

# The struggles of the new employee ;)

Luzern, 04. Juni 2021

Samuel Hopf

ALCOR Academy Training



# Agenda

- Why do I call it a struggle?
- Production Code Example
- What did we do?
- About the related Module Test
- Refactoring
- Learnings

# Why do I call it a struggle?

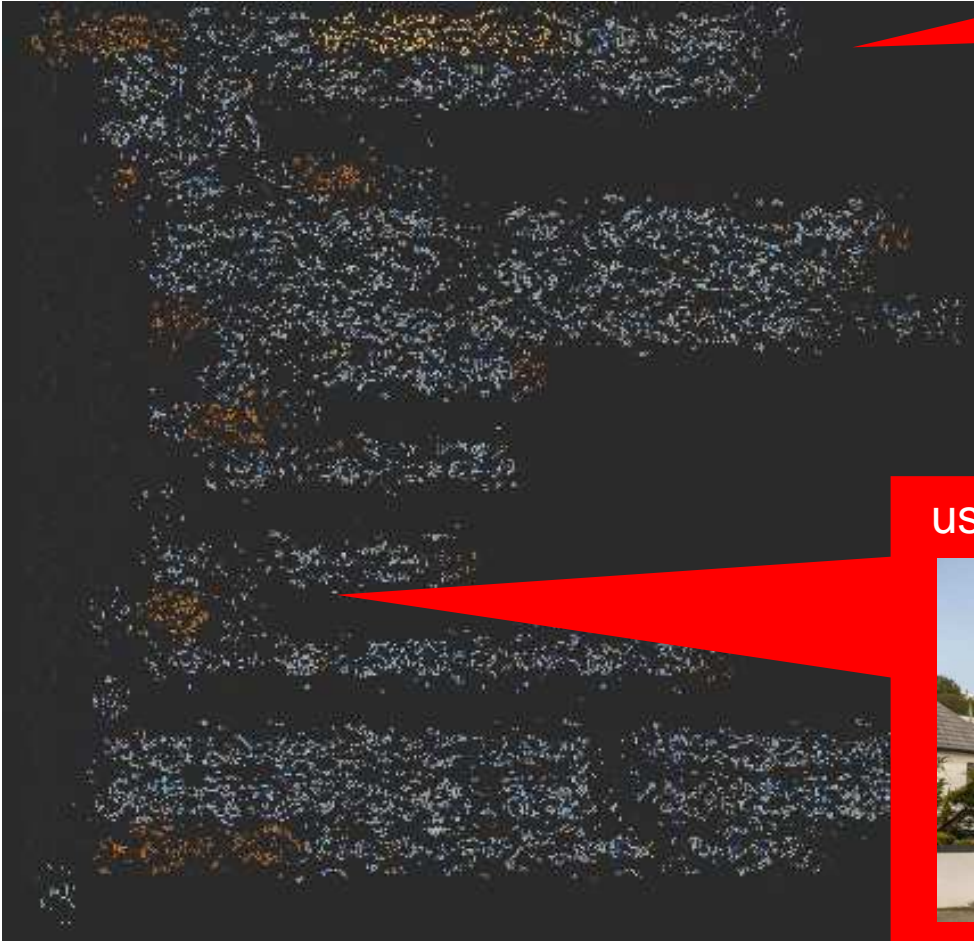
- Legacy Code almost always isn't...
  - easy and understandable
  - properly separated
  - clean
  - side effect free
- ... and almost never has complete test coverage
- Not being confident in the (new) domain does not help
- Refactoring is not always possible
  - There is no time
  - I have to write tests first
  - The scope of the issue is bigger than my knowledge

# Production Code Example



# How about Object Calistenics

It's a big class with big methods



has two levels of intendation



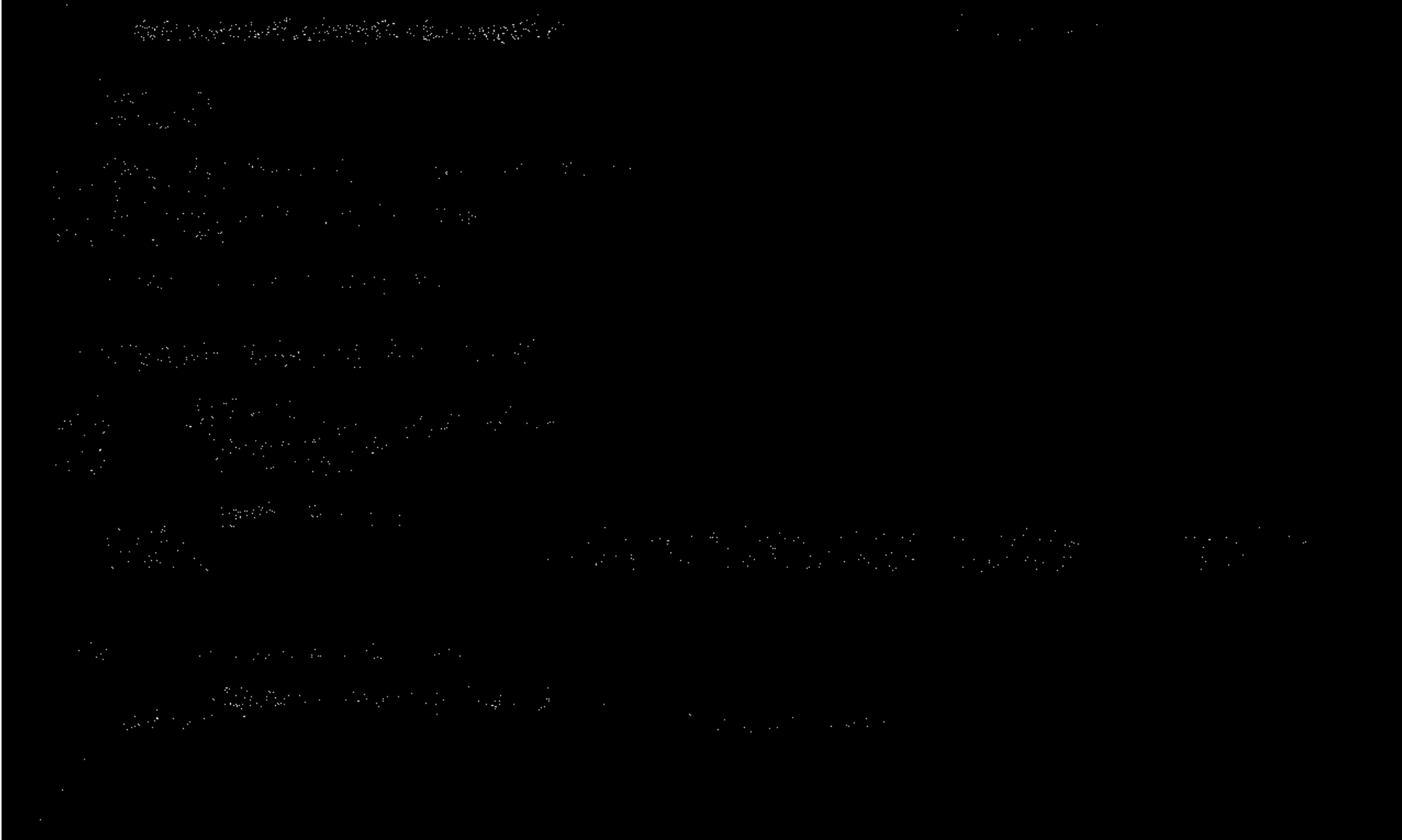
uses the ELSE keyword



# What did we do?

- Yes I know, don't check internals (Lesson 2 TDD Habits), but
  - It is critical
  - The method needs refactoring
- There was a module test we did extend
  - additional tests check all current behaviour
- We refactored before implementing the new feature
- We wrote a new test for the new behaviour
- We implemented the feature

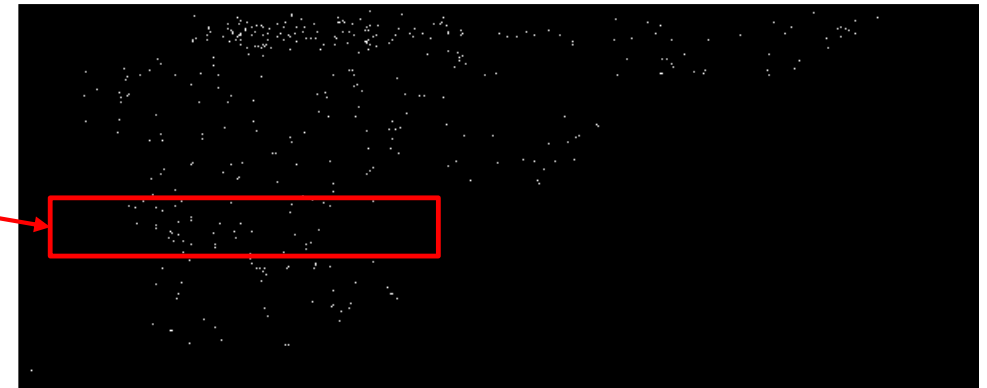
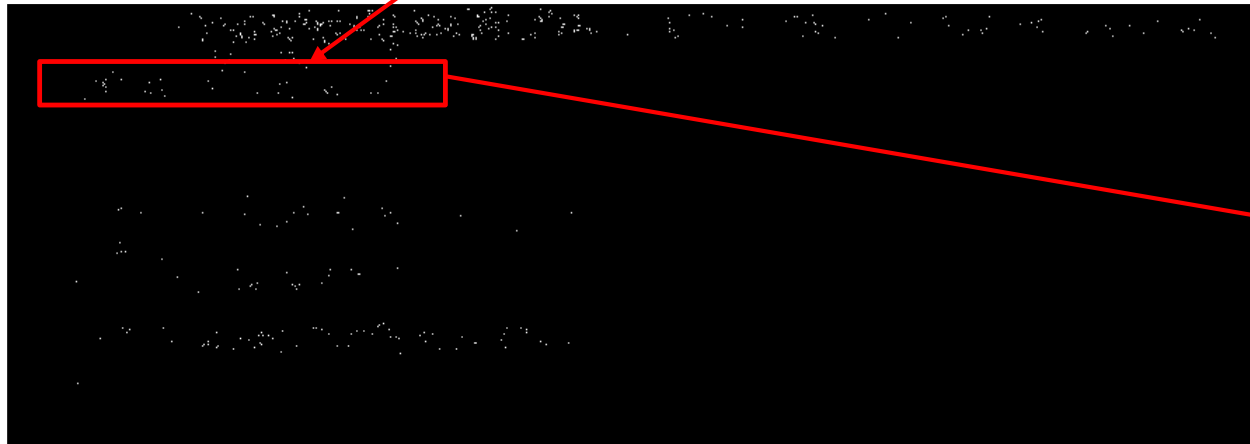
# About the related Module Test





# Module Test – Arrange

This defines the entrance date and has a wrong name ...





# Module Test – Assert

The image shows a screenshot of Verilog code with several red annotations. A large red box highlights a block of code containing an `assert` statement. A smaller red box highlights a specific line within that block, with a red arrow pointing to it from a red callout box containing the text "it asserts something here". Another red box highlights a different section of code below, with a red arrow pointing to it from a red callout box containing the text "... and something else here". The code is written in white text on a black background.

# What to do?



# Refactoring – Production Code



IntelliJ does not get it somehow

A white sculpture of a horse, possibly a classical or modern piece, standing on a base. The sculpture is shown in a close-up view, highlighting its form and texture.

# Refactoring –Test

```
function sum(a, b) {
  return a + b;
}

function sum(a, b) {
  return a + b;
}
```

I still have  
two different  
assertions



# Learnings

- 2 rules of the Object Calistenics made everything much easier
- test code is as important as production code

# Questions

Thank you for your patience – any Questions?

