

Static Analysis Report for Team 7

Project Team

Team 7

Team Information

컴퓨터공학과 201111341 김성민

컴퓨터공학과 201111345 김종우

컴퓨터공학과 201211356 송원종

1 Code Scroll

2 Sonar Qube

3 PMD

1. CodeScroller

CodeScroll Code Inspector

규칙	규칙 설명
CERT_Secure_Coding_Standard	Java Platform standard edition, 6,7에 대한 시큐어 프로그래밍을 위한 Rule과 recommendation
CODESCROLL_JAVA_RULES	일반적인 자바프로그램에서 지켜야 할 규칙과 권고들
Sun_Code_Conventions_for_Java	파일이름, 파일 구성, 들여쓰기, 주석, 선언, 문장, 이름 규칙, 프로그래밍 practice등 Sun에서 따르기를 권고하는 표준 코딩 가이드라인

CodeScroll Code Inspector Violation

규칙	규칙 모음	위배 수	규칙 설명
Sun_24	Sun	82	숫자 상수 사용 금지
Sun_08	Sun	40	블록의 시작 부분에서만 선언 검사
Sun_16	Sun	38	메서드 선언 사이를 빈칸으로 구분
Sun_20	Sun	34	필드 이름 검사
Sun_19	Sun	32	메서드 이름 검사

Sun_04	Sun	14	파일에 package 선언이 있는지 검사
Sun_06	Sun	14	소스 라인 길이 검사
Sun_03	Sun	12	소스 파일 시작의 C 스타일 주석 검사
Sun_14	Sun	10	한 줄에 한 문장만 존재하는지 검사
Sun_15	Sun	10	class나 interface 선언 사이를 빈칸으로 구분
Sun_10	Sun	7	지역 변수 선언 시 초기화 검사
Sun_17	Sun	7	메서드 내부의 지역 변수 선언부와 실행 문장 사이를 빈 줄로 구분
Sun_26	Sun	5	괄호 안의 수식에 연산자 혼용 금지
Sun_05	Sun	3	클래스 또는 인터페이스에서 멤버의 선언 순서 검사
Sun_13	Sun	1	Switch 문에 default case가 존재하는지 검사
Sun_09	Sun	1	필드 hide 제한
MET05-J	CERT_SECURE	73	생성자에서 override 가능한 메서드를 호출하면 안 된다
OBJ04-J	CERT_SECURE	27	mutable 클래스(regular expression으로 지정한 이름의 클래스. Default: all)는 신뢰할 수 없는 코드에 객체를 안전하게 전달하기 위한 복사 기능을 제공해야 한다
OBJ10-J	CERT_SECURE	14	public static nonfinal 멤버를 사용하면 안 된다
OBJ01-J	CERT_SECURE	14	데이터 멤버(non-static, non-final 필드)는 private 으로 선언하라.
SER05-J	CERT_SECURE	13	내부(inner) 클래스들의 인스턴스를 직렬화하지 마라.
ERR02-J	CERT_SECURE	2	로그를 기록할 때는 예외가 발생하지 않도록 하라
JAVA_72	CODESROLL_JAVA	70	메서드의 설명 주석이 있어야 함
JAVA_71	CODESROLL_JAVA	14	클래스의 설명 주석이 있어야 함
JAVA_49	CODESROLL_JAVA	13	static, local, anonymous가 아닌 내부 클래스 사용 금지
JAVA_68	CODESROLL_JAVA	3	* 을 이용한 import 구문의 사용 금지
JAVA_64	CODESROLL_JAVA	2	반복문 안에서 String concatenation assignmet 연산자의 사용 금지
JAVA_61	CODESROLL_JAVA	1	switch 구문 안의 default 검사
JAVA_70	CODESROLL_JAVA	1	변수 hiding 금지

Inspection Violation Summation – Rules

규칙 모음	포함된 규칙 수	위배 수	무시된 위배 수	설명
Sun_Code_Conventions_for_Java	27	310	0	파일이름, 파일 구성, 들여쓰기, 주석, 선언, 문장, 이름 규칙, 프로그래밍 practice등 Sun에서

				따르기를 권고하는 표준 코딩 가이드 라인
CERT_Secure_Coding_Standard	99	143	0	Java Platform standard edition 6,7에 대한 시큐어 프로그래밍을 위한 Rule과 recommendation 제시
CODESCROLL_JAVA_RULES	72	104	0	일반적인 자바프로그램에서 지켜야 할 규칙과 권고들

Total: 557 = CERT(143) + CODESCROLL(104) + SUN(310)

Inspection Violation Summation – Source Files

소스	위배 수	무시된 위배 수	위배 규칙 수	*RCR (%)	**RVD	파일 경로
Mainview.java	225	0	19	90.4	0.52	C:\Users\User\git\SM4\src\main\java
Controller.java	85	0	15	92.42	0.46	C:\Users\User\git\SM4\src\main\java
Payback.java	66	0	16	91.92	0.73	C:\Users\User\git\SM4\src\main\java
Bank.java	43	0	17	91.41	0.54	C:\Users\User\git\SM4\src\main\java
Account.java	28	0	10	94.95	1.03	C:\Users\User\git\SM4\src\main\java
Commission.java	23	0	10	94.95	0.74	C:\Users\User\git\SM4\src\main\java
PrintStatement.java	20	0	13	93.43	0.68	C:\Users\User\git\SM4\src\main\java
Send.java	17	0	12	93.94	1.06	C:\Users\User\git\SM4\src\main\java
CheckRemain.java	14	0	9	95.45	0.87	C:\Users\User\git\SM4\src\main\java
Withdraw.java	11	0	9	95.45	0.68	C:\Users\User\git\SM4\src\main\java
Deposit.java	9	0	07	96.46	0.81	C:\Users\User\git\SM4\src\main\java
LibraryTest.java	6	0	6	96.97	0.75	C:\Users\User\git\SM4\src\Test\java
Main.java	6	0	6	96.97	0.75	C:\Users\User\git\SM4\src\main\java
Library.java	4	0	4	97.98	0.8	C:\Users\User\git\SM4\src\main\java

2. SonarQube

Bugs & Weakness

Bugs	Weakness	New Bugs	New Weakness
3	35	3	35

Stink Codes

Liability	Stinking Codes	New Liability	New Stinking
-----------	----------------	---------------	--------------

			Codes
8 Days	332	7 Days	329

Coverage

Coverage	Coverage on new Code
0.0 %	0.0 %

Code reduplication

Code Reduplication	Duplicated blocks	Code Reduplication New Code lines
2.7 %	2	2.7 % 1.3K New Code Lines

3. PMD

Rule	Description
Basic rulesets/basic.xml	대부분의 개발자들이 동의하는 규칙: catch 블록들은 비어있어서는 안되고, equals()를 오버라이딩 할 때 마다 hashCode()를 오버라이드한다.
Naming rulesets/naming.xml	표준 자바 네이밍 규약을 위한 테스트: 변수 이름들은 너무 짧아서 안된다; 메소드 이름은 너무 길어서 안된다. 클래스 이름은 대문자로 시작해야 하고, 메소드와 필드 이름들은 소문자로 시작해야 한다.
Unused code rulesets/unusedcode.xml	결코 읽히지 않은 프라이빗 필드와 로컬 변수, 접근할 수 없는 문장, 결코 호출되지 않는 프라이빗 메소드 등을 찾기
Design rulesets/design.xml	다양한 좋은 디자인 원리 체크, 이를 테면: switch 문장은 default 블록을 갖고 있어야 하고, 심하게 중첩된 if 블록은 피해야 하고, 매개변수들은 재할당되어서는 안되며, 더블(double)이 동일함(equality)과 비교되어서도 안된다
Import statements rulesets/imports.xml	임포트 문장에 대한 작은 문제들 점검. 같은 클래스를 두 번 반입하는 것이나 java.lang 에서 클래스를 임포트하는 것 등
JUnit tests rulesets/junit.xml	테스트 케이스와 테스트 메소드 관련 특정 문제 검색. 메소드 이름의 정확한 스펠링과 suite() 메소드가 정적이고 퍼블릭인지 여부
Strings rulesets/string.xml	스트링 관련 작업을 할 때 발생하는 일반적인 문제들 규명. 스트링 리터럴 중복, String 구조체 호출, String 객체에 toString() 호출하기 등.

Braces rulesets/braces.xml	for, if, while, else 문장이 괄호를 사용하는지 여부 검사.
Code size rulesets/codesize.xml	과도하게 긴 메소드, 너무 많은 메소드를 가진 클래스, 리팩토링에 대한 유사한 후보들을 위한 테스트.
Javabeans rulesets/javabeans.xml	직렬화 될 수 없는 bean 클래스 같이 JavaBeans 코딩 규약을 위배하는 JavaBeans 컴포넌트 검사.
Finalizers	finalize() 메소드는 자바에서 일반적인 것은 아니기 때문에 사용법에 대한 규칙이 비교적 익숙하지 않다. 이 그룹의 검사는 finalize() 메소드 관련한 다양한 문제들을 찾는다. 이를테면, 비어있는 finalizer, 다른 메소드를 호출하는 finalize() 메소드 finalize()로의 호출 등이 그것이다
Clone rulesets/clone.xml	clone() 메소드에 대한 규칙: clone()을 오버라이드하는 클래스는 Cloneable 을 구현해야 하고, clone() 메소드는 super.clone()을 호출해야 하며, clone() 메소드는 실제로 던지지 않더라도 CloneNotSupportedException 을 던지도록 선언되어야 한다.
Coupling rulesets/coupling.xml	클래스들간 과도한 커플링 표시 검색. 지나치게 많은 임포트, supertype 또는 인터페이스가 충분한 곳에서 subclass 유형 사용하기, 너무 적은 필드, 변수, 클래스 내의 리턴 유형 등.
Strict exceptions rulesets/strictexception.xml	예외 테스트: 메소드는 java.lang.Exception 을 던지도록 선언되어서는 안되고, 예외는 플로우 제어에 사용되어서는 안되며, Throwable 은 잡혀서는 안된다.
Controversial rulesets/controversial.xml	일부 PMD 규칙들은 유능한 자바 프로그래머가 받아들일 수 있는 것들이다. 하지만 어떤 것은 논쟁의 여지가 충분하다. 이 규칙 세트에는 좀더 의심스러운 검사들이 포함되어 있다. 변수에 null 할당하기, 메소드에서 온 다중의 리턴 포인트 sun 패키지에서 임포팅 등이 포함된다
Logging rulesets/logging-java.xml	java.util.logging.Logger 를 위험하게 사용하는 경우 검색: 끝나지 않고 정적이지 않은 logger 와 한 클래스에 한 개 이상의 logger 등

PMD report

Problems found

#	File	Line	Problem
1	Account.java	2	All classes and interfaces must belong to a named package
2	Account.java	2	Each class should declare at least one constructor
3	Account.java	2	headerCommentRequirement Required
4	Account.java	3	Avoid unused private fields such as 'Account'.
5	Account.java	3	Field Account has the same name as a method
6	Account.java	3	It is somewhat confusing to have a field name matching the declaring class name
7	Account.java	3	Variables should start with a lowercase character, 'Account' starts with uppercase character.
8	Account.java	3	fieldCommentRequirement Required
9	Account.java	4	Field Password has the same name as a method
10	Account.java	4	Variables should start with a lowercase character, 'Password' starts with uppercase character.
11	Account.java	4	fieldCommentRequirement Required
12	Account.java	5	Only variables that are final should contain underscores (except for underscores in standard prefix/suffix), 'Total_Amount' is not final.
13	Account.java	5	Variables should start with a lowercase character, 'Total_Amount' starts with uppercase character.
14	Account.java	5	fieldCommentRequirement Required
15	Account.java	6	Only variables that are final should contain underscores (except for underscores in standard prefix/suffix), 'Limit_Amount' is not final.
16	Account.java	6	Variables should start with a lowercase character, 'Limit_Amount' starts with uppercase character.
17	Account.java	6	fieldCommentRequirement Required

792	PrintStatement.java	45	Local variable 'c' could be declared final
793	Send.java	2	All classes and interfaces must belong to a named package
794	Send.java	2	Avoid short class names like Send
795	Send.java	2	Each class should declare at least one constructor
796	Send.java	2	headerCommentRequirement Required
797	Send.java	3	Found non-transient, non-static member. Please mark as transient or provide accessors.
798	Send.java	3	To avoid mistakes add a comment at the beginning of the commission field if you want a default access modifier
799	Send.java	3	Use explicit scoping instead of the default package private level
800	Send.java	3	fieldCommentRequirement Required
801	Send.java	4	Variables should start with a lowercase character, 'Amount' starts with uppercase character.
802	Send.java	4	fieldCommentRequirement Required
803	Send.java	5	Only variables that are final should contain underscores (except for underscores in standard prefix/suffix), 'Receiver_Account' is not final.
804	Send.java	5	Variables should start with a lowercase character, 'Receiver_Account' starts with uppercase character.
805	Send.java	5	fieldCommentRequirement Required
806	Send.java	7	Method names should not contain underscores
807	Send.java	7	Parameter 'amount' is not assigned and could be declared final
808	Send.java	7	To avoid mistakes add a comment at the beginning of the get_Amount method if you want a default access modifier
809	Send.java	7	Use explicit scoping instead of the default package private level
810	Send.java	13	Method names should not contain underscores
811	Send.java	13	To avoid mistakes add a comment at the beginning of the send_Amount method if you want a default access modifier
812	Send.java	13	Use explicit scoping instead of the default package private level
813	Withdraw.java	2	All classes and interfaces must belong to a named package
814	Withdraw.java	2	Each class should declare at least one constructor
815	Withdraw.java	2	headerCommentRequirement Required
816	Withdraw.java	3	Variables should start with a lowercase character, 'Amount' starts with uppercase character.
817	Withdraw.java	3	fieldCommentRequirement Required
818	Withdraw.java	5	Method names should not contain underscores
819	Withdraw.java	5	Parameter 'amount' is not assigned and could be declared final
820	Withdraw.java	5	To avoid mistakes add a comment at the beginning of the get_Amount method if you want a default access modifier
821	Withdraw.java	5	Use explicit scoping instead of the default package private level
822	Withdraw.java	11	Method names should not contain underscores
823	Withdraw.java	11	To avoid mistakes add a comment at the beginning of the send_Amount method if you want a default access modifier
824	Withdraw.java	11	Use explicit scoping instead of the default package private level
825	Withdraw.java	13	Local variable 'commission' could be declared final

Element	# Violations	# Violations/K	# Violations/...	Project
▼ (default package)	825	951.6	14.22	New
> Account.java	26	1238.1	5.20	New
> Bank.java	65	1048.4	32.50	New
> CheckRemain.java	20	2500.0	6.67	New
> Commission.java	38	1583.3	6.33	New
> Controller.java	180	1374.0	8.18	New
> Deposit.java	12	1500.0	6.00	New
> Library.java	4	2000.0	4.00	New
> Main.java	9	2250.0	9.00	New
> Mainview.java	343	686.0	114.33	New
> Payback.java	52	928.6	10.40	New
> PrintStatement.java	43	1653.8	10.75	New
> Send.java	20	1538.5	10.00	New
> Withdraw.java	13	1083.3	6.50	New