



VALIDATION ET VÉRIFICATION TP 2

Aymeric Pierre

Mathieu Merien



Exercise 1

to have LCC=TLL we must have this condition check:

If a method if using multiple attribute, every other method that use one of those attribute must use all of them.

```
public class X{
    private int x;
    private int y;
    private int z;
    X(){
        x=0;
        y=0;
        z=0;
    }
    public getSum(){
        return x+y;
    }
    public setMul(){
        return x*y;
    }

    public getZ(){
        return z;
    }
}
```

Exercise 2

false positive from apache common collection

/home/user/Documents/outilsVV/extp1/commons-collections/src/main/java/org/apache/commons/collections4/bidimap/TreeBidiMap.java:1098: CompareObjectsWithEquals: Use equals() to compare object references.

```
/**
 * Compares for equals as per the API.
 *
 * @param obj the object to compare to
 * @param dataElement either the {@link DataElement#KEY key}
 *                    or the {@link DataElement#VALUE value}.
 * @return true if equal
 */
private boolean doEquals(final Object obj, final DataElement dataElement) {
1098     if (obj == this) {
        return true;
    }
    if (!(obj instanceof Map)) {
        return false;
    }
    final Map<?, ?> other = (Map<?, ?>) obj;
    if (other.size() != size()) {
        return false;
    }

    if (nodeCount > 0) {
        try {
```

```

        for (final MapIterator<?, ?> it = getMapIterator(dataElement);
it.hasNext(); ) {
            final Object key = it.next();
            final Object value = it.getValue();
            if (!value.equals(other.get(key))) {
                return false;
            }
        }
    } catch (final ClassCastException | NullPointerException ex) {
        return false;
    }
}
return true;
}

```

here we do a comparator, a comparator equals exist but is using the here code doEquals. As such using the equals is not possible. Furthermore the `obj == this` is not an error because if the object is actually the current “this” object, we must return that they are equals. For this, the line `obj == this` compare the reference of the 2 objects which in this case is the same to return true. As such this error is a false positive.

another one

from apache common collection

/home/user/Documents/outilsVV/extp1/commons-collections/src/main/java/org/apache/commons/collections4/MultiMapUtils.java:94: ReturnEmptyCollectionRatherThanNull: Return an empty collection rather than null.

true positive

On another old project of my own, I found that :

- src/main/java/risk/checks/TeamChecks.java:14: UnusedFormalParameter: Avoid unused constructor parameters such as ‘idServer’.
- src/main/java/risk/checks/TeamChecks.java:14: UnusedFormalParameter: Avoid unused constructor parameters such as ‘idPlayer’.

the code of the constructor is the following :

```

public TeamChecks(int idServer,int idPlayer) {
    super();
}

```

The constructor here is completely useless, and must be removed or updated.

Exercise 3

```

<rule name="ComplexIfStatements"
  language="java"
  message="Too many if, consider refactoring this code"
  class="net.sourceforge.pmd.lang.rule.xpath.XPathRule">
  <description>
    Triggered when 3 following if are detected
  </description>
  <priority>4</priority>
  <properties>
    <property name="xpath">
      <value>

```

```
<![CDATA[  
  //IfStatement//IfStatement//IfStatement  
]]>  
</value>  
</property>  
</properties>  
</rule>
```

Exercice 4

see the code file.

Exercice 5

see the code file.