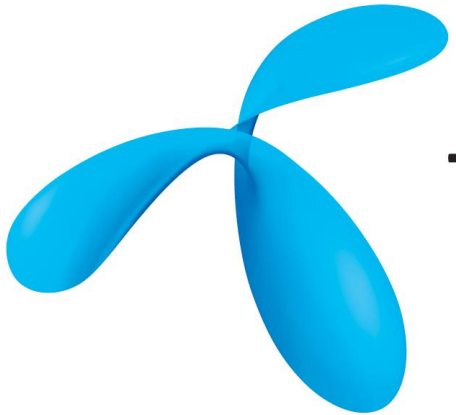
Abacourse Spring 2016

Testing with friends



telenor | digital

internship@telenordigital.com

Agenda

- Why test?
- Different types of testing
- How to write good tests
- Mocking
- Demo
- Test it out in practice

Why test?

- Make sure the code works when you write it (duh!)
- Make sure it continues to work when the next guy makes changes

Additional bonus:

- Works as a nice description of how the code is intended to work/not work
- Can be used as help to define API (TDD)

Different types of testing

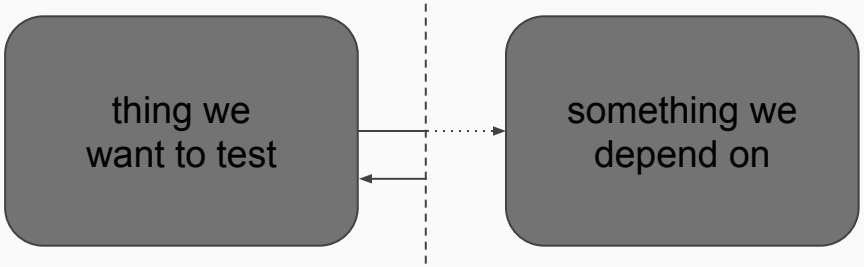
- Unit tests
  - Test one simple method. Don’t care about anything else (using mocking).
- Feature tests
  - Test one API endpoint of a service. Mock everything else than this service.
- Scenario tests
  - Test one user journey. Mock only external systems.
- Load tests
  - Test how much traffic the service can handle. Mock only external systems.

How do we write good tests?

- Cover the happy case (the easy part)
- Cover all the sad cases (the hard part)
  - Cover edge cases
  - Cover ‘illegal’ input (null values, empty strings, special characters ++)
- Assert that the correct things happen.
- Don’t test the internals of the class, but the behavior:
  Refactoring that don’t change the API of the class should not break tests
- No side effects (don’t rely on execution order, state of DB ++)

Mocking

- We often want to test as small parts of the system as possible.
- To be able to do this all the other parts of the system should behave as we want it to for the purpose of the tests.
- This is called mocking.



Tools

- Java
  - Test frameworks
    - JUnit
    - AssertJ
  - Mocking
    - Mockito
    - EasyMock
- System testing
  - Gatling (load testing)
  - Selenium (front end testing)

Code example

```
@Test
public void testSomething() {
    // assertj
    assertThat(actual).isEqualTo(expected);
    assertThat(actualSet).containsOnly(expected1, expected2);

    // hamcrest
    assertThat(actual, is(expected));
    assertThat(actual, is(not(expected)));
}
```

Code example: Reverse a string

```
@Test
public void testReverse() {
    // assertj
    assertThat(Strings.reverse("abc")).isEqualTo("cba");

    // hamcrest
    assertThat(Strings.reverse("abc"), is("cba"));
}
```

Code example: Reverse a string

```
@Test
public void testReverseWithNullValues() {
    assertThatExceptionOfType(NullPointerException.class)
        .isThrownBy( () -> Strings.reverse(null) )
        .withMessage("Input String cannot be null");
}

@Test
public void reverseEmptyString() {
    assertThat(Strings.reverse("")).isEqualTo("");
}
```

Code example: Mocking using Mockito

```
@Mock
private DatabaseClient databaseClient;

@Test
public void testDatabaseThingy() {
    when(databaseClient.store(anyString()).thenReturn(Status.SUCCESS);
    when(databaseClient.store(anyString()).thenThrow(new RunTimeException());

    Something something = new Something(databaseClient);
    (...)
}
```

Problem set

- Find the VAT of the price
- Palindrome
- Reverse a string
- Bowling
- Diff two files

https://github.com/kjheimark/abacourse2016

Find VAT of price

vat amount = price x vat rate

Palindrome

https://en.wikipedia.org/wiki/Palindrome

“A palindrome is a word, phrase, number, or other sequence of characters which reads the same backward or forward.”

Examples:

- abba
- agnes i senga

Reverse String

My fantastic string → gnirts citsatnaf yM

gnirts citsatnaf yM → My fantastic string

Bowling

Strike

- A strike is worth 10, plus value of your next two rolls

Spare

- A spare is worth 10, plus value of your next roll

Everything else

- Score is equal to the number of pins you knock down

http://bowling.about.com/od/rulesofthegame/a/bowlingscoring.htm

Diff two files

I am Groot		I am Groot
I'll be back		I'll be back
May the Force be with you	←	
His name is MacGyver; he can fix anything		His name is MacGyver; he can fix anything
	→	**Only a fool is sure of anything, a wise man keeps on guessing**