Writing (un)testable code

Maurício Aniche
Delft University of Technology
@mauricioaniche

Why should I write tests?

- Let’s code: Numbers.java

Because…

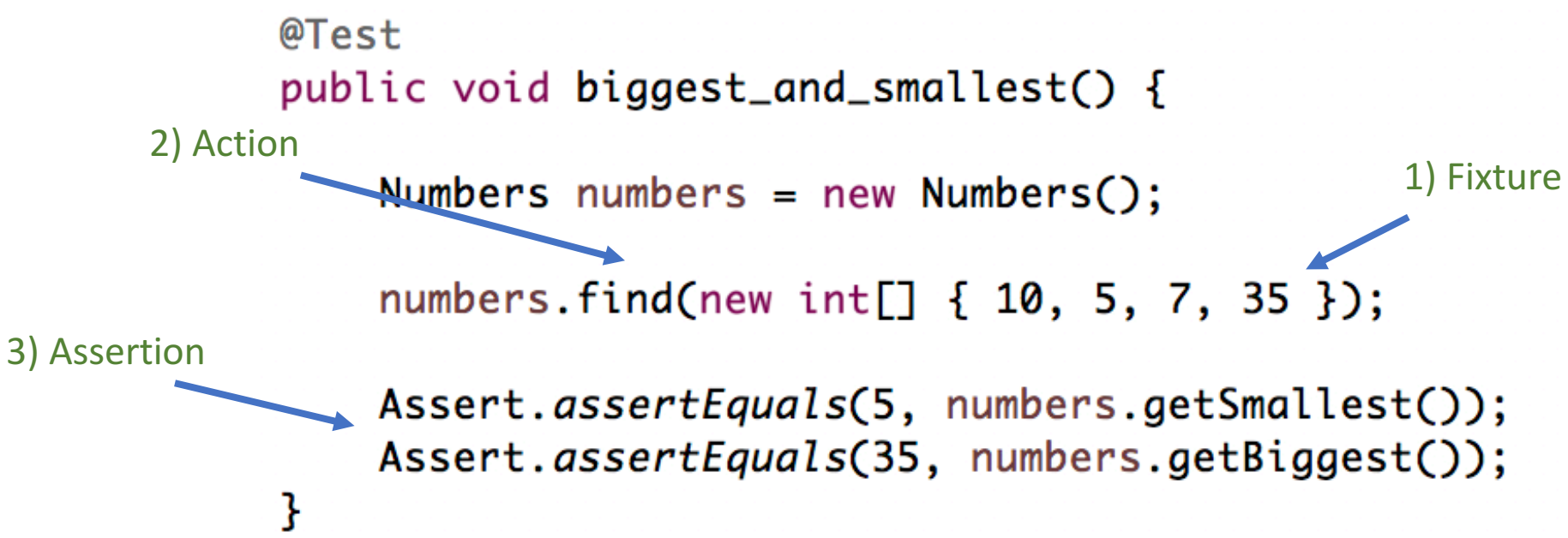
- Manual testing is too expensive
- It is too easy to make mistakes
- How can I make sure my new code doesn’t break the existing one?
- It’s the right thing to do

My Story on Dominican Republic!

Young me working on a low quality code that I wrote myself at my first international project (Dominican Republic, 2006).

Sorry about my hair.

What's an automated test?



In other words, a test is just a piece of code that invokes a method and tests its output.

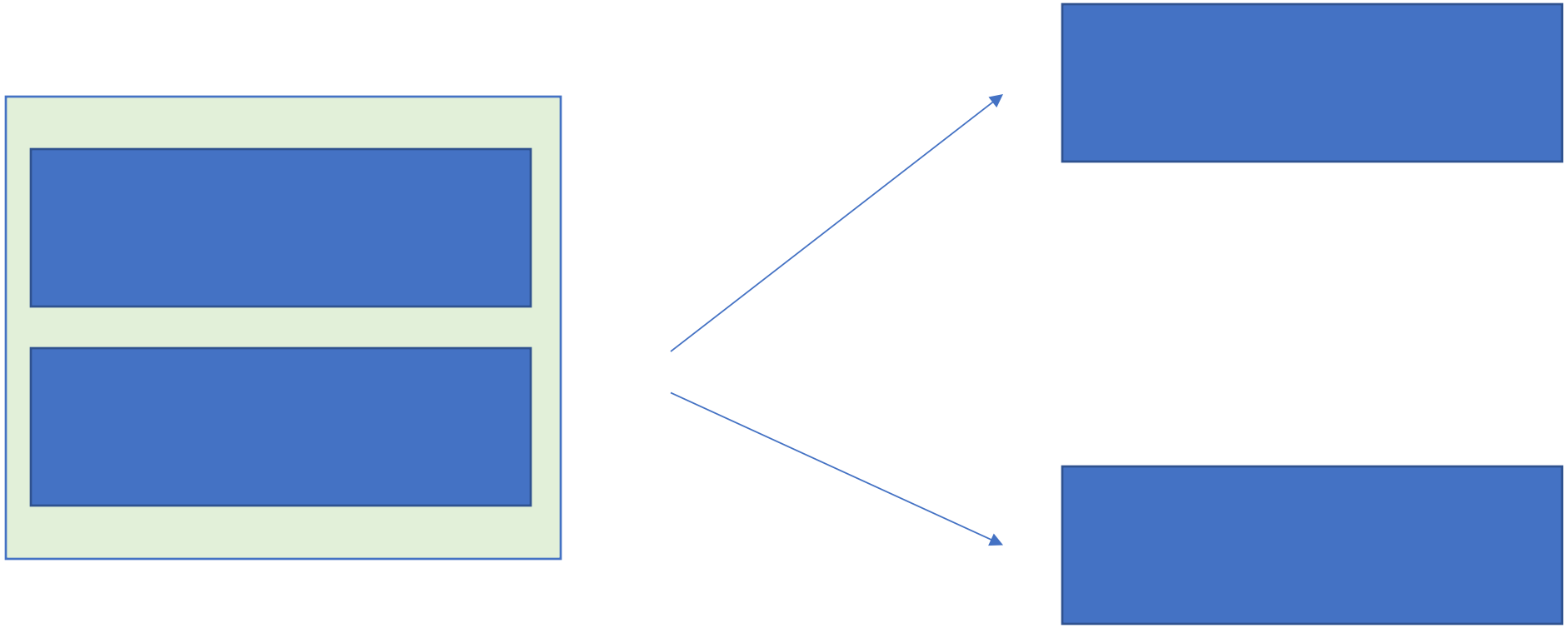
If you cannot do this easily, then you have an untestable code.

Can you tell me symptoms of untestable code?

Complex classes

If I have a method with 4 ifs, how many tests do I need? What about 8 ifs? …

Break the class!



Well-designed class -> testable class

- Cohesive classes!
- Single Responsibility Principle (SRP)
- Design patterns

Let’s code!

(SalaryCalculator.java)

Should I test a private method?

- Probably not.
- You should test the class from its public interface.
- If you want to test the private method in an isolated way, that's the test telling you that your code is not cohesive.

Dependencies

- What happens if I wanna test A, but A depends on B, which depends on C, which depends on D, …. ?
  - Fixture gets complicated
- Your test gets fragile!
  - What if a dependency changes?

Reduce coupling

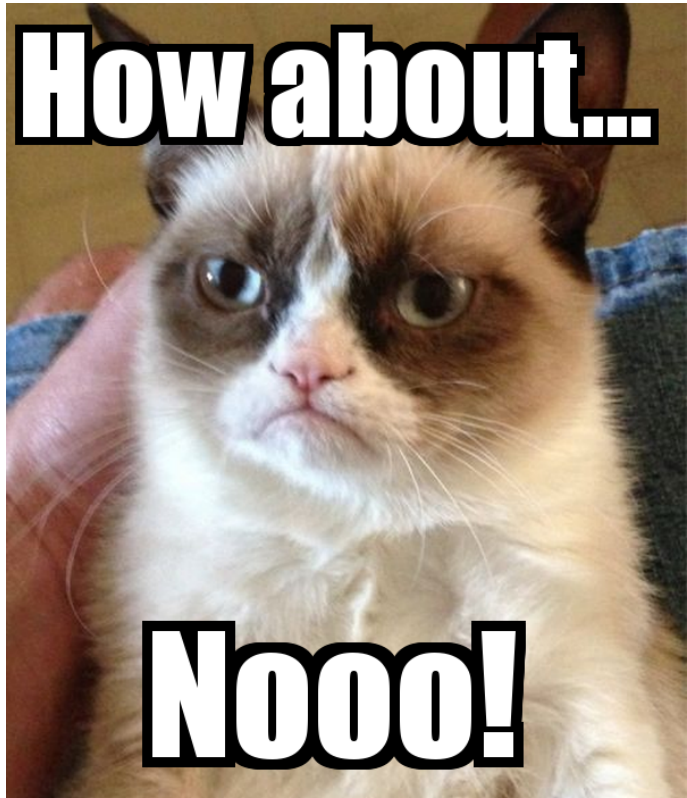
- Group dependencies
- Define better interfaces
- Rely on mock objects

Mock Objects

- Let’s code!
- (InvoiceFilter.java)
- (InvoiceGenerator.java)

Static methods

- Now that you know mock, what can we do with static methods?
- Static methods can’t be mocked.
- DON’T USE THEM.



Don’t be afraid of creating layers

- How can I test date/time related things?
- How can I test environment-dependent things?

- Create a layer on top of APIs
- These layers are easily mockable

Let’s code!

(Christmas.java)

Infrastructure

- My class depends on my database.
- My class depends on a file.

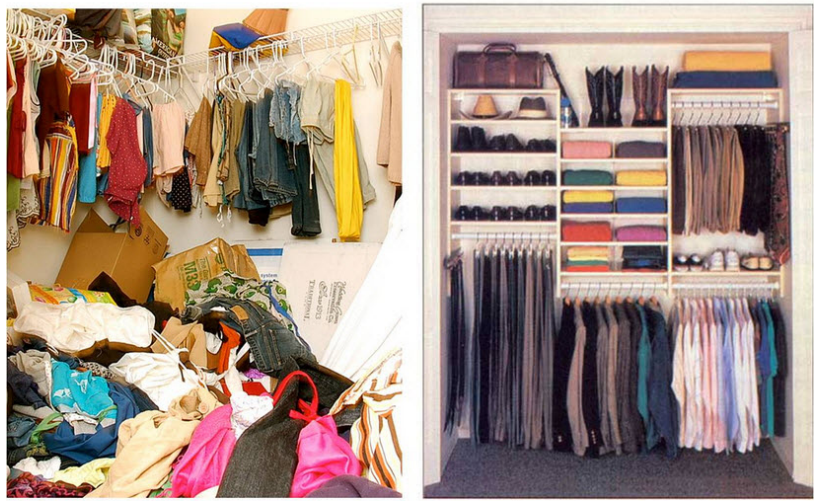
- Setting up databases or files can be expensive and hard.
- How can you simulate failures?

- (again InvoiceGenerator.java)

No infrastructure in your domain!

- You should isolate your infrastructure from the domain
- Hexagonal architecture / Ports and adapters, Evans' DDD

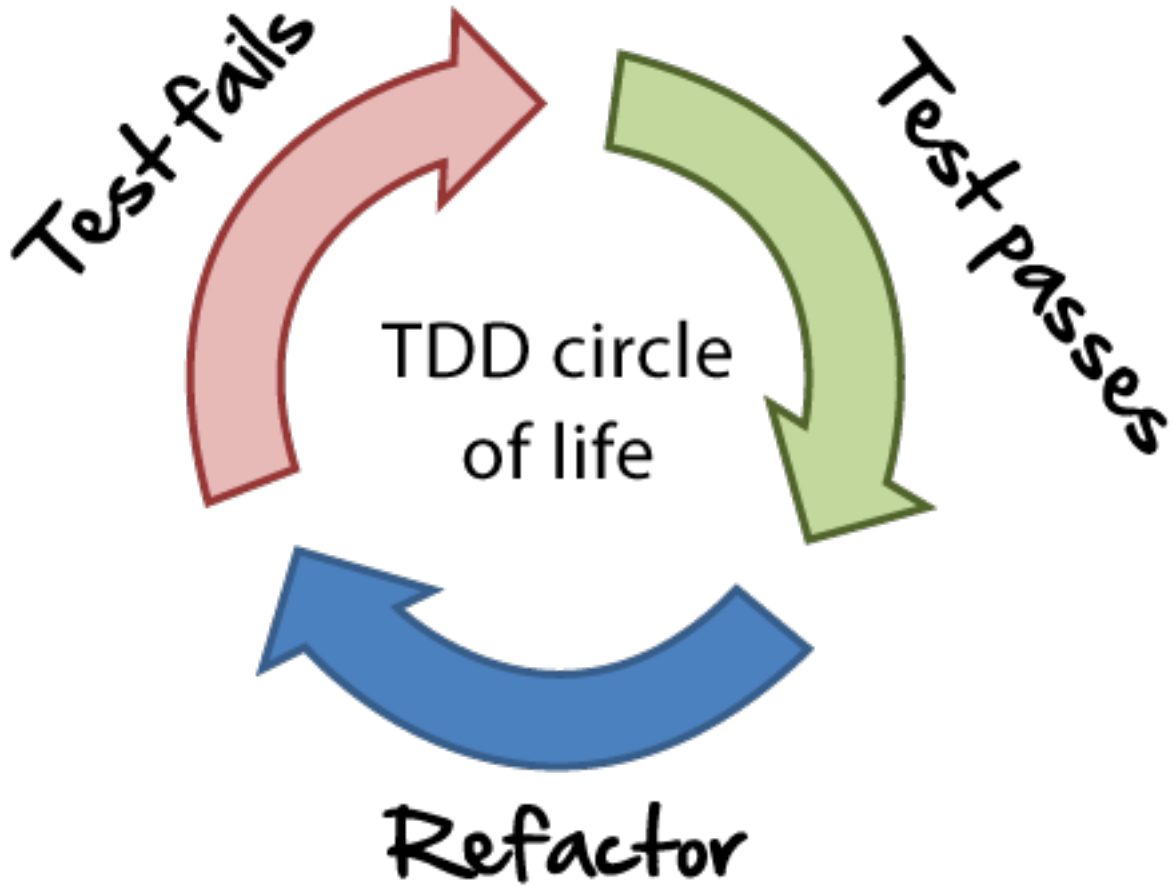
BEFORE AFTER



Encapsulation

- Behavior should be in the right test.
  - Otherwise, test gets too complicated.
- Demeter’s Law

Test-Driven Development



Advantages

- Forces you to write tests
- Forces you to refactor
- Forces you to write simple testable code

Baby steps

- You should go slow when you need to.
- Be smart.

Should I do it 100% of my time?

- Experience matters.
- Do it when you do not know what to expect… when you need to draft!
- Nevertheless, before or after… write tests!

Write testable code is not easy…
but necessary!

Writing (un)testable code

Maurício Aniche
Delft University of Technology
@mauricioaniche