

## 02161: Software Engineering 1

### 4.1 Systematic tests

Given the following code that tests whether a number  $a$  is prime or not. A number is prime, if it is greater than 1 and only dividiable by itself or 1.

```
public class Primes {  
    public static boolean isPrime(int n) {  
        if (n <= 1) { // 1  
            return false; // 1a  
        }  
        for (int i = 2; i < n; i++ ) { // 2  
            if (n % i != 0) { // 3  
                return false; // 3a  
            }  
        }  
        return true; // 4  
    }  
}
```

- a Devise a set of white-box partition tests for this method using two tables (similar to what we have done in the lecture). The first table describes the input partitions, and the second table describes the test cases.
  - Submit the two tables as a PDF file to [huba@dtu.dk](mailto:huba@dtu.dk).
- b Implement the class Primes and the JUnit tests corresponding to the table of the test cases from part *a* of this exercise.
  - What is the code coverage for the isPrime method reported by EcEmma?
  - Do the tests pass? If not, fix the code and create new partition tests and implement them in JUnit. Note that, since the partition tests are based on the code, you have to create new partition tests, when the code changes!
  - Submit the Eclipse project as Zip file to [huba@dtu.dk](mailto:huba@dtu.dk)