

2018 SOFTWARE VERIFICATION

Final Presentation

2018.06.11 (월) SV T1

201311263 김민환

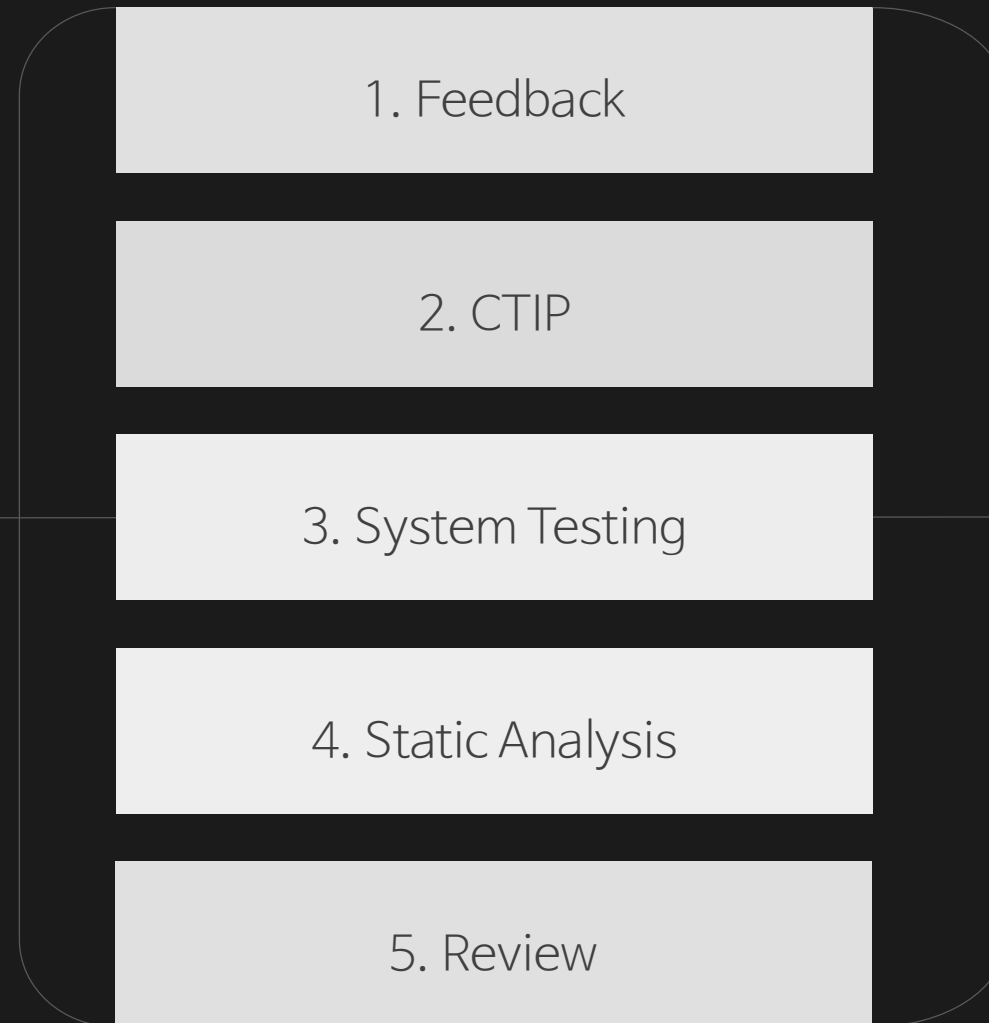
201311308 전세진

201411278 서희진

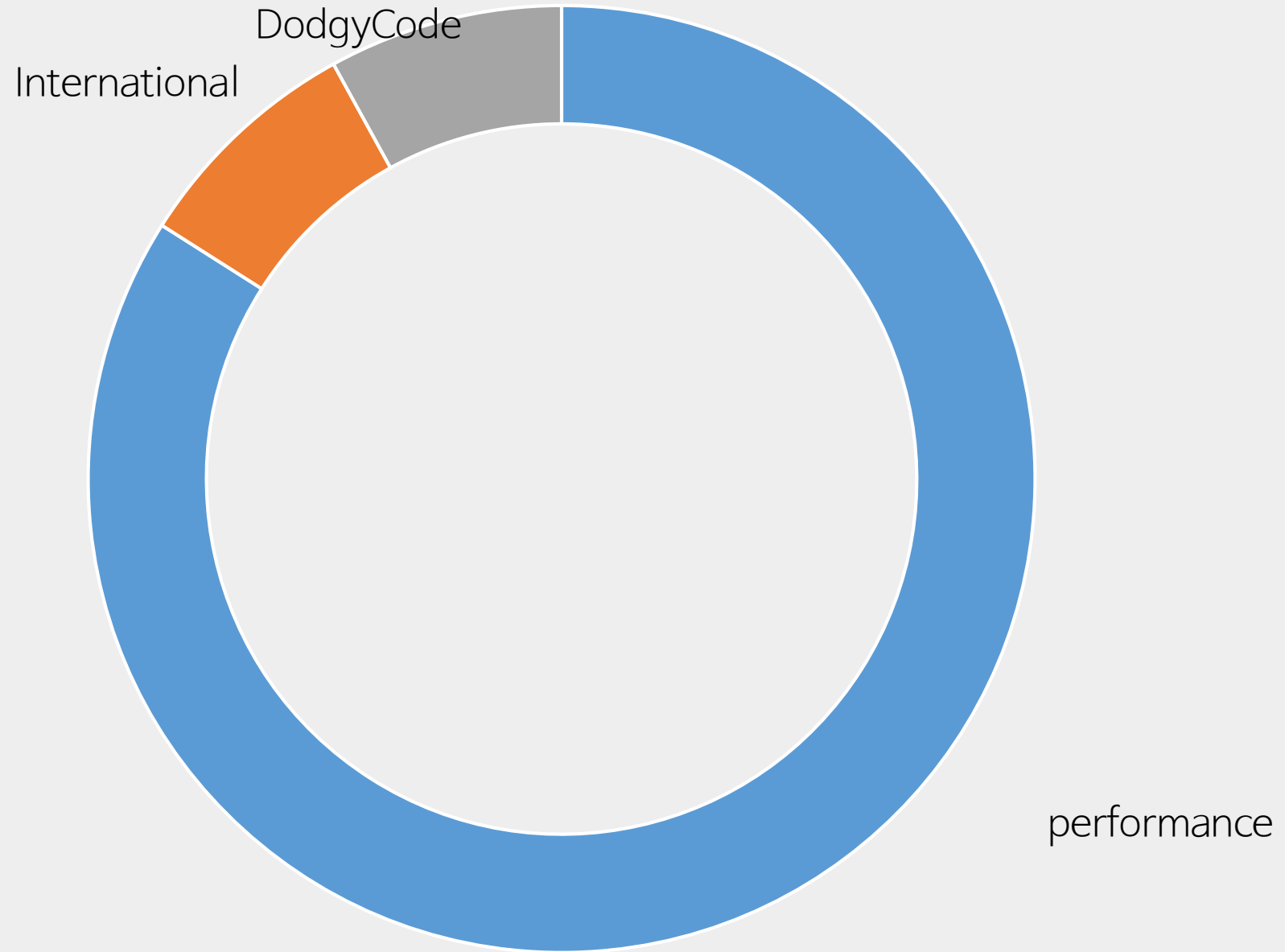
201411317 조민규

1st Testing

2018 SOFTWARE VERIFICATION



01. Feedback



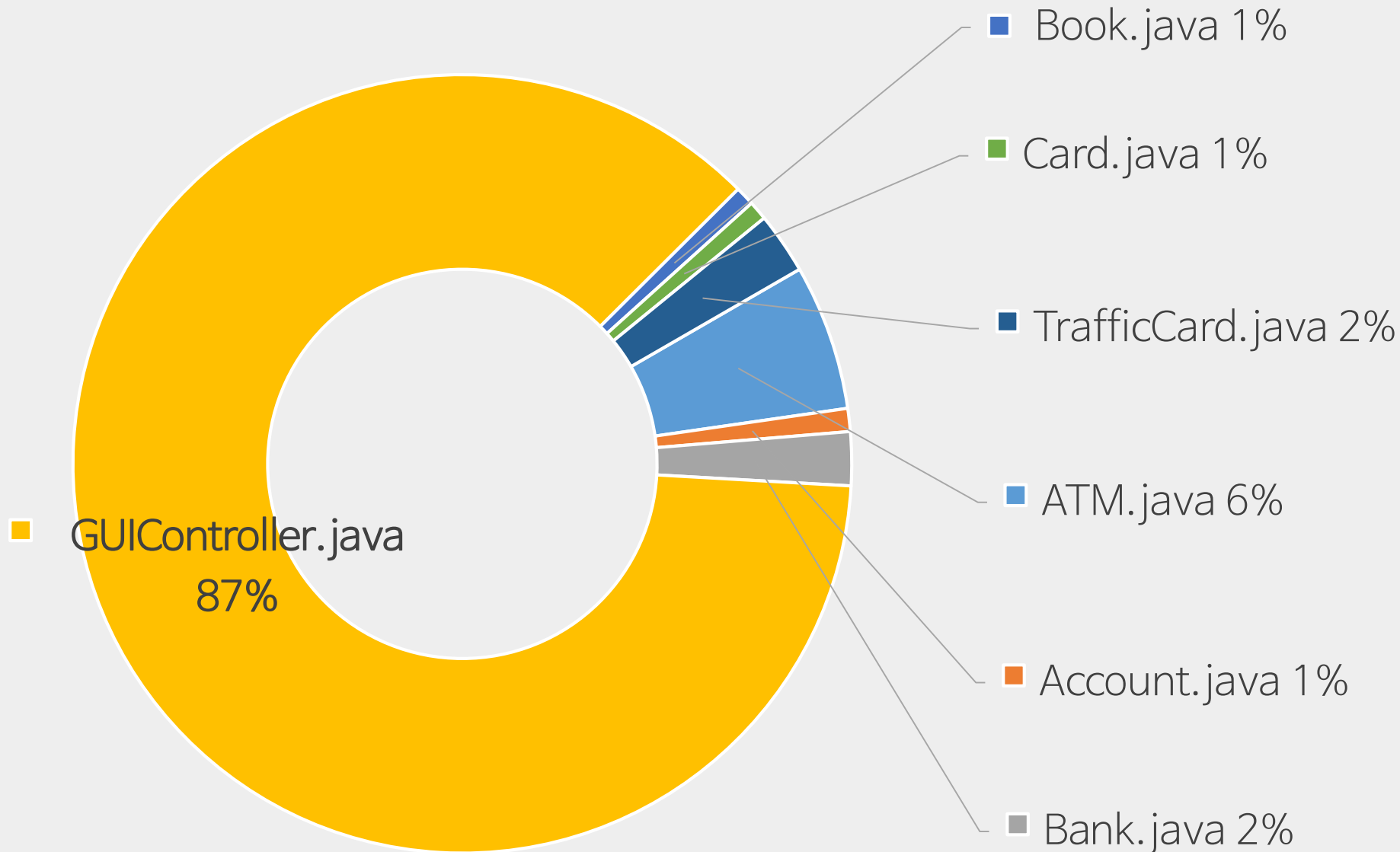
Best Practices

Rules which enforce generally accepted best practices.

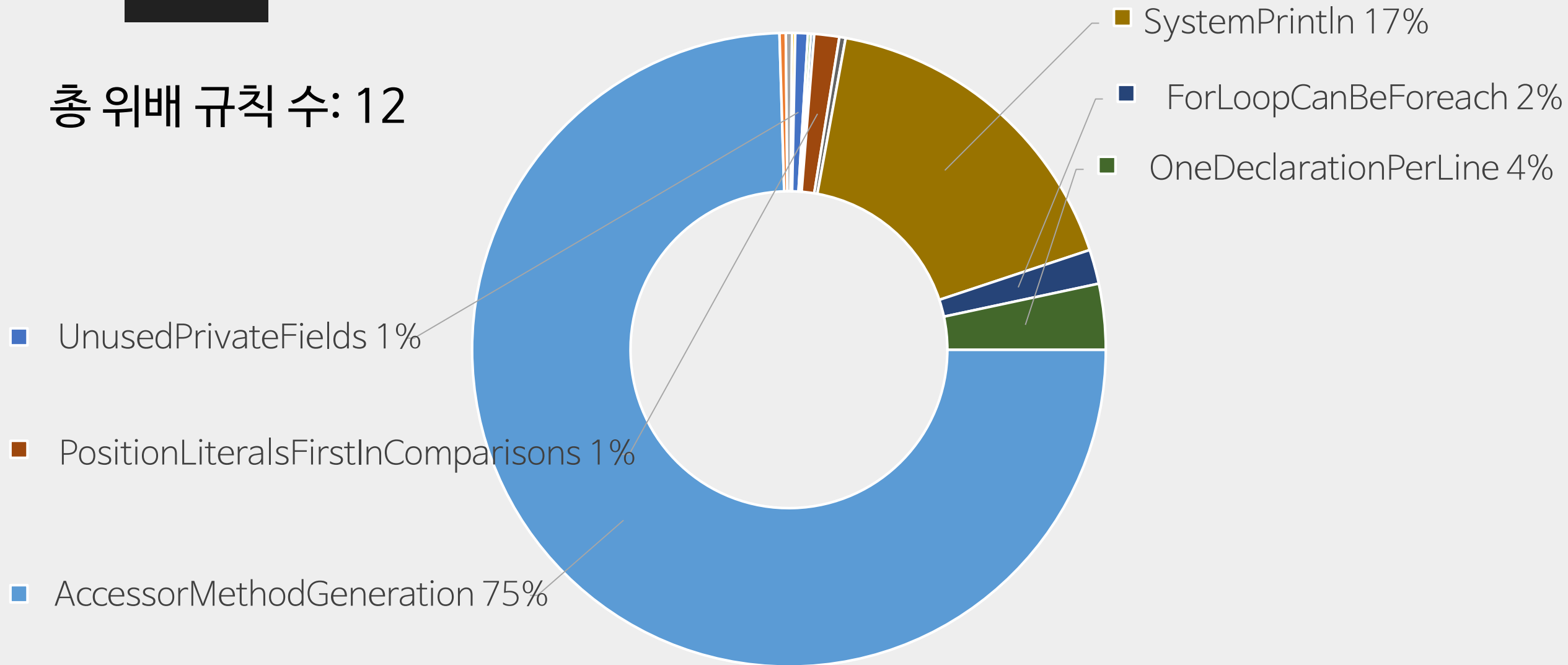
- [AbstractClassWithoutAbstractMethod](#): The abstract class does not contain any abstract methods. An abstract class suggests an incomplete...
- [AccessorClassGeneration](#): Instantiation by way of private constructors from outside of the constructor's class often causes...
- [AccessorMethodGeneration](#): When accessing a private field / method from another class, the Java compiler will generate a acc...
- [ArraysStoredDirectly](#): Constructors and methods receiving arrays should clone objects and store the copy. This prevents f...
- [AvoidPrintStackTrace](#): Avoid `printStackTrace()`; use a logger call instead.
- [AvoidReassigningParameters](#): Reassigning values to incoming parameters is not recommended. Use temporary local variables inst...
- [AvoidStringBufferField](#): `StringBuffers/StringBuilders` can grow considerably, and so may become a source of memory leaks if ...

43개의 'Best Practice' 들을 Rule으로 적용

총 위배 수: 628



총 위배 규칙 수: 12



02. CTIP

Github Issue Tracker



sonarqube

GitHub



Pmd





간편한 진행상황 공유

빌드 및 테스트 자동화

환경구성에 많은 노력 소모

사용자 역량에 따라 활용성이 좌우



프로젝트 규모 등 외적 요인에 따라

CTIP 사용 여부, CTIP 환경의 규모 등을 적절하게 조정해야 한다

CTIP 실습 개선안

문서 작성이 끝난 후 Issue Tracker에 Issue 등록 후 P/F만 입력하여 효율성을 높인다.

Git Hub를 활용한 협업 경험이 부족한 학생들을 위한 협업 관련 수업 및 실습 진행한다.

SMA학생들에게 '검증'에 대한 간단한 개념 설명을 통해 검증팀과의 소통 및 이해를 높인다.

03. System Testing

CPT

카테고리 분류의 어려움

하나의 예외만으로 테스트케이스가 완전히 실패



Pairwise Testing

CPT에 비해 테스트케이스 간소



무식한 이름과 달리 높은 적중률

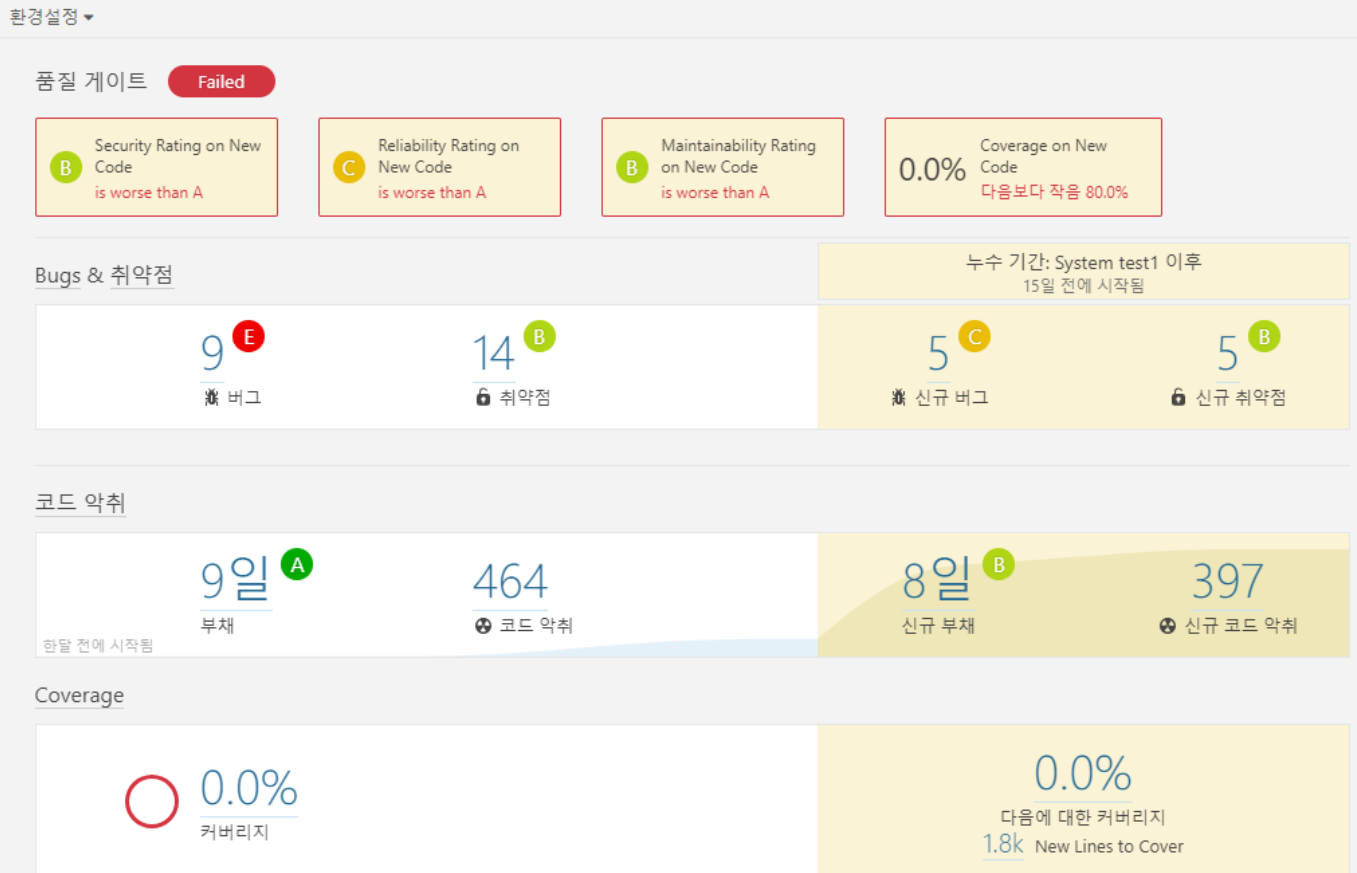
다른 테스트의 약점을 보완

04. Static Analysis



코드에 일관성을 더하여 **가독성**을 높여줌
에러를 유발할 수 **버그들**을 잡아줌

코드 작성 전 **coding rule**에 대한 **합의** 필요
Code smell 등이 생각보다 많음



프로젝트에 대한 전체적인 품질을 요약

Coverage를 통해 Testing의 방향성 제시

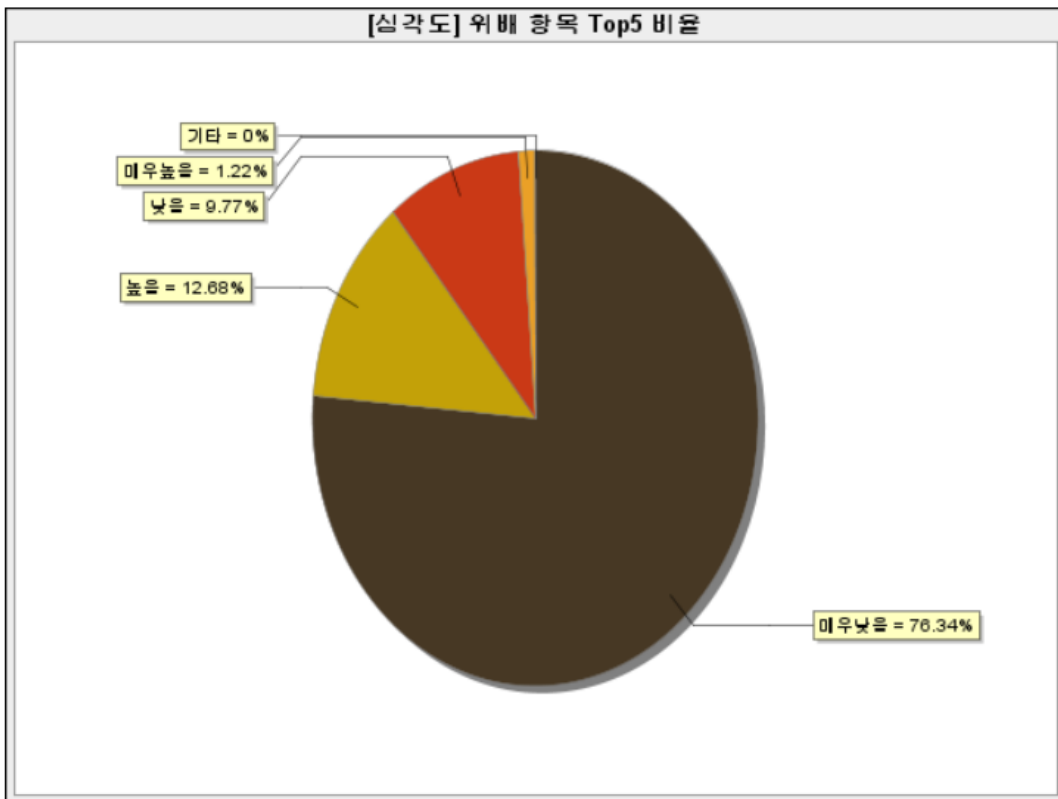
플러그인 시스템으로 사용이 간편

04

인스펙션 위배 요약 - 심각도

main

2018-05-30 14:22:42



심각도	위배 수	무시된 위배 수	포함된 규칙 수	위배 규칙 수	*SCR	설명
매우높음	13	0	20	1	95%	프로그램에 매우 심각한 영향을 미치므로 오류 수정이 필요함
높음	135	0	46	6	86.96%	프로그램에 심각한 영향을 줄 수 있으므로 오류 수정이 강력히 권장됨
낮음	104	0	10	4	60%	프로그램에 영향을 줄 수 있으므로 오류 수정이 권장됨
매우낮음	813	0	23	14	39.13%	프로그램에 큰 영향은 없지만 오류 수정이 권장됨
기타	0	0	0	0	0%	심각도가 정의되지 않음

다양한 항목에 대한 위배 내역 제공

위배 항목에 대한 친절한 설명

직관적인 UI를 통해 손쉽게 파악 가능

예시 코드를 통해 손쉽게 위배내역 파악 가능

```
<description> PMD test </description>
<rule ref="category/java/bestpractices.xml/AbstractClassWithoutAbstractMethod" />
<rule ref="category/java/bestpractices.xml/AccessorClassGeneration" />
<rule ref="category/java/bestpractices.xml/AccessorMethodGeneration" />
<rule ref="category/java/bestpractices.xml/ArrayIsStoredDirectly" />
<rule ref="category/java/bestpractices.xml/AvoidPrintStackTrace" />
<rule ref="category/java/bestpractices.xml/AvoidReassigningParameters" />
<rule ref="category/java/bestpractices.xml/AvoidStringBufferField" />
<rule ref="category/java/bestpractices.xml/AvoidUsingHardCodedIP" />
<rule ref="category/java/bestpractices.xml/CheckResultSet" />
<rule ref="category/java/bestpractices.xml/ConstantsInInterface" />
<rule ref="category/java/bestpractices.xml/DefaultLabelNotLastInSwitchStmt" />
<rule ref="category/java/bestpractices.xml/ForLoopCanBeForeach" />
<rule ref="category/java/bestpractices.xml/GuardLogStatement" />
<rule ref="category/java/bestpractices.xml/JUnit4SuitesShouldUseSuiteAnnotation" />
<rule ref="category/java/bestpractices.xml/JUnit4TestShouldUseAfterAnnotation" />
<rule ref="category/java/bestpractices.xml/JUnit4TestShouldUseBeforeAnnotation" />
<rule ref="category/java/bestpractices.xml/JUnit4TestShouldUseTestAnnotation" />
<rule ref="category/java/bestpractices.xml/JUnitAssertionsShouldIncludeMessage" />
<rule ref="category/java/bestpractices.xml/JUnitTestsShouldIncludeAssert" />
<rule ref="category/java/bestpractices.xml/JUnitUseExpected" />
<rule ref="category/java/bestpractices.xml/LooseCoupling" />
<rule ref="category/java/bestpractices.xml/MethodReturnsInternalArray" />
<rule ref="category/java/bestpractices.xml/OneDeclarationPerLine" />
<rule ref="category/java/bestpractices.xml/PositionLiteralsFirstInCaseInSensitive" />
<rule ref="category/java/bestpractices.xml/PositionLiteralsFirstInComparisons" />
<rule ref="category/java/bestpractices.xml/PreserveStackTrace" />
<rule ref="category/java/bestpractices.xml/ReplaceEnumerationWithIterator" />
<rule ref="category/java/bestpractices.xml/ReplaceHashtableWithMap" />
<rule ref="category/java/bestpractices.xml/ReplaceVectorWithList" />
<rule ref="category/java/bestpractices.xml/SwitchStmtsShouldHaveDefault" />
<rule ref="category/java/bestpractices.xml/SystemPrintln" />
<rule ref="category/java/bestpractices.xml/UnusedFormalParameter" />
<rule ref="category/java/bestpractices.xml/UnusedImports" />
<rule ref="category/java/bestpractices.xml/UnusedLocalVariable" />
```

명명 규칙, 코드 길이 등에 대한 위배 내역

참재된 에러, 위험요소 등을 사전 탐지

Rule을 일일이 지정해주어야 함

요약 정보 따로 생성해주지 않음

Metrics

2530 lines of code analyzed, in 57 classes, in 4 packages.

Metric	Total	Density*
High Priority Warnings	12	4.74
Medium Priority Warnings	84	33.20
Total Warnings	96	37.94

(* Defects per Thousand lines of non-commenting source statement)

Summary

Warning Type	Number
Internationalization Warnings	10
Performance Warnings	76
Dodgy code Warnings	10
Total	96

성능, 위배 등 항목 별로 다양한 Warning 제공

High, Medium, Low로 우선 순위 부여

유료 도구에 비해 직관적이지 않은 요약

다른 도구에 비해 상대적으로 떨어지는 성능

05. Summary

아는 것

VS

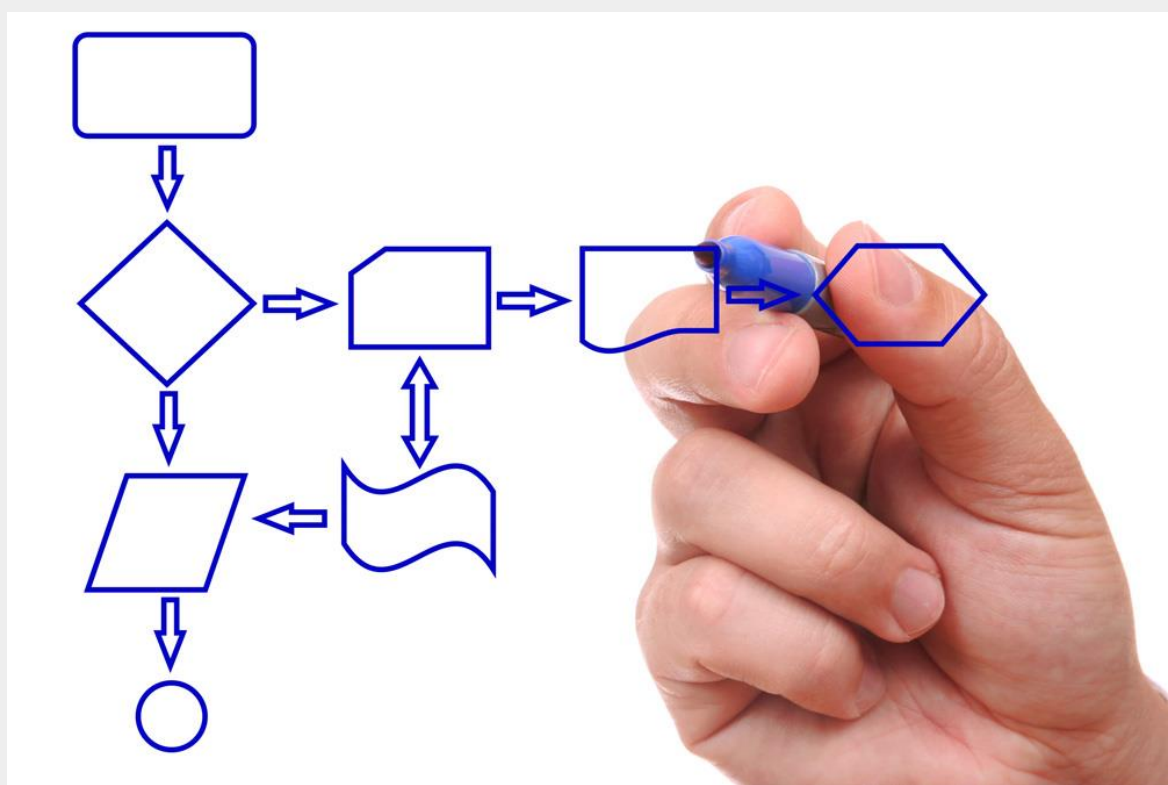
적용하는 것





경험의 중요성

Model-based Categorization



Procedure-based Categorization

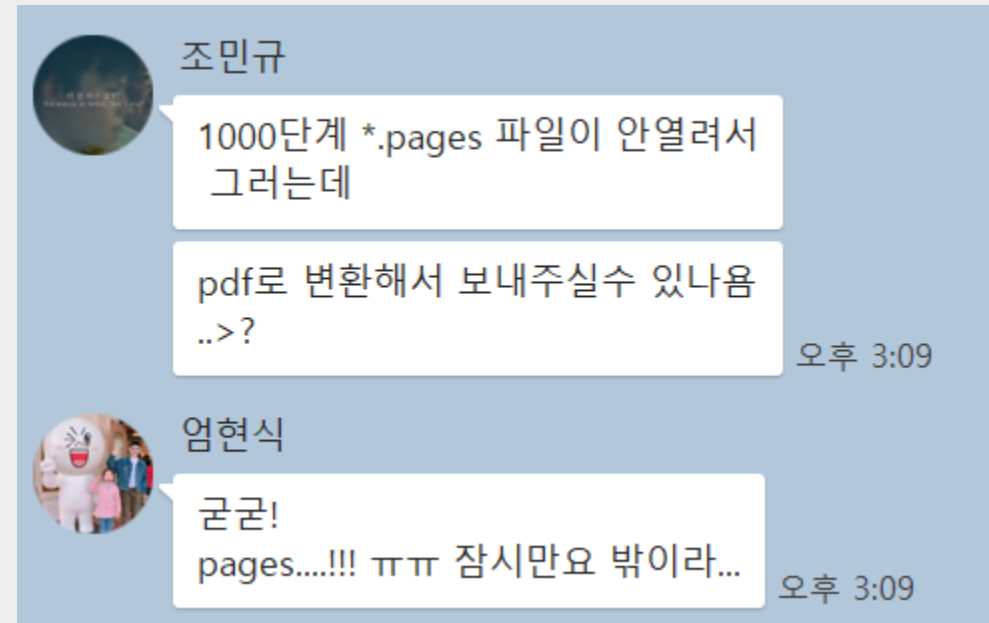
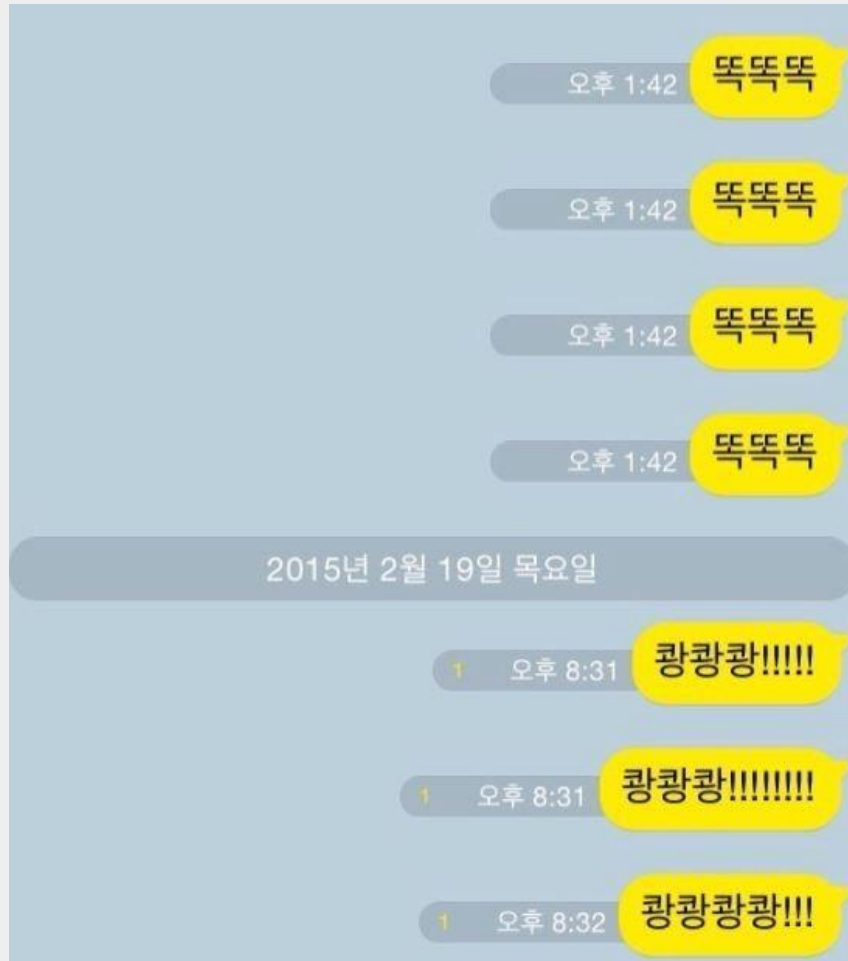


재미

```
java.io.FileNotFoundException: TrafficCard.txt (지정된 파일을 찾을 수 없습니다)
  at java.base/java.io.FileInputStream.open0(Native Method)
  at java.base/java.io.FileInputStream.open(Unknown Source)
  at java.base/java.io.FileInputStream.<init>(Unknown Source)
  at java.base/java.io.FileReader.<init>(Unknown Source)
  at Item.TrafficCard.<init>(TrafficCard.java:43)
  at ATM.ATM.<init>(ATM.java:16)
  at GUI.GUIController.readyReadCard(GUIController.java:108)
  at GUI.GUIController.<init>(GUIController.java:101)
  at GUI.GUIController.main(GUIController.java:94)
java.io.FileNotFoundException: management.txt (지정된 파일을 찾을 수 없습니다)
  at java.base/java.io.FileInputStream.open0(Native Method)
  at java.base/java.io.FileInputStream.open(Unknown Source)
  at java.base/java.io.FileInputStream.<init>(Unknown Source)
```



좋은 마무리



커뮤니케이션



코드 품질

김민환

평소에 사용하여 보기 힘들지만, 소프트웨어 개발에 **중요한 소프트웨어 검증** 수업

부서간의 **커뮤니케이션**은 굉장히 중요하다

이후 현업에서 **V&V** 업무 수행에 **자신**

전세진

학부생 수준에서 **배움을 구하기 어려운** 지식

3학년 **SMA 수업과 연계**하여 진행하는 신선하고 재미있는 수업방식

다양한 도구를 직접 사용하는 살아있는 수업

서희진

소프트웨어 검증에 대해 배울 수 있는 유일한 수업

협업의 중요성, 팀원들의 소중함

노하우가 좀 더 쌓이면 더 좋은 결과를 뽑아낼 수 있을 것

조민규

쉽게 이용할 수 없는 유료도구인 CodeScroll을 범접
우리가 작성한 코드에 버그 / 에러들이 생각보다 많았다
안전하고 일관성 있는 프로젝트의 개발에 대해 고찰