Sonar fixer

Clement Maxime - Piorun Jordan

November 3, 2015

1 Transformation sonar

Nous avons commencé par identifier les erreurs majeures levées par sonar que nous voulions corriger. Dans cette première liste sont répertoriées les erreurs qui semblent aisément corrigeable :

Noncompliant	Compliant
`if (myClass.compareTo(arg) == -1) {...}`	`if (myClass.compareTo(arg) < 0) {...}`

`class Bar {`
`  public boolean doSomething(){...}`
`}`
`class Foo extends Bar {`
`  public boolean doSomething(){...}`
`}`
	`class Bar {`
	`  public boolean doSomething(){...}`
	`}`
	`class Foo extends Bar {`
	`  @Override`
	`  public boolean doSomething(){...}`
	`}`

`public void a(Map<String,Object> map) {`
`  for (String key : map.keySet()) {`
`  Object value = map.get(key);`
`  ...`
`  }`
`}`
	`public void a(Map<String,Object> map) {`
	`  for (Map.Entry<String,Object> entry :`
	`  map.entrySet()) {`
	`  String key = entry.getKey();`
	`  Object value = entry.getValue();`
	`  ...`
	`  }`
	`}`

`foo.equals(bar.toUpperCase());`	`foo.equalsIgnoreCase(bar);`

`switch (myVariable) {`
`  case 0: // 6 lines till next case`
`  methodCall1("");`
`  methodCall2("");`
`  methodCall3("");`
`  methodCall4("");`
`  break;`
`  case 1:`
`  ...`
`}`
	`switch (myVariable) {`
	`  case 0:`
	`  doSomething()`
	`  break;`
	`  case 1:`
	`  ...`
	`}`
	`...`
	`private void doSomething(){`
	`  methodCall1("");`
	`  methodCall2("");`
	`  methodCall3("");`
	`  methodCall4("");`
	`}`

`public class MyClass {`
`  private int foo = 42;`
 	
`  public int compute(int a) {`
`  return a * 42;`
`  }`
`}`
	`public class MyClass {`
	`  public int compute(int a) {`
	`  return a * 42;`
	`  }`
	`}`

`class A {`
`  private int field;`
`}`
	`class A {`
	`  private int field;`
	
	`  A(int field) {`
	`  this.field = field;`
	`  }`
	`}`