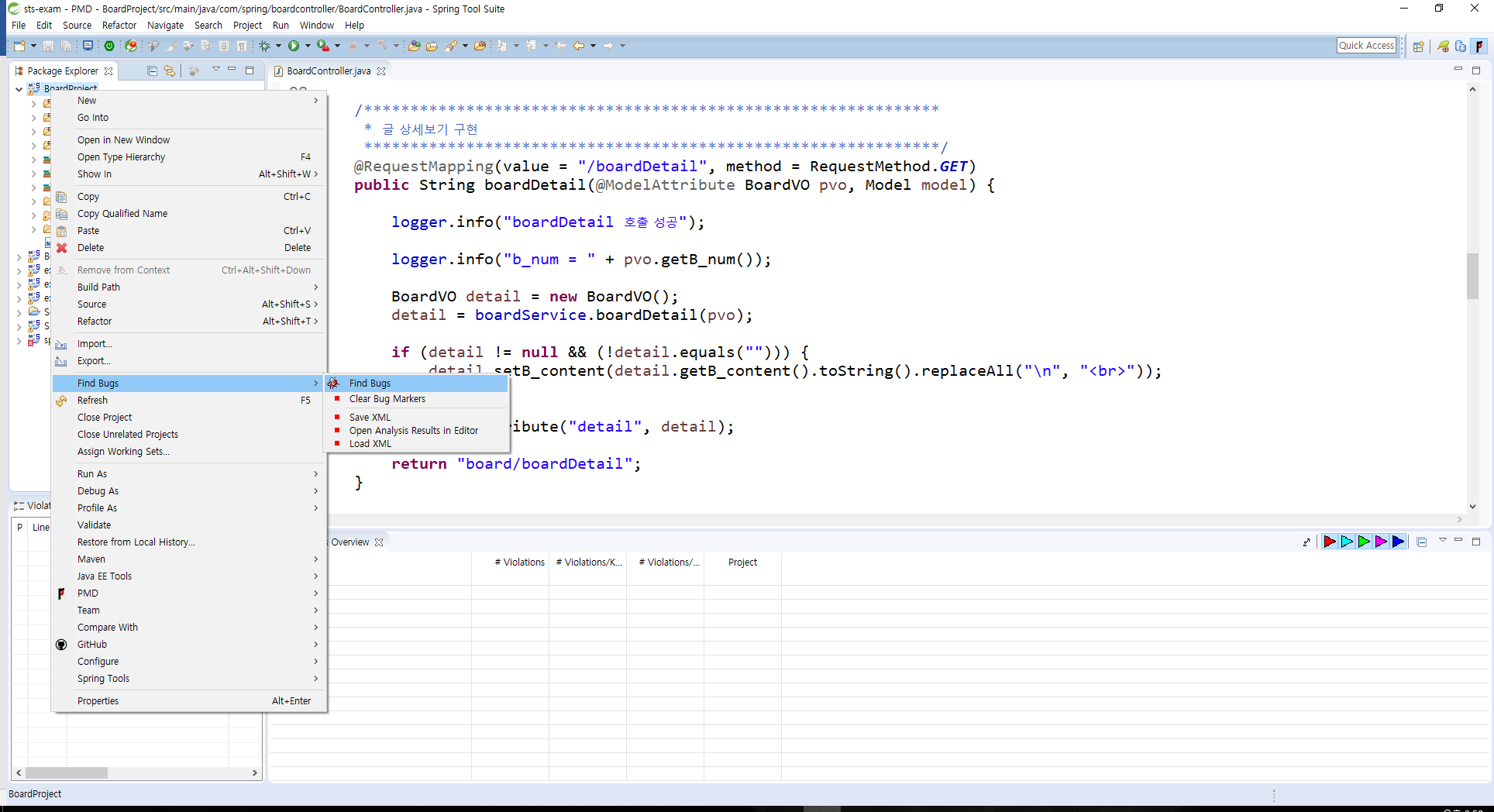
Name: **FindBugs**

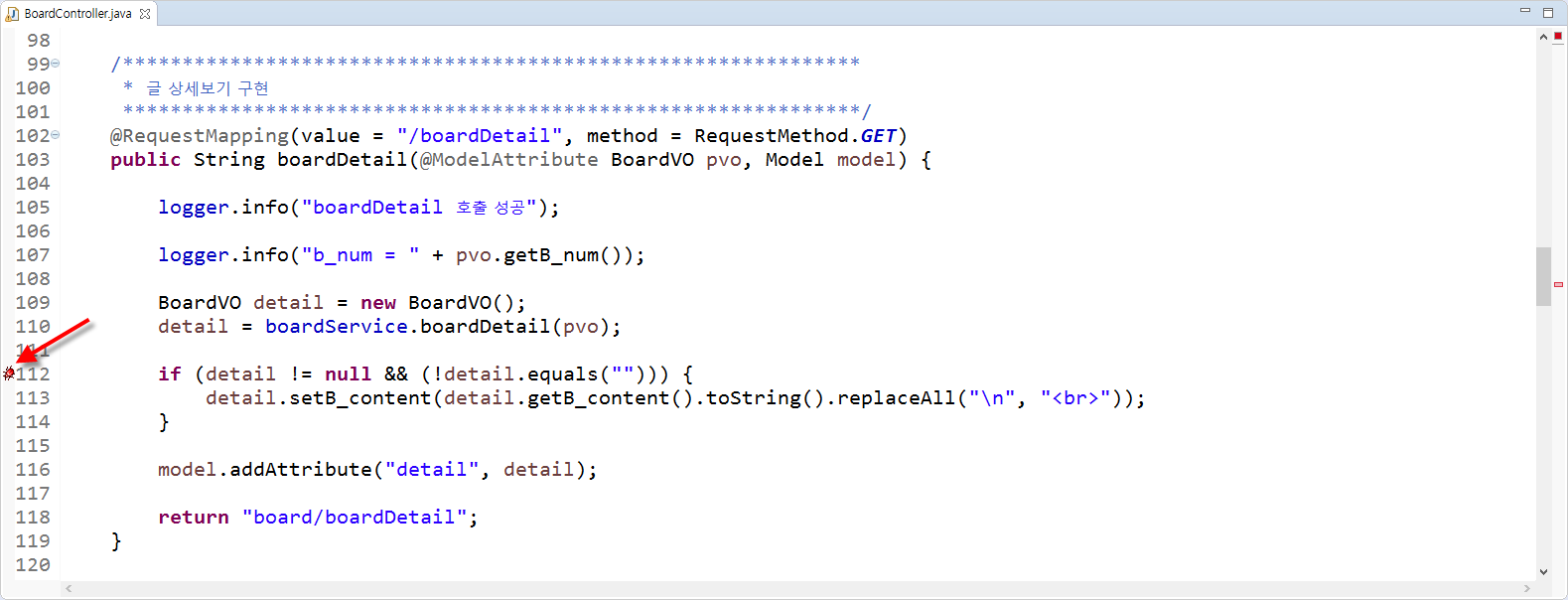
URL: **http://findbugs.cs.umd.edu/eclipse-daily**

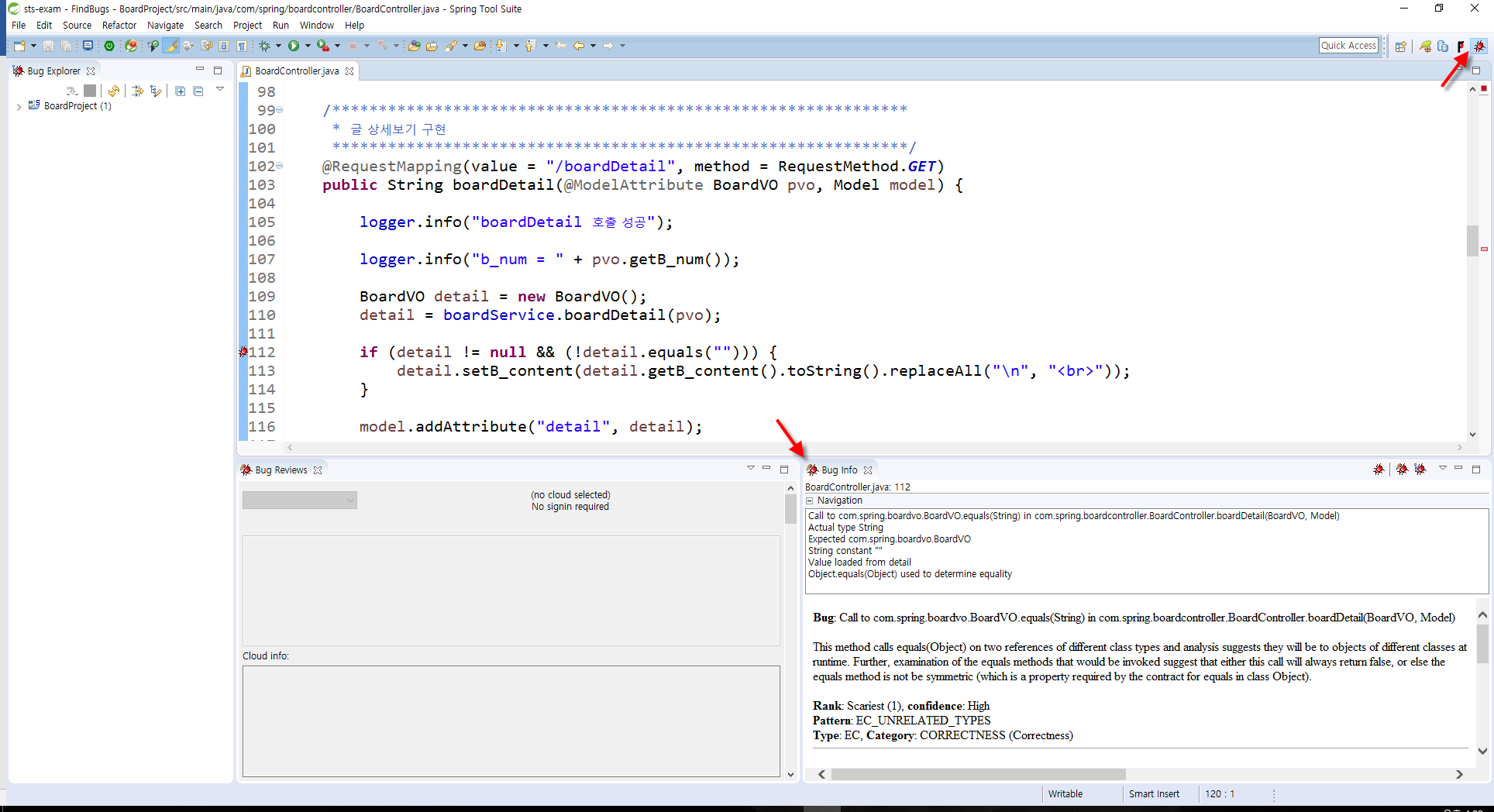
and click OK.

[FindBugs 실행] Open Perspective로 추가

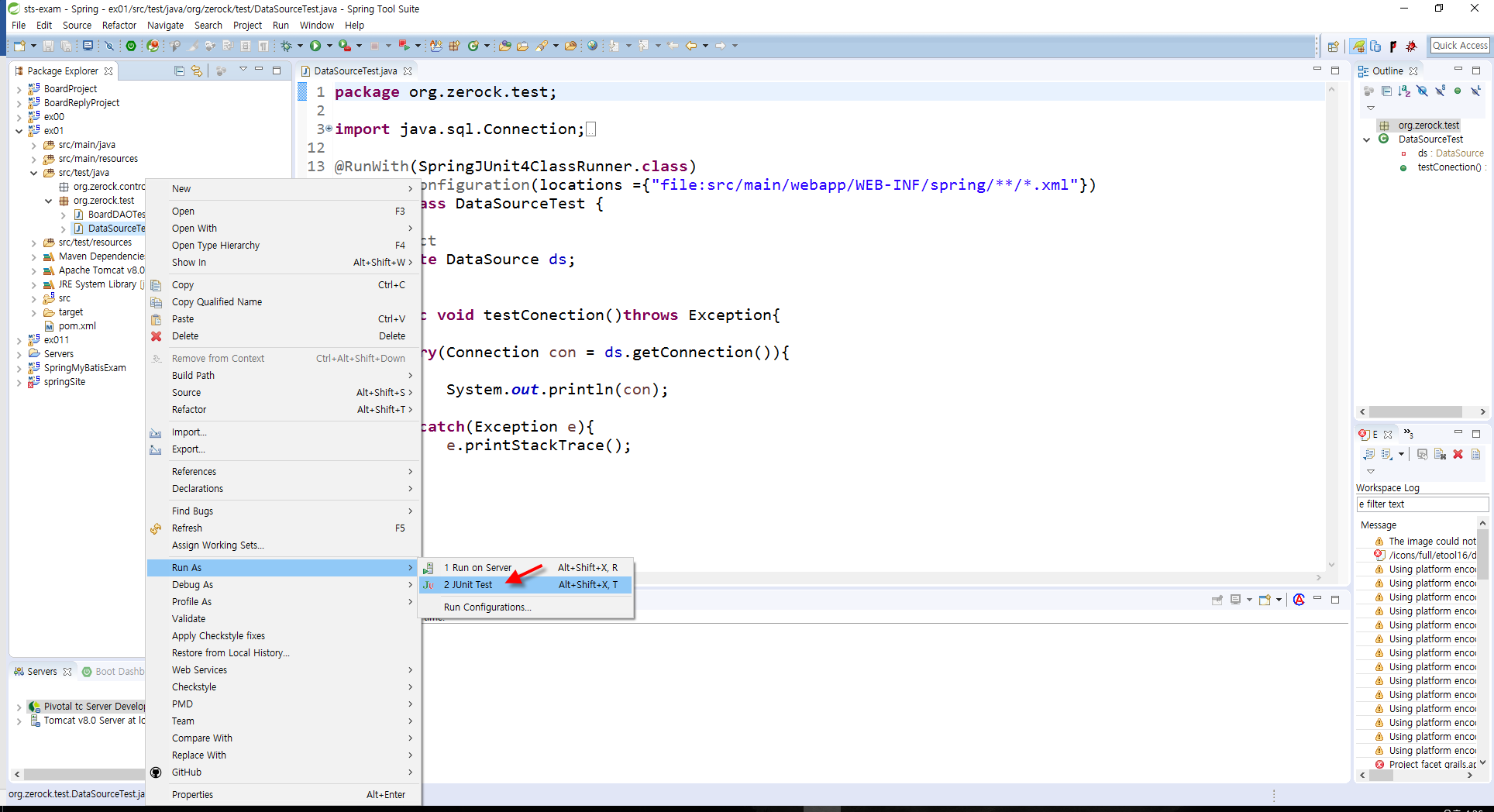
1. 검사할 위치(패키지나 파일) 클릭 후 우클릭 Find Bugs > Find Bugs 클릭

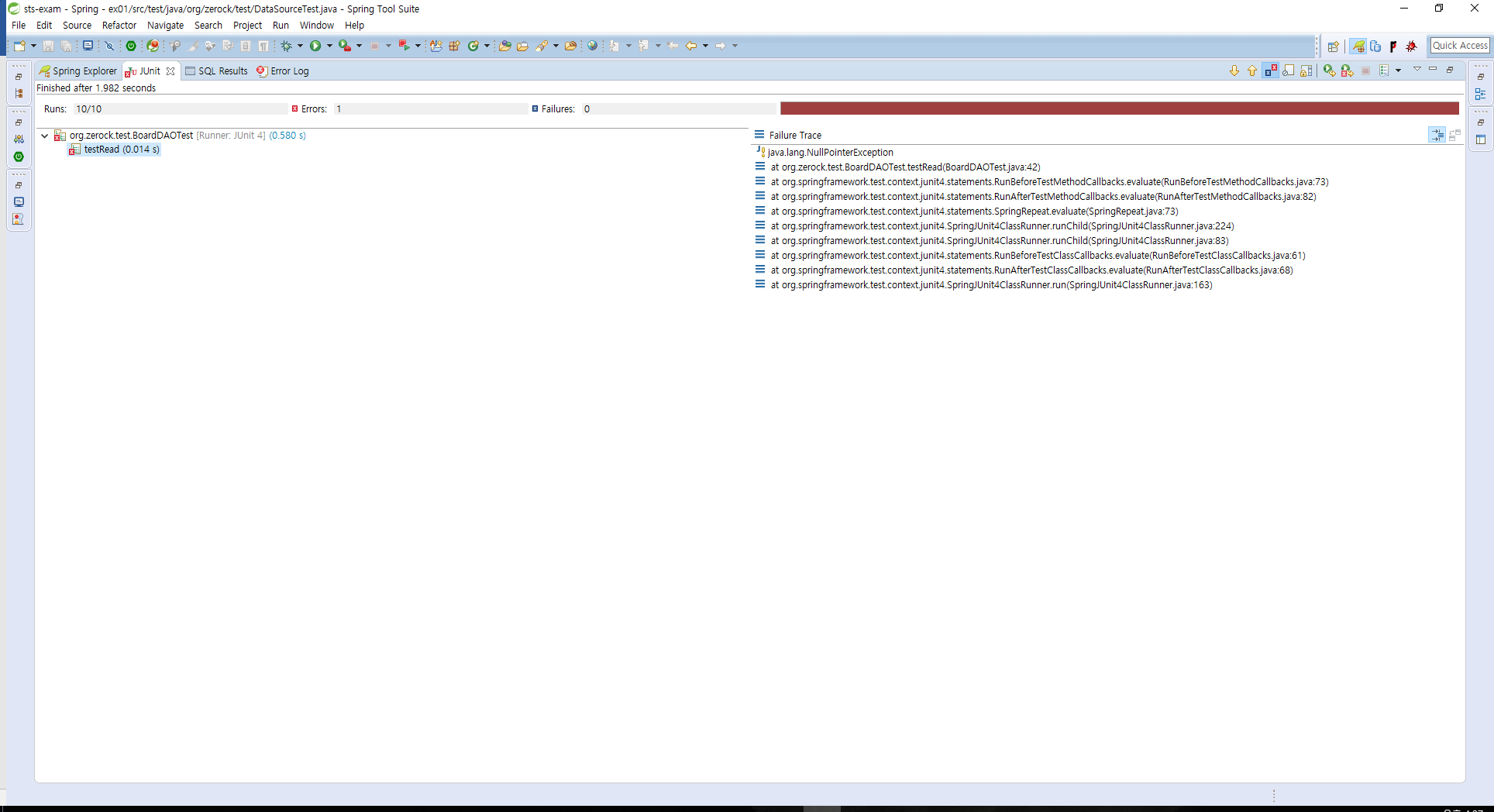






**- Junit 테스트**





**- CheckStyle Plugin 설치**

CheckStyle은 개발된 코드가 얼마나 코딩 룰을 잘 따르고 있는지 분석해주는 정적 분석 도구 입니다. 기본적으로 내장된 코딩 룰은 Google's Style과 Unix Style입니다. 하지만 조직만의 코딩 룰을 정의하고 작성된 코드들이 이를 잘 따르고 있는지 점검할 수 있다.

1. Help -> Install New Software...

2. Click on Add..

3. Enter the following:

Name: **CheckStyle**

URL: **http://eclipse-cs.sf.net/update/**

and click OK.

4. Restart Eclipse

