Automated testing in Java

Martin Pola

- Validate that a program does what is expected
- Run tests periodically – catch errors early
- Manual tests are humans following steps in documents
- Automatic tests are carried out through the use of a *test framework*

Practise of testing

- Validate that a program does what is expected
- Run tests periodically – catch errors early
- Manual tests are humans following steps in documents
- Automatic tests are carried out through the use of a *test framework*

- Testing doesn't find bugs
- Testing only makes correctness *probable*, it doesn't *prove* it

Unit tests
Test single features, typically a method call.

Integration tests
Test that different parts of a program can integrate with each other, e.g. a database or a network service.

Unit tests
Test single features, typically a method call.

Integration tests
Test that different parts of a program can integrate with each other, e.g. a database or a network service.

We will use *JUnit* as our test framework.

- A unit test should *assert* something.
- *Expected value* is compared with *actual value*

Concrete example

```
1  @Test
2  public void testSetAndGetName() {
3      Dog dog = new Dog("Fido", 4);
4      Assert.assertEquals("Fido", dog.getName());
5  }
```

Concrete example

```java
1  @Test
2  public void testSetAndGetName() {
3      Dog dog = new Dog("Fido", 4);
4      Assert.assertEquals("Fido", dog.getName());
5  }
```

Methods for verifying assertion are found in the `Assert` class.

- assertEquals(<u>expected</u>, <u>actual</u>)
- assertTrue(<u>actual</u>)
- assertFalse(<u>actual</u>)
- assertNotNull(<u>actual</u>)
- ...

- Many tests run quickly; thousands of tests in just a few seconds
- What if you need to change the tests?

How many tests should you create?

- Many tests run quickly; thousands of tests in just a few seconds
- What if you need to change the tests?

How many lines from our program do our tests touch?

How many tests should you create?

- Many tests run quickly; thousands of tests in just a few seconds
- What if you need to change the tests?

How many lines from our program do our tests touch?

Measure *code coverage* and aim to cover all lines of code.