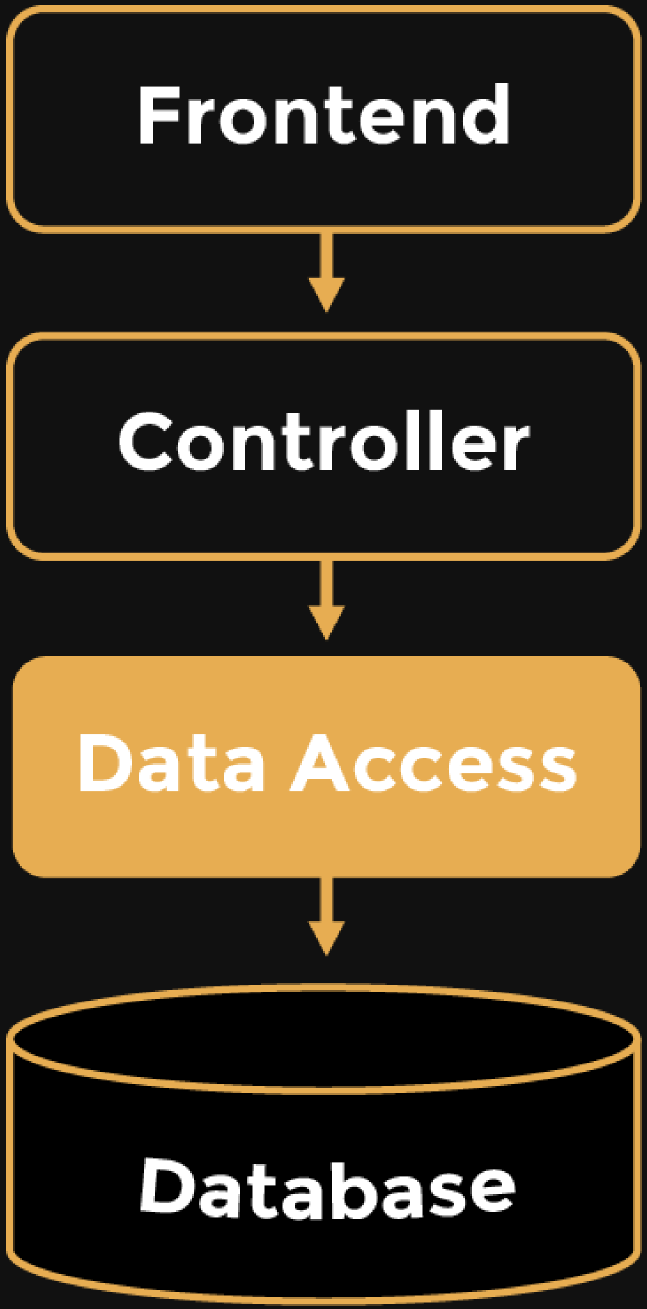
Und wer testet die Tests?

Mutationstesten mit PIT

Johannes Dienst

Warum Mutationstesten?

Legacy Projekt


Frontend
Controller
Data Access
Database

100% aussagekräftige Tests

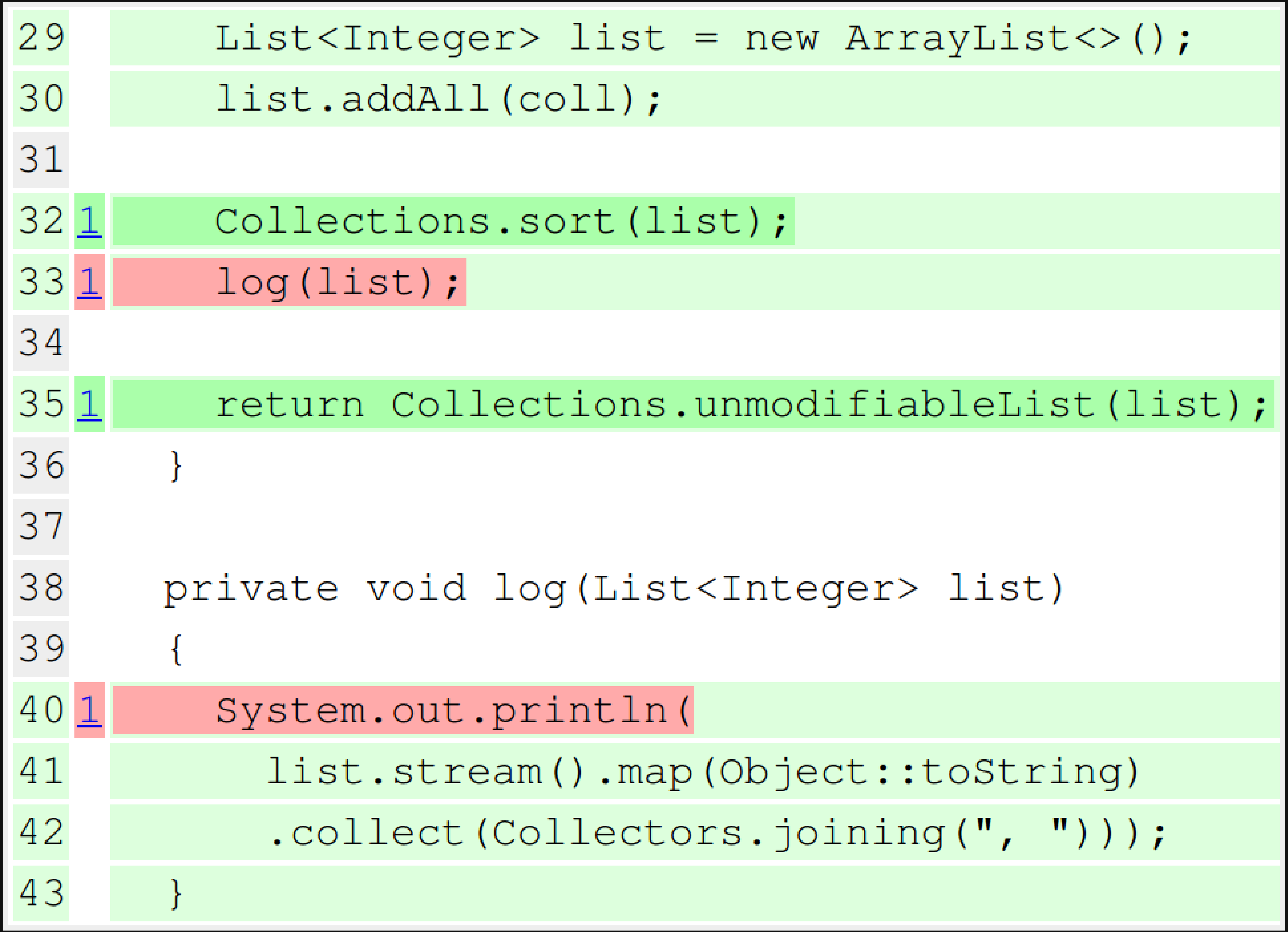
100% Zeilenabdeckung

Wir sind fertig!

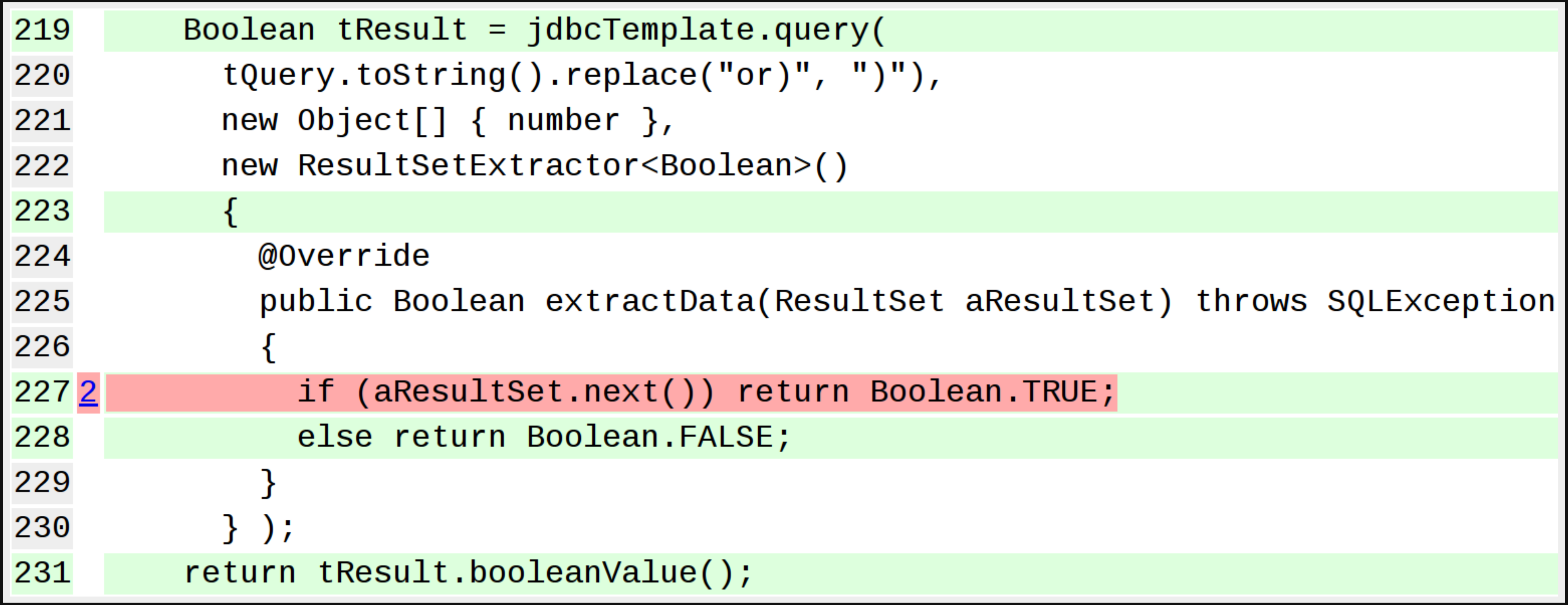
So einfach?

100% ≠ Fehlerfreiheit

⚠ Subtile Bugs ⚠



```
        List<Integer> list = new ArrayList<>();
        list.addAll(coll);

        Collections.sort(list);
        log(list);

        return Collections.unmodifiableList(list);
    }

    private void log(List<Integer> list)
    {
        System.out.println(
            list.stream().map(Object::toString)
            .collect(Collectors.joining(", ")));
    }
```



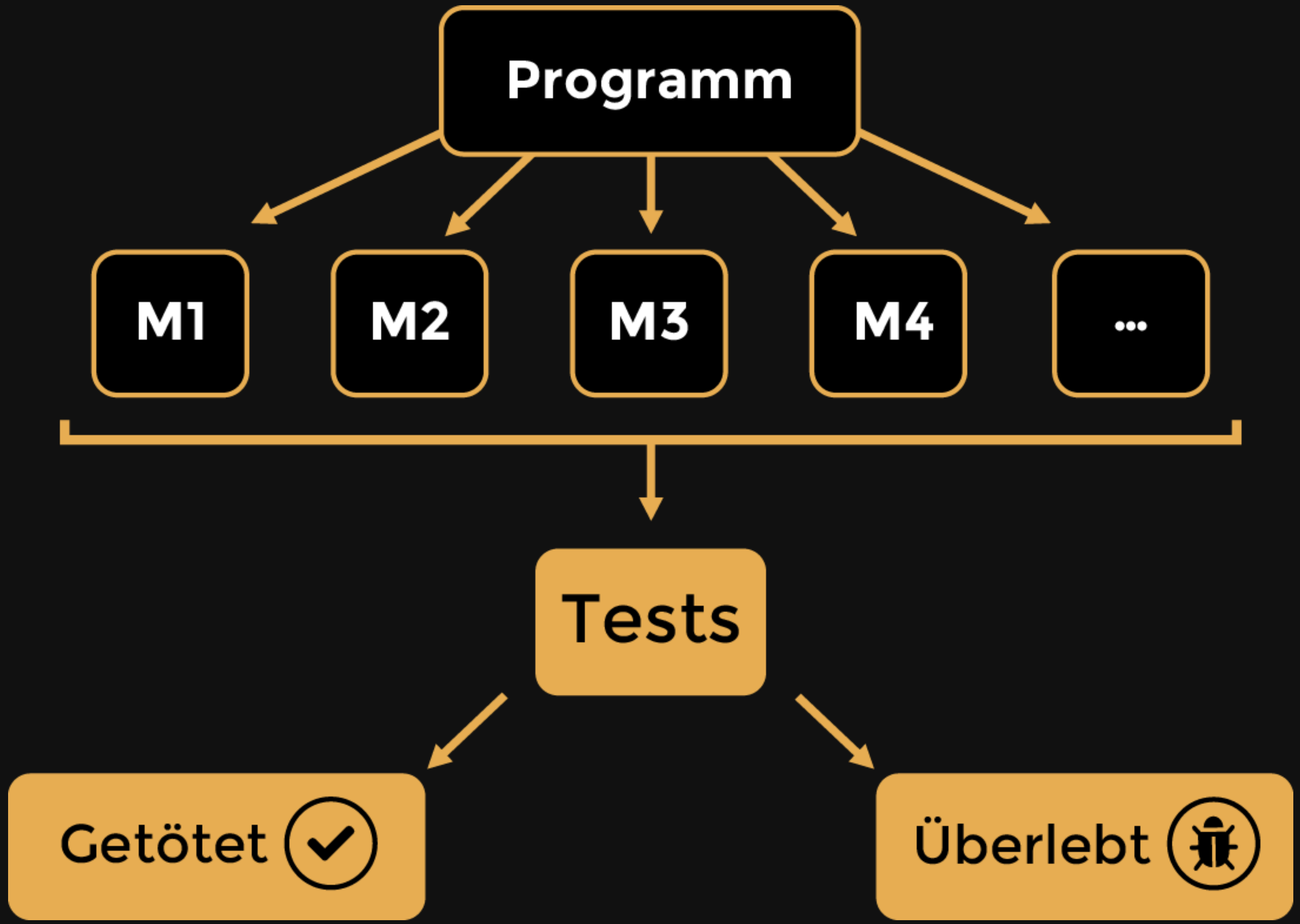
```
    Boolean tResult = jdbcTemplate.query(
      tQuery.toString().replace("or)", ")"),
      new Object[] { number },
      new ResultSetExtractor<Boolean>()
      {
        @Override
        public Boolean extractData(ResultSet aResultSet) throws SQLException
        {
          if (aResultSet.next()) return Boolean.TRUE;
          else return Boolean.FALSE;
        }
      } );
    return tResult.booleanValue();
```

Tests härten!

Wie?

Mutationstesten

Richard Lipton 1971


Programm
M1
M2
M3
M4
...
Tests
Getötet
Überlebt

Gold Standard

⇨ aussagekräftige Tests

PIT

pitest.org

7 Default Mutatoren

12 Experimentelle Mutatoren

Default

Bedingungen

```
if (i == 0) {
  return 0;
}
```

```
if (i != 0) {
  return 0;
}
```

Mathematisch

```
int a = b + c;
```

```
int a = b - c;
```

Rückgabetyp

Entfernung von void-Methodenaufrufen

Schnell

ant, maven, gradle etc.

Menschenlesbare Reports

Testsuite

171 Tests

Laufzeit

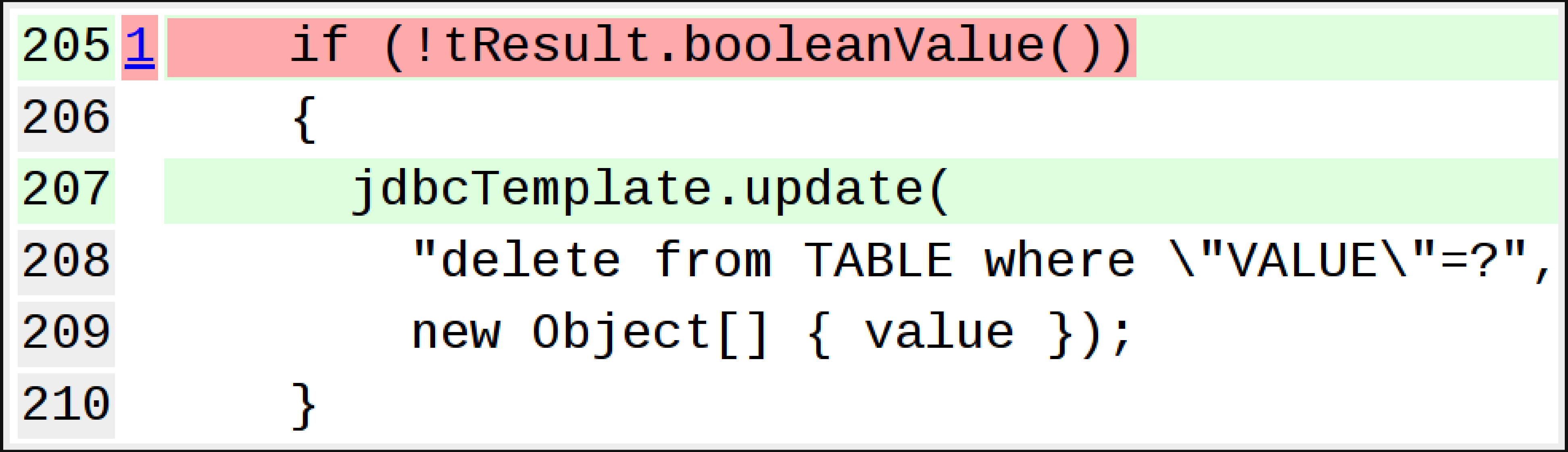
Ohne PIT: 0.5 min

Mit PIT: 4 min

Erkenntnisse

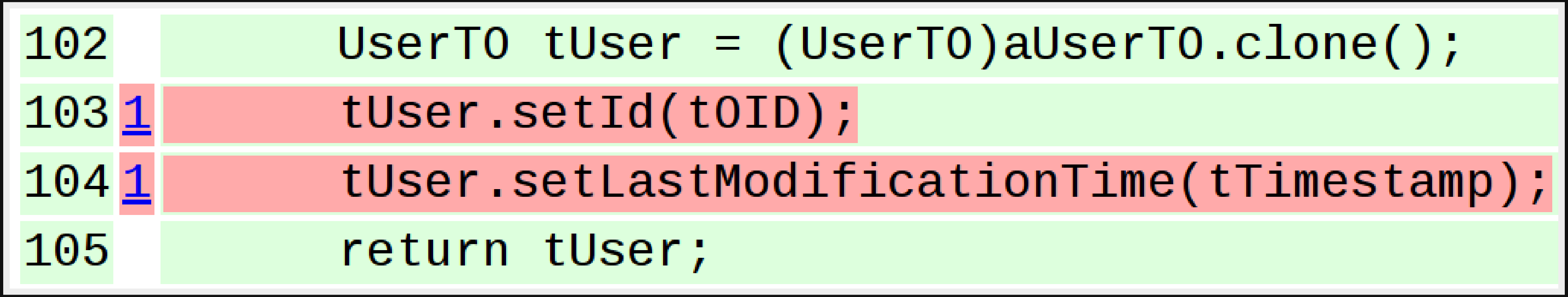
Bugs

Seiteneffekte



```
1       if (!tResult.booleanValue())
        {
          jdbcTemplate.update(
            "delete from TABLE where \"VALUE\"=?",
            new Object[] { value });
        }
```

False positives



```
        UserTO tUser = (UserTO)aUserTO.clone();
1       tUser.setId(tOID);
1       tUser.setLastModificationTime(tTimestamp);
        return tUser;
```

Demo

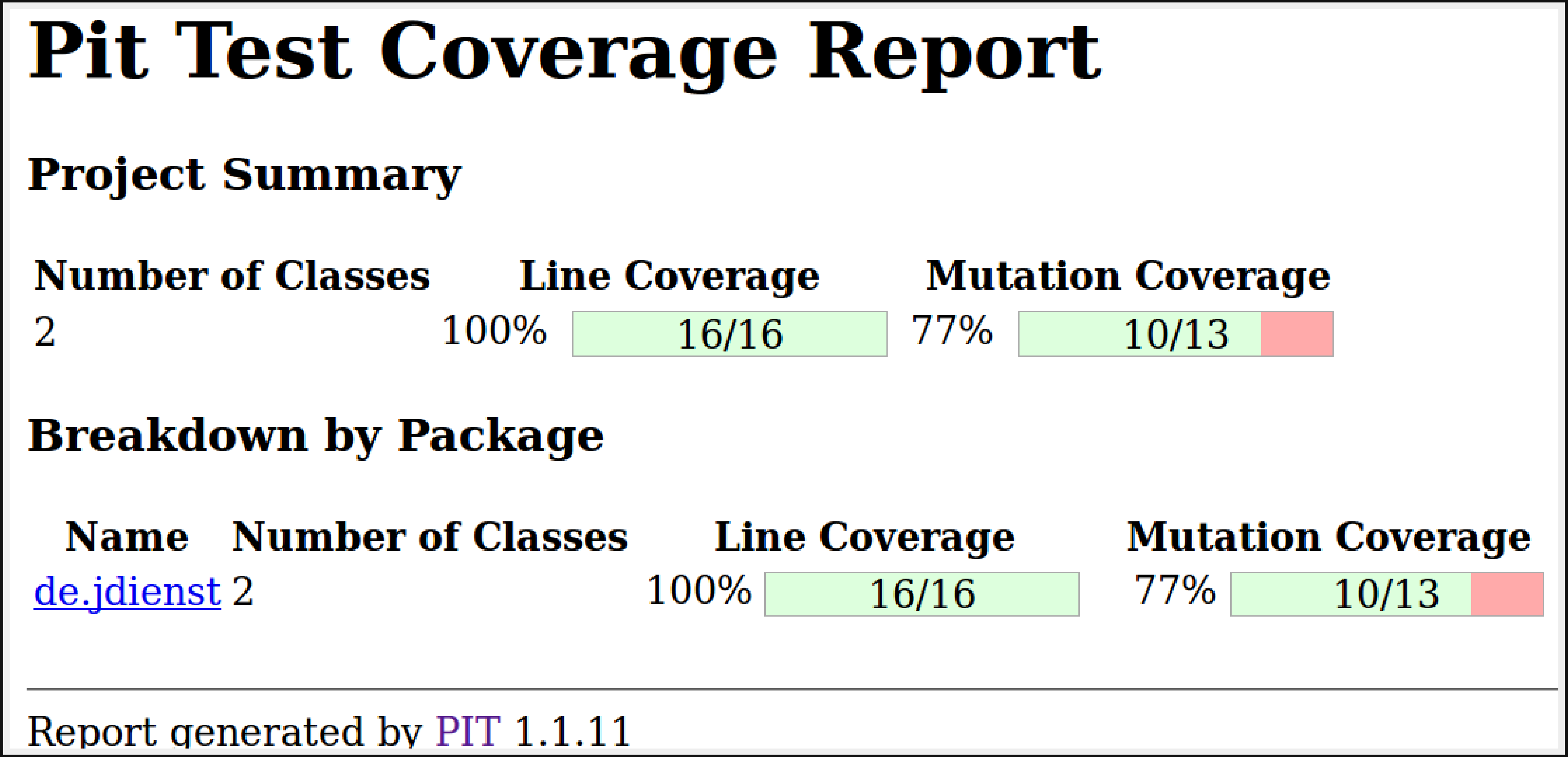
```xml
<plugin>
  <groupId>org.pitest</groupId>
  <artifactId>pitest-maven</artifactId>
  <version>1.1.11</version>
  <configuration>
    <targetClasses>
      <param>de.*</param>
    </targetClasses>
    <targetTests>
      <param>de.*</param>
    </targetTests>
    <threads>3</threads>
  </configuration>
</plugin>
```

```java
public class Fibonacci {

  public int calc(int i) {
    if (i == 0) {
      return 0;
    }

    if (i <= 2) {
      return 1;
    }

    return calc(i-1) + calc(i-2);
  }
}
```

```java
@Test public void seedValue0() {
  assertEquals( 0, fib.calc( 0));
}

@Test public void seedValue1() {
  assertEquals( 1, fib.calc( 1));
}

@Test public void seedValue2() {
  assertEquals( 1, fib.calc( 2));
}

@Test public void value3() {
  assertEquals( 2, fib.calc( 3));
}

@Test public void value11() {
  assertEquals( 89, fib.calc( 11));
}
```

```java
public class Sort {

  public static List sort(List coll) {
    List list = new ArrayList<>();
    list.addAll(coll);

    Collections.sort( list);
    log( list);

    return Collections.unmodifiableList( list);
  }

  private static void log( List list) {
    System.out.println(
      list.stream().map(Object::toString)
      .collect(Collectors.joining( ", ")));
  }
}
```

```
@Test public void emptyList() {
  assertEquals( true, Sort.sort(Collections.<Integer>emptyList()).isEmpty());
}

@Test public void oneList() {
  assertEquals( false,
    Sort.sort(Stream.of( 42).collect(Collectors.toList())).isEmpty());
}

@Test public void twoList() {
  assertEquals( new Integer(1),
    Sort.sort(Stream.of( 2, 3, 1, 8).collect(Collectors.toList())).get( 0));
}
```



Pit Test Coverage Report

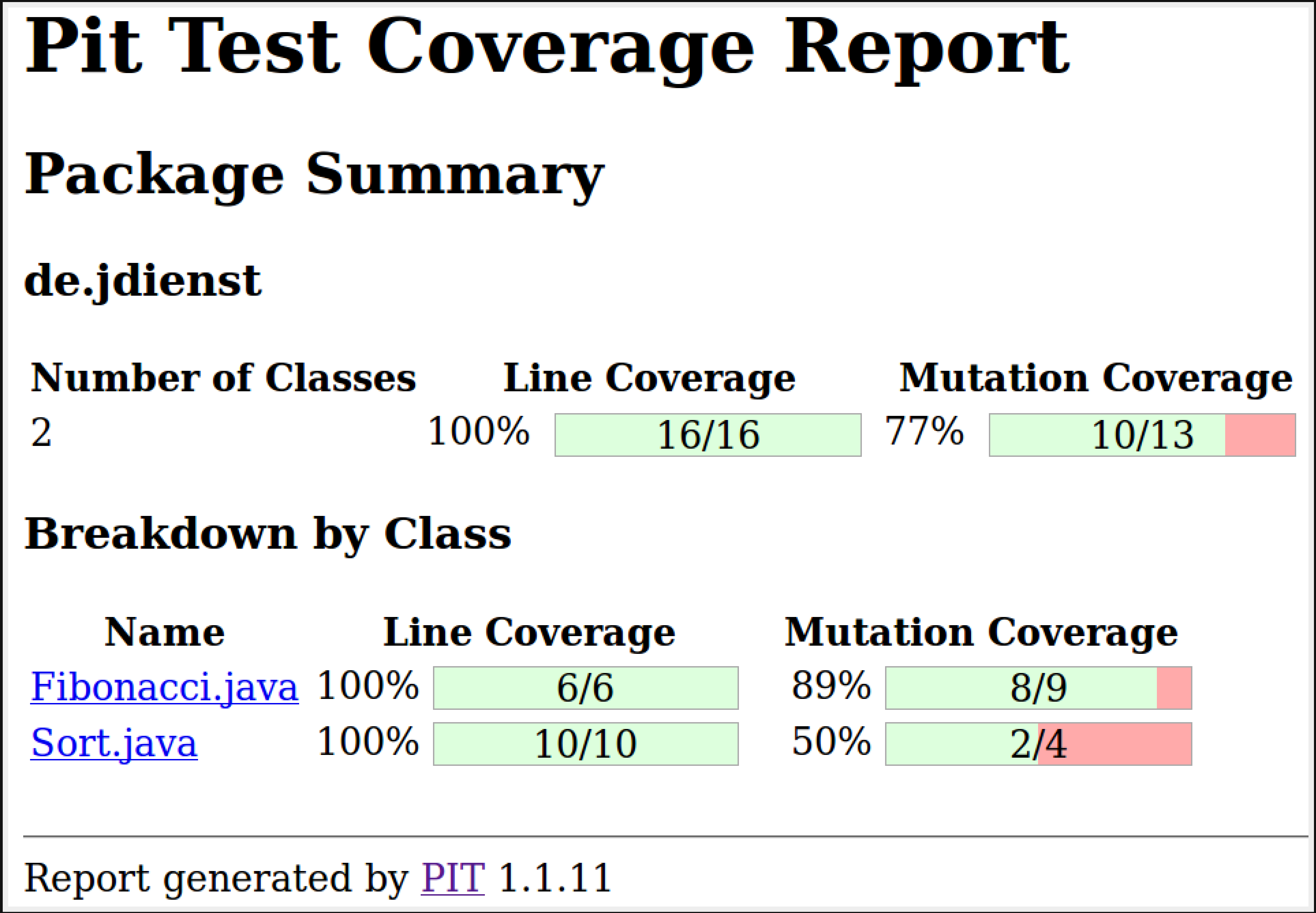
Project Summary

Number of Classes	Line Coverage		Mutation Coverage	
2	100%	16/16	77%	10/13

Breakdown by Package

Name	Number of Classes	Line Coverage		Mutation Coverage	
de.jdienst	2	100%	16/16	77%	10/13

Report generated by PIT 1.1.11



Pit Test Coverage Report

Package Summary

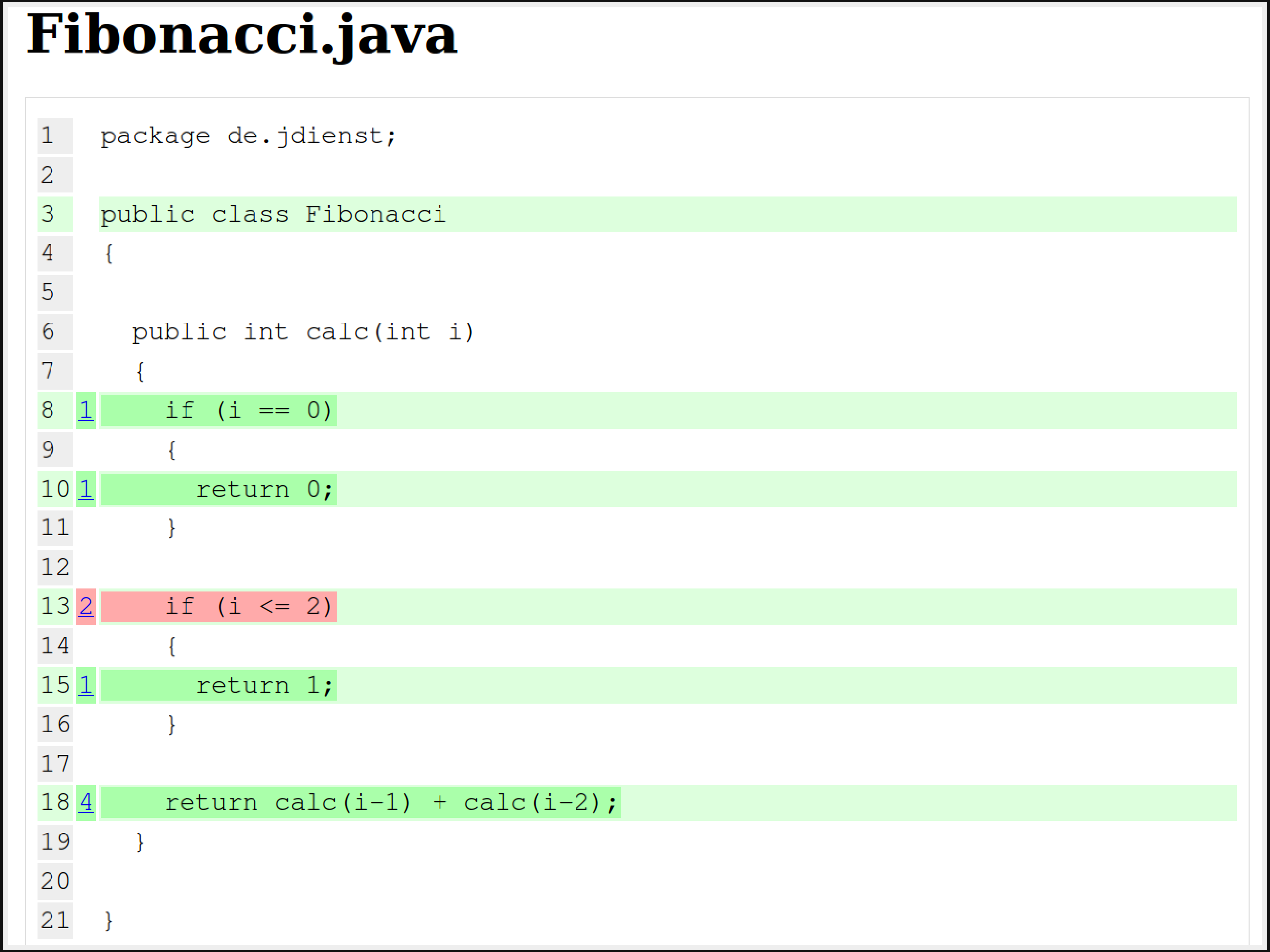
de.jdienst

Number of Classes	Line Coverage		Mutation Coverage	
2	100%	16/16	77%	10/13

Breakdown by Class

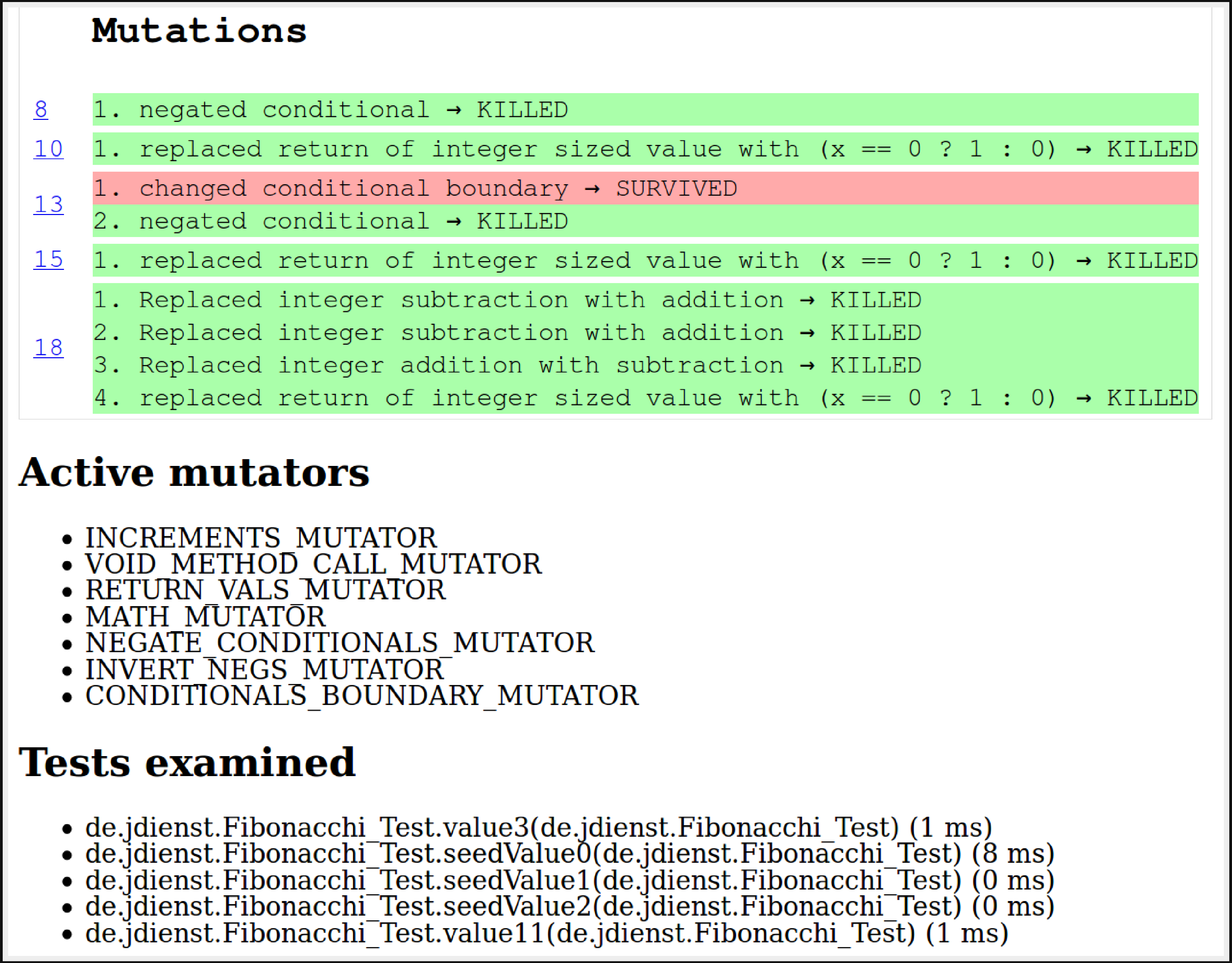
Name	Line Coverage		Mutation Coverage	
Fibonacci.java	100%	6/6	89%	8/9
Sort.java	100%	10/10	50%	2/4

Report generated by PIT 1.1.11



Fibonacci.java

```
   package de.jdienst;

   public class Fibonacci
   {

     public int calc(int i)
     {
1      if (i == 0)
       {
1        return 0;
       }

2      if (i <= 2)
       {
1        return 1;
       }

4      return calc(i-1) + calc(i-2);
     }

   }
```



Mutations

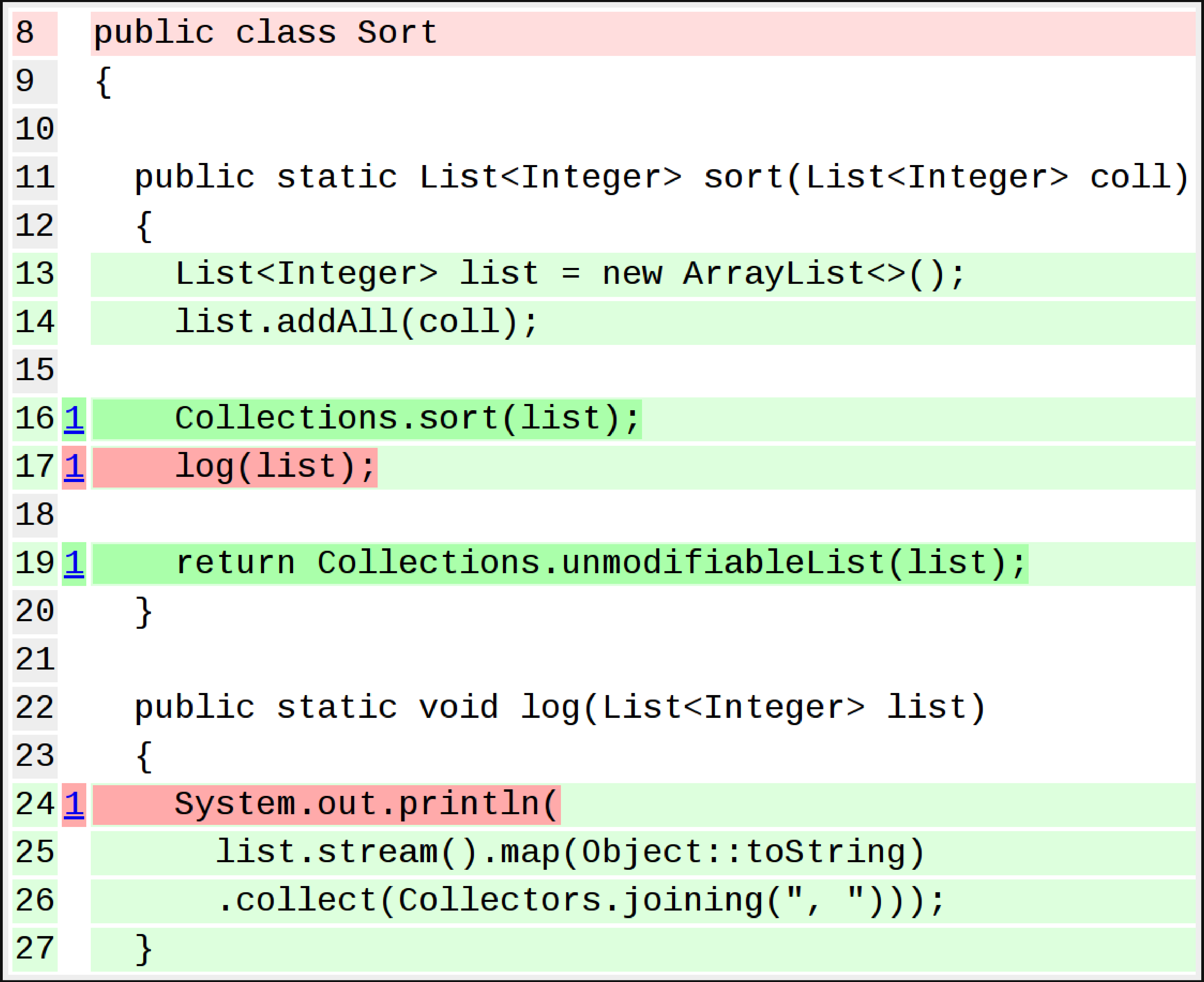
```
8   1. negated conditional → KILLED
10  1. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
13  1. changed conditional boundary → SURVIVED
    2. negated conditional → KILLED
15  1. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
18  1. Replaced integer subtraction with addition → KILLED
    2. Replaced integer subtraction with addition → KILLED
    3. Replaced integer addition with subtraction → KILLED
    4. replaced return of integer sized value with (x == 0 ? 1 : 0) → KILLED
```

Active mutators

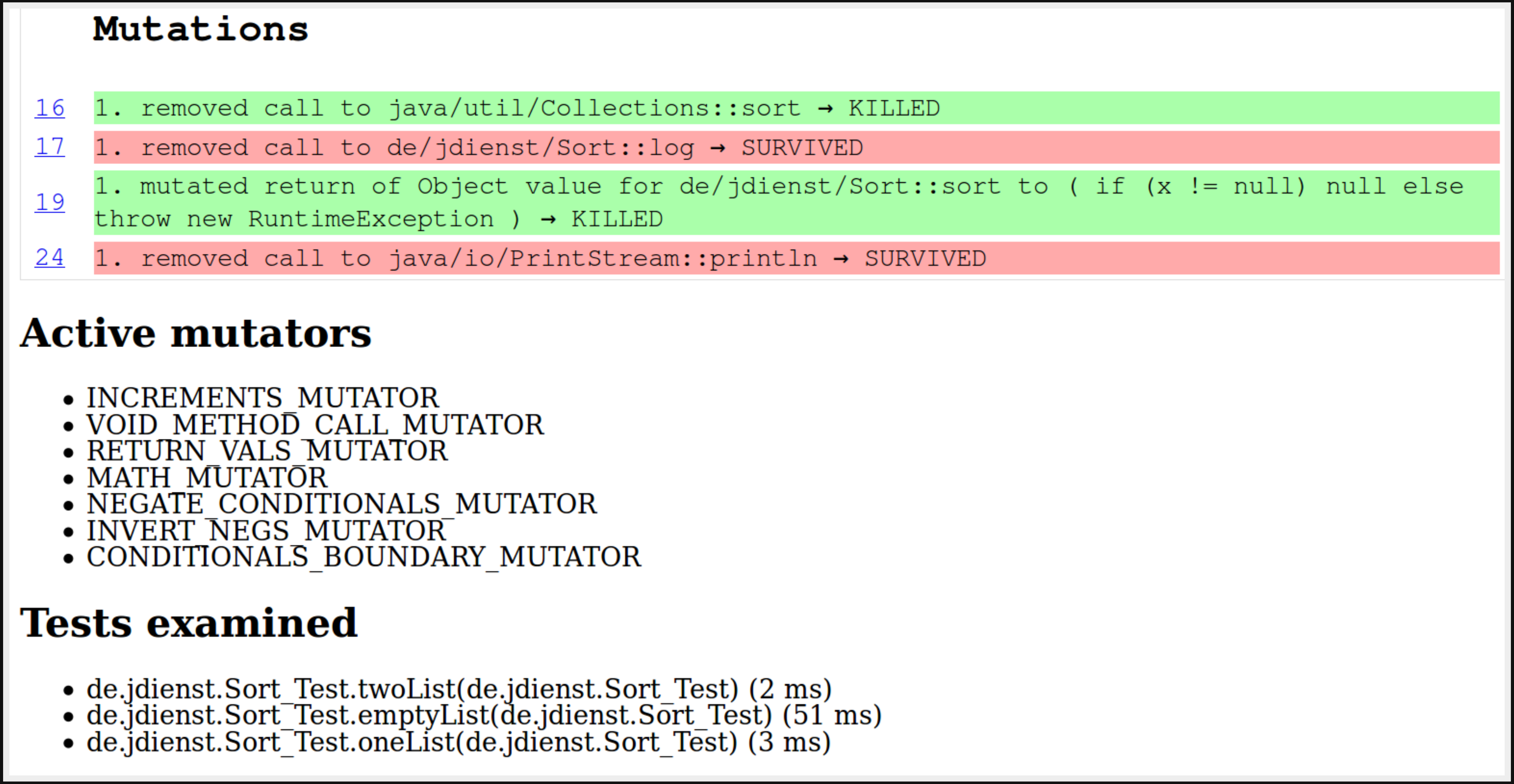
- INCREMENTS_MUTATOR
- VOID_METHOD_CALL_MUTATOR
- RETURN_VALS_MUTATOR
- MATH_MUTATOR
- NEGATE_CONDITIONALS_MUTATOR
- INVERT_NEGS_MUTATOR
- CONDITIONALS_BOUNDARY_MUTATOR

Tests examined

- de.jdienst.Fibonacci_Test.value3(de.jdienst.Fibonacci_Test) (1 ms)
- de.jdienst.Fibonacci_Test.seedValue0(de.jdienst.Fibonacci_Test) (8 ms)
- de.jdienst.Fibonacci_Test.seedValue1(de.jdienst.Fibonacci_Test) (0 ms)
- de.jdienst.Fibonacci_Test.seedValue2(de.jdienst.Fibonacci_Test) (0 ms)
- de.jdienst.Fibonacci_Test.value11(de.jdienst.Fibonacci_Test) (1 ms)



```java
public class Sort
{

  public static List<Integer> sort(List<Integer> coll)
  {
    List<Integer> list = new ArrayList<>();
    list.addAll(coll);

    Collections.sort(list);
    log(list);

    return Collections.unmodifiableList(list);
  }

  public static void log(List<Integer> list)
  {
    System.out.println(
      list.stream().map(Object::toString)
      .collect(Collectors.joining(", ")));
  }
```



Mutations

```
16  1. removed call to java/util/Collections::sort → KILLED
17  1. removed call to de/jdienst/Sort::log → SURVIVED
19  1. mutated return of Object value for de/jdienst/Sort::sort to ( if (x != null) null else
    throw new RuntimeException ) → KILLED
24  1. removed call to java/io/PrintStream::println → SURVIVED
```

Active mutators

- INCREMENTS_MUTATOR
- VOID_METHOD_CALL_MUTATOR
- RETURN_VALS_MUTATOR
- MATH_MUTATOR
- NEGATE_CONDITIONALS_MUTATOR
- INVERT_NEGS_MUTATOR
- CONDITIONALS_BOUNDARY_MUTATOR

Tests examined

- de.jdienst.Sort_Test.twoList(de.jdienst.Sort_Test) (2 ms)
- de.jdienst.Sort_Test.emptyList(de.jdienst.Sort_Test) (51 ms)
- de.jdienst.Sort_Test.oneList(de.jdienst.Sort_Test) (3 ms)

Fazit

100% ≠ Fehlerfreiheit

Arbeitsaufwand

Bessere Testsuite

JohannesDienst

johannesdienst.net

info@johannesdienst.net