Mutation Testing

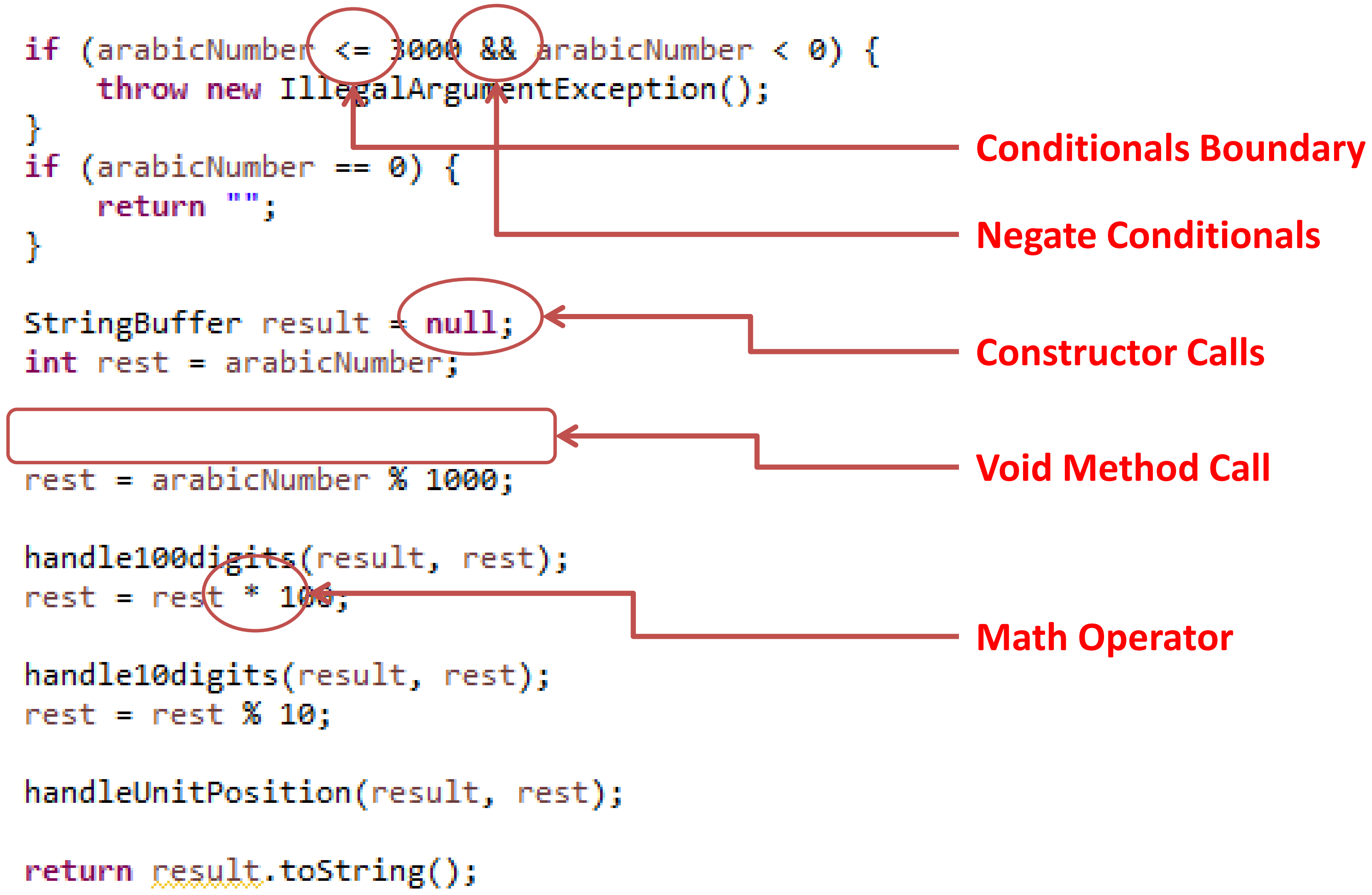
Why is 100% not enough?

Trainer: Michael Albrecht

„Ich bin Brian"
„Nein, ich bin Brian"
„Nein, ich bin Brian und meine Frau ist auch Brian"

Mutanten

Code mutants



```
if (arabicNumber <= 3000 && arabicNumber < 0) {
    throw new IllegalArgumentException();
}
if (arabicNumber == 0) {
    return "";
}

StringBuffer result = null;
int rest = arabicNumber;


rest = arabicNumber % 1000;

handle100digits(result, rest);
rest = rest * 100;

handle10digits(result, rest);
rest = rest % 10;

handleUnitPosition(result, rest);

return result.toString();
```

Outcomes

Mutant → **red Tests** → **Killed**

Mutant → **green Tests** → **Survived**

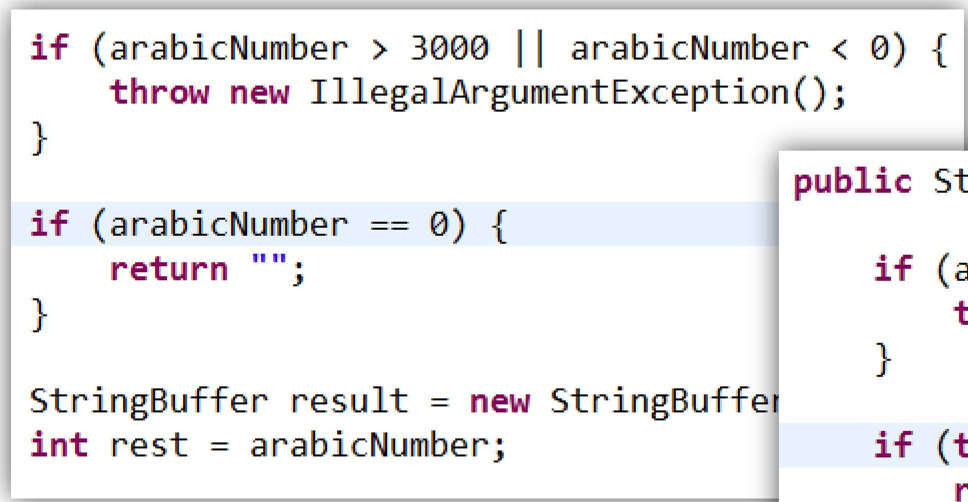
Mutant → **ungültig** → **Non viable**

Timeout

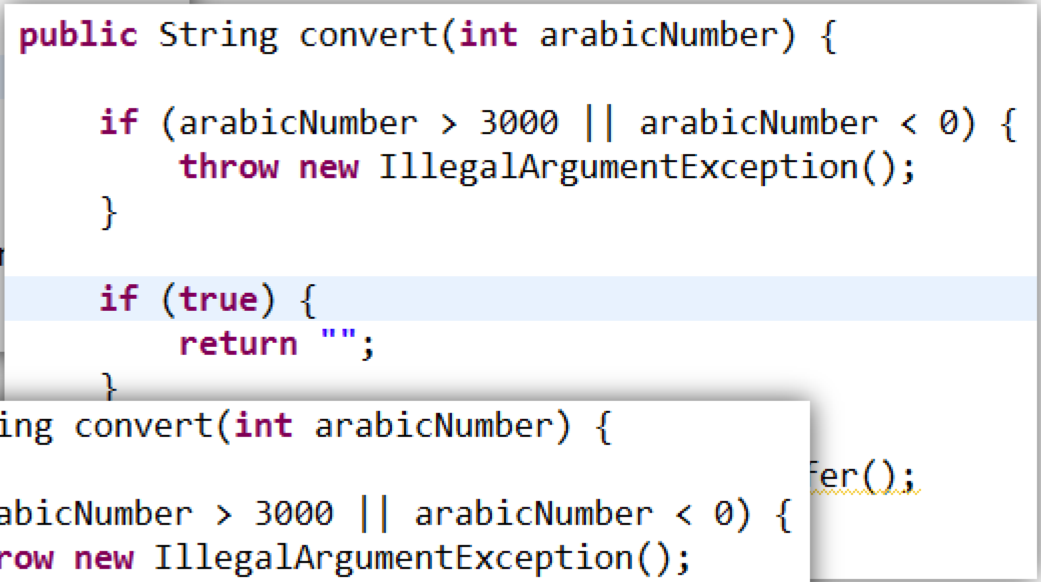
Memory error

Run error

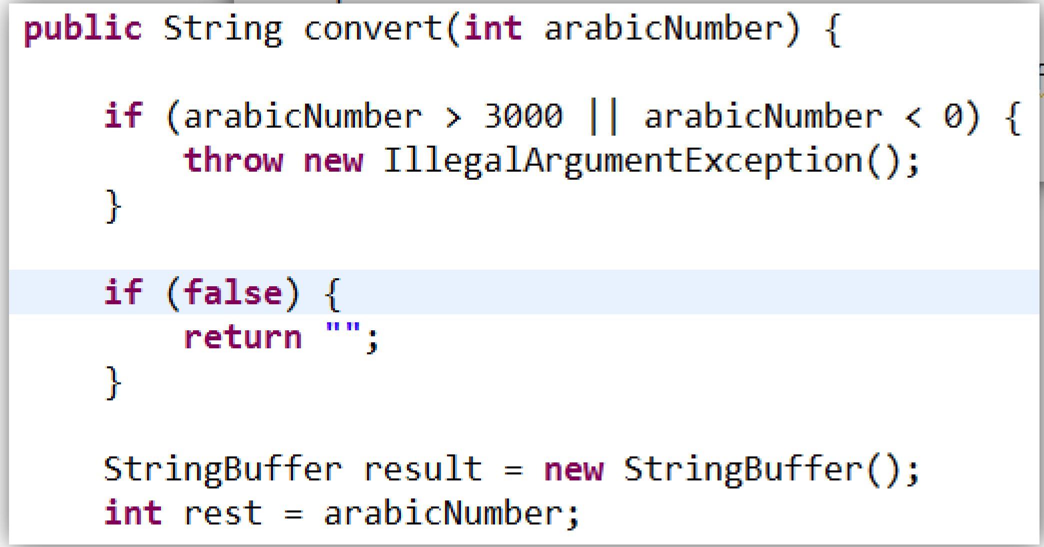
Remove Conditionals



```java
if (arabicNumber > 3000 || arabicNumber < 0) {
    throw new IllegalArgumentException();
}

if (arabicNumber == 0) {
    return "";
}

StringBuffer result = new StringBuffer
int rest = arabicNumber;
```



```java
public String convert(int arabicNumber) {

    if (arabicNumber > 3000 || arabicNumber < 0) {
        throw new IllegalArgumentException();
    }

    if (true) {
        return "";
    }

fer();
```



```java
public String convert(int arabicNumber) {

    if (arabicNumber > 3000 || arabicNumber < 0) {
        throw new IllegalArgumentException();
    }

    if (false) {
        return "";
    }

    StringBuffer result = new StringBuffer();
    int rest = arabicNumber;
```

Execution

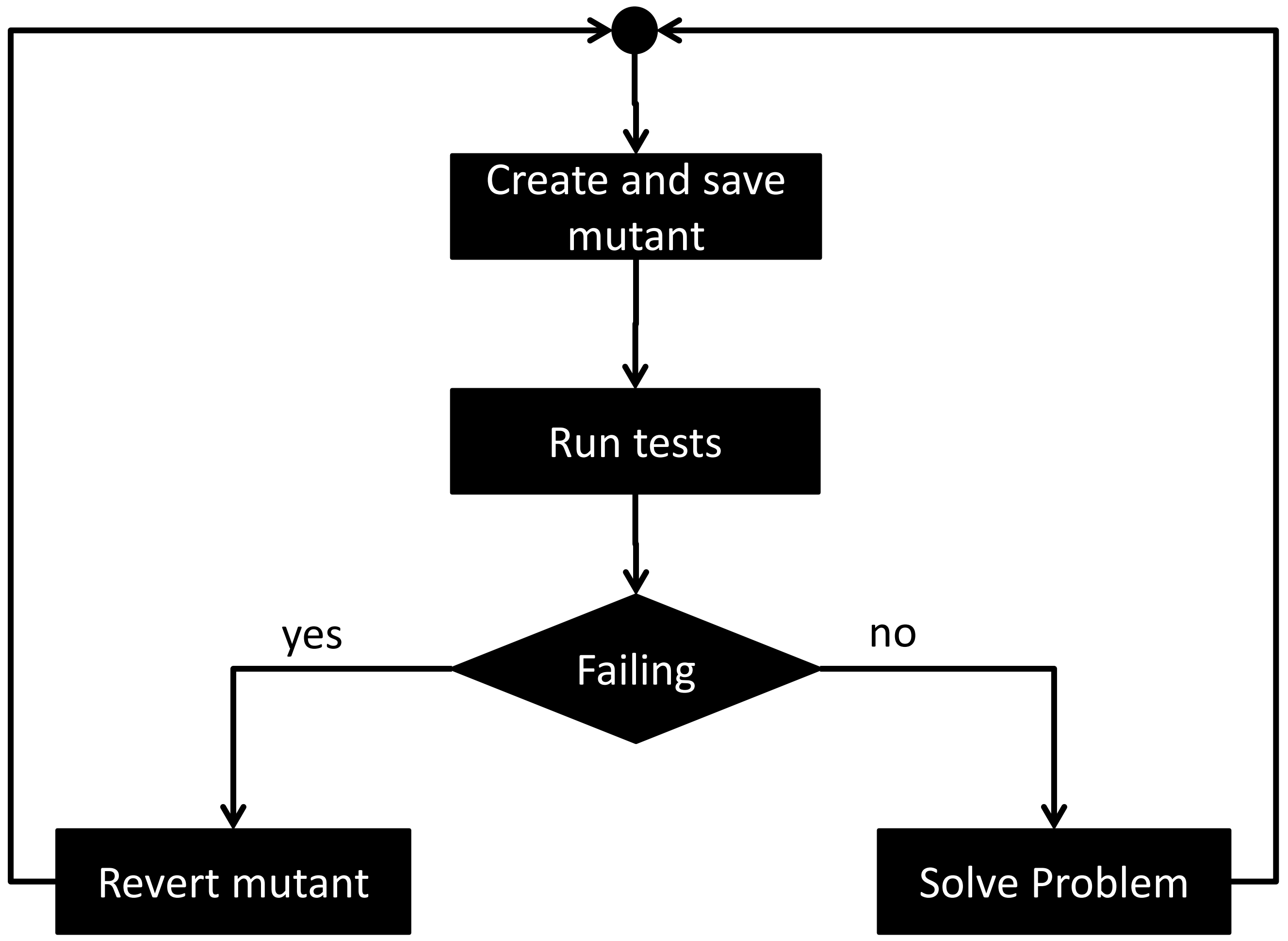
Manual

- Review phase
- More complex mutants
- High flexibility

Automatic / tool based

- Integration in CI-Server
- Integration in Sonar
- Coverage calculation

Manual sequence



Tools and compatibility matrix

Name	Last release	Maven	Sonar	cmd	Eclipse
Pitest	**11/2014**	✓	✓	✓	✓
Jumble	04/2013	✗	✗	✓	✗
Jester	11/2009	✗	✗	✓	✗
Judy	01/2014	✗	✗	✓	✗
Mutant Testing	**11/2014**	✗	✗	✓	✗

Name	Mockito	PowerMock
Pitest	✓	✓
Jumble	✗	✗
Jester	✓	✓
Judy	✗	✗
Mutant Testing	✗	✗

Features : Pitest.org

- Executable as

– Maven Plugin	mvn org.pitest:pitest-maven:mutationCoverage
– Ant Target	ant pit
– Command line	java –cp …MutationCoverageReport

- Selective coverage of classes and tests
- Selective mutators (ALL, DEFAULT,…)
- Output format (HTML, XML, CSV)
- Scalable
- Definable Thresholds

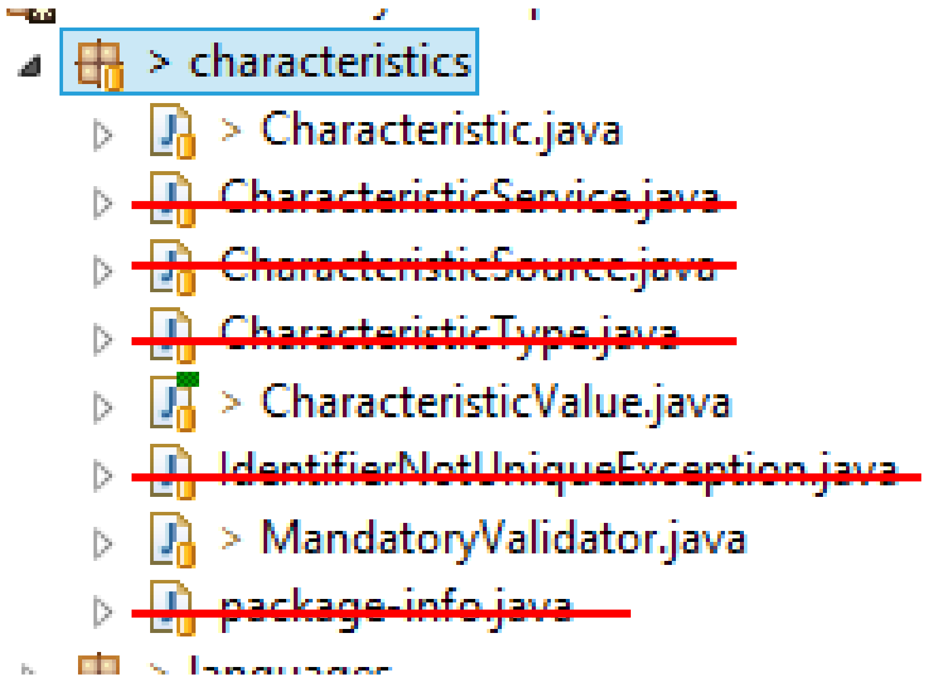
Additional Features

- Incremental analysis
- Extensions
  - Current engine: gregor
  - MutantFilter
  - Output Formatter
- Plugins for
  - Eclipse: pitclipse
  - Sonar: PIT sonar plugin

pitest.org

Pitest : Scope

Initial situation



> characteristics

> Characteristic.java

~~CharacteristicService.java~~

~~CharacteristicSource.java~~

~~CharacteristicType.java~~

> CharacteristicValue.java

~~IdentifierNotUniqueException.java~~

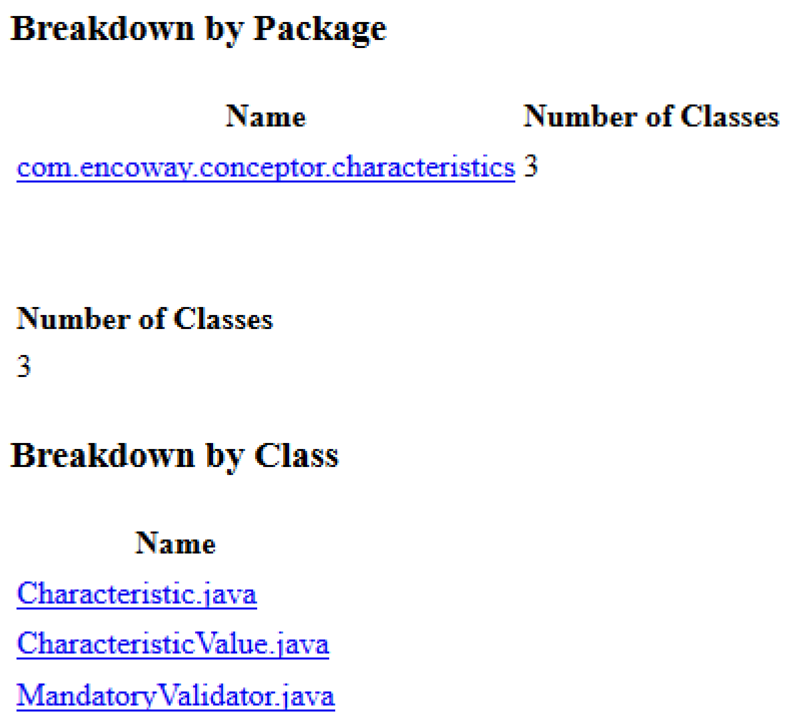
> MandatoryValidator.java

~~package-info.java~~

Package contains 7 artifacts of types:

- Javadoc
- Enum
- Interfaces
- Classes

Breakdown result



Breakdown by Package

Name	Number of Classes
com.encoway.conceptor.characteristics	3

Number of Classes

3

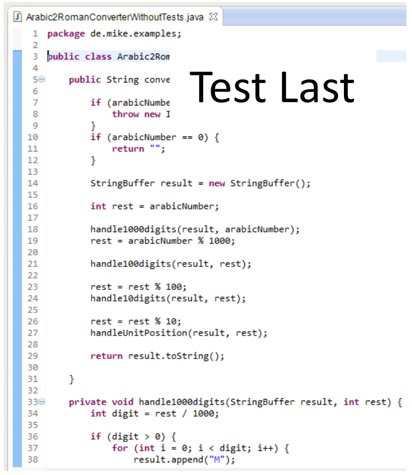
Breakdown by Class

Name
Characteristic.java
CharacteristicValue.java
MandatoryValidator.java

Log Output

```
[INFO]
[INFO] --- pitest-maven:1.1.1:mutationCoverage
...
09:02:28 PIT >> INFO : Sending 10 test classes to slave
09:02:28 PIT >> INFO : Sent tests to slave
09:02:28 PIT >> INFO : SLAVE : 09:02:28 PIT
             >> INFO : Found  31 tests
09:02:28 PIT >> INFO : Dependency analysis reduced number
                       of potential tests by 0
09:02:28 PIT >> INFO : 31 tests received

09:02:29 PIT >> INFO : Created  18 mutation test units
...
```

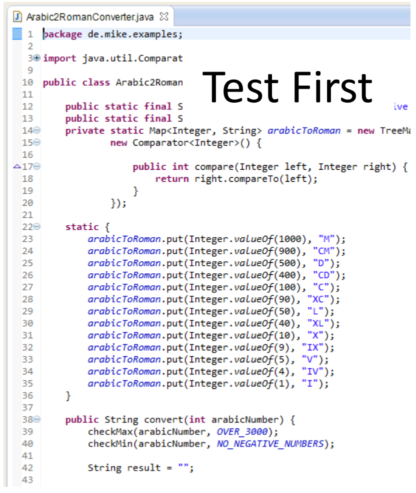
Szenario: Arabic2Roman Converter



Test Last

Arabic2RomanConverterWithoutTests.java

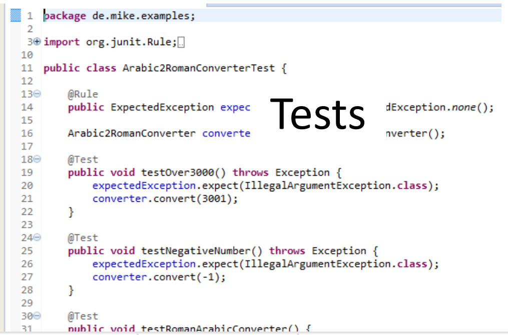
```java
package de.mike.examples;

public class Arabic2Rom

    public String conve

        if (arabicNumbe
            throw new I
        }
        if (arabicNumber == 0) {
            return "";
        }

        StringBuffer result = new StringBuffer();

        int rest = arabicNumber;

        handle1000digits(result, arabicNumber);
        rest = arabicNumber % 1000;

        handle100digits(result, rest);

        rest = rest % 100;
        handle10digits(result, rest);

        rest = rest % 10;
        handleUnitPosition(result, rest);

        return result.toString();

    }

    private void handle1000digits(StringBuffer result, int rest) {
        int digit = rest / 1000;

        if (digit > 0) {
            for (int i = 0; i < digit; i++) {
                result.append("M");
```



Test First

Arabic2RomanConverter.java

```java
package de.mike.examples;

import java.util.Comparat

public class Arabic2Roman

    public static final S                                          ive
    public static final S
    private static Map<Integer, String> arabicToRoman = new TreeMa
            new Comparator<Integer>() {

                public int compare(Integer left, Integer right) {
                    return right.compareTo(left);
                }
            });

    static {
        arabicToRoman.put(Integer.valueOf(1000), "M");
        arabicToRoman.put(Integer.valueOf(900), "CM");
        arabicToRoman.put(Integer.valueOf(500), "D");
        arabicToRoman.put(Integer.valueOf(400), "CD");
        arabicToRoman.put(Integer.valueOf(100), "C");
        arabicToRoman.put(Integer.valueOf(90), "XC");
        arabicToRoman.put(Integer.valueOf(50), "L");
        arabicToRoman.put(Integer.valueOf(40), "XL");
        arabicToRoman.put(Integer.valueOf(10), "X");
        arabicToRoman.put(Integer.valueOf(9), "IX");
        arabicToRoman.put(Integer.valueOf(5), "V");
        arabicToRoman.put(Integer.valueOf(4), "IV");
        arabicToRoman.put(Integer.valueOf(1), "I");
    }

    public String convert(int arabicNumber) {
        checkMax(arabicNumber, OVER_3000);
        checkMin(arabicNumber, NO_NEGATIVE_NUMBERS);

        String result = "";

```

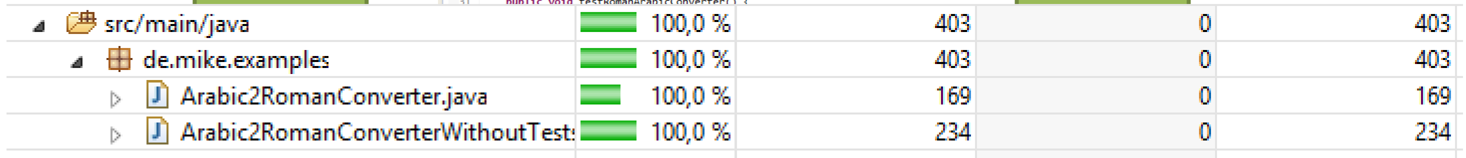


Tests

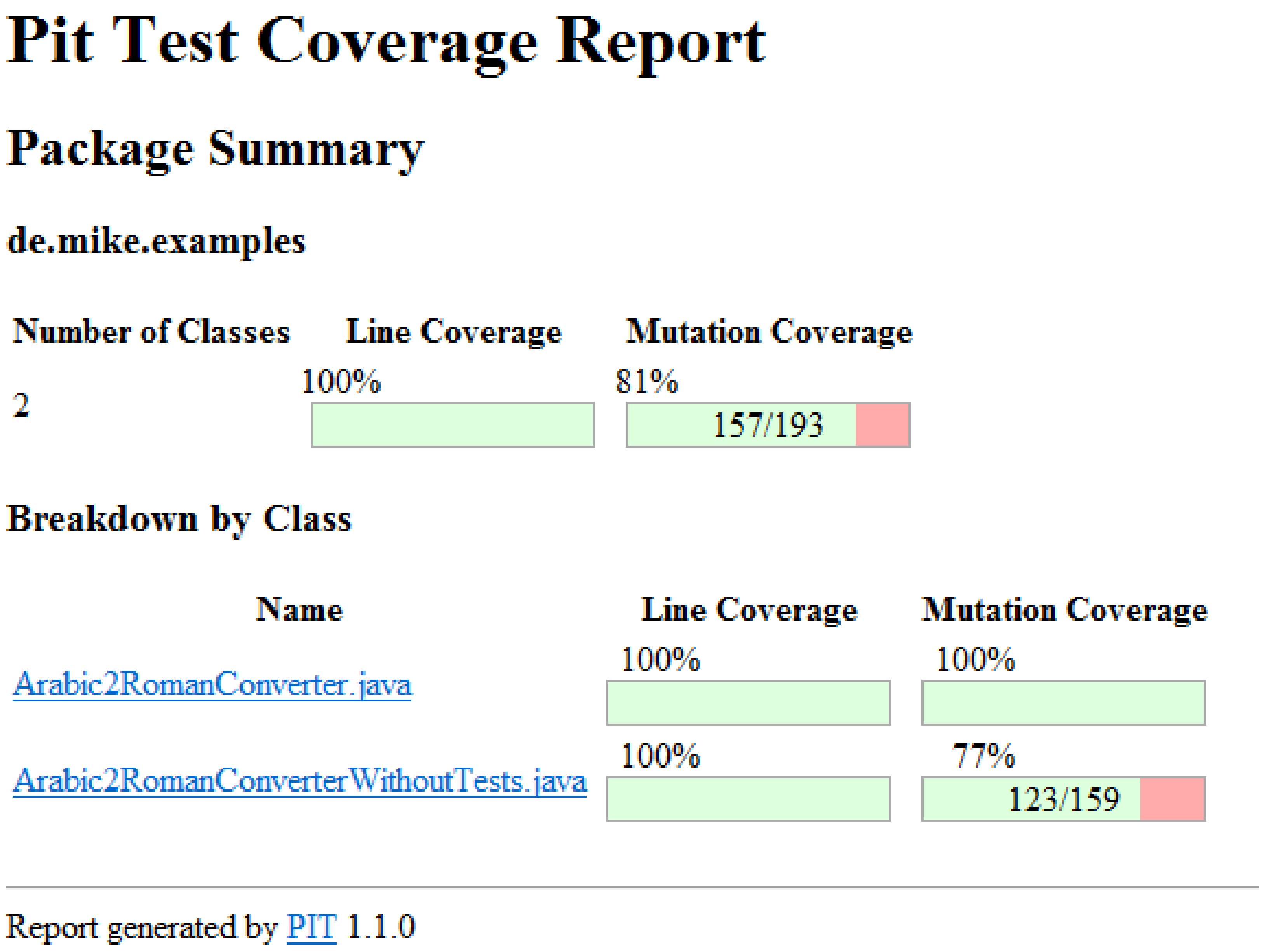
```java
package de.mike.examples;

import org.junit.Rule;

public class Arabic2RomanConverterTest {

    @Rule
    public ExpectedException expec                    dException.none();

    Arabic2RomanConverter converte                    nverter();

    @Test
    public void testOver3000() throws Exception {
        expectedException.expect(IllegalArgumentException.class);
        converter.convert(3001);
    }

    @Test
    public void testNegativeNumber() throws Exception {
        expectedException.expect(IllegalArgumentException.class);
        converter.convert(-1);
    }

    @Test
    public void testRomanArabicConverter() {
```

100% Line Coverage

100% Line Coverage



◢ src/main/java	🟩 100,0 %	403	0	403
◢ de.mike.examples	🟩 100,0 %	403	0	403
▷ Arabic2RomanConverter.java	🟩 100,0 %	169	0	169
▷ Arabic2RomanConverterWithoutTest:	🟩 100,0 %	234	0	234



Pit Test Coverage Report

Package Summary

de.mike.examples

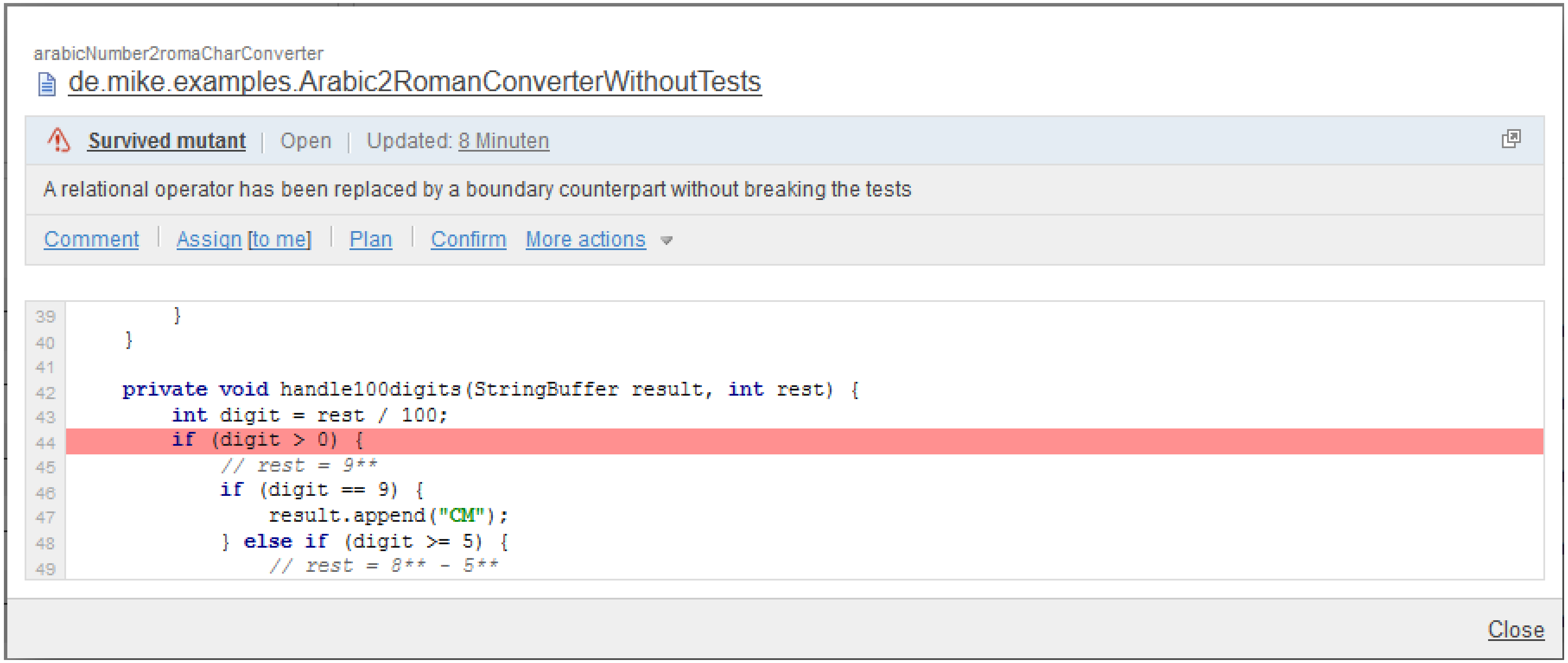
Number of Classes	Line Coverage	Mutation Coverage
2	100%	81% 157/193

Breakdown by Class

Name	Line Coverage	Mutation Coverage
Arabic2RomanConverter.java	100%	100%
Arabic2RomanConverterWithoutTests.java	100%	77% 123/159

Report generated by PIT 1.1.0

PIT Sonar Plugin


arabicNumber2romaCharConverter

de.mike.examples.Arabic2RomanConverterWithoutTests

Survived mutant | Open | Updated: 8 Minuten

A relational operator has been replaced by a boundary counterpart without breaking the tests

```
            }
        }

    private void handle100digits(StringBuffer result, int rest) {
        int digit = rest / 100;
        if (digit > 0) {
            // rest = 9**
            if (digit == 9) {
                result.append("CM");
            } else if (digit >= 5) {
                // rest = 8** - 5**
```

```
mvn org.pitest:pitest-maven:mutationCoverage

mvn sonar:sonar -Dsonar.pitest.mode=reuseReport
```