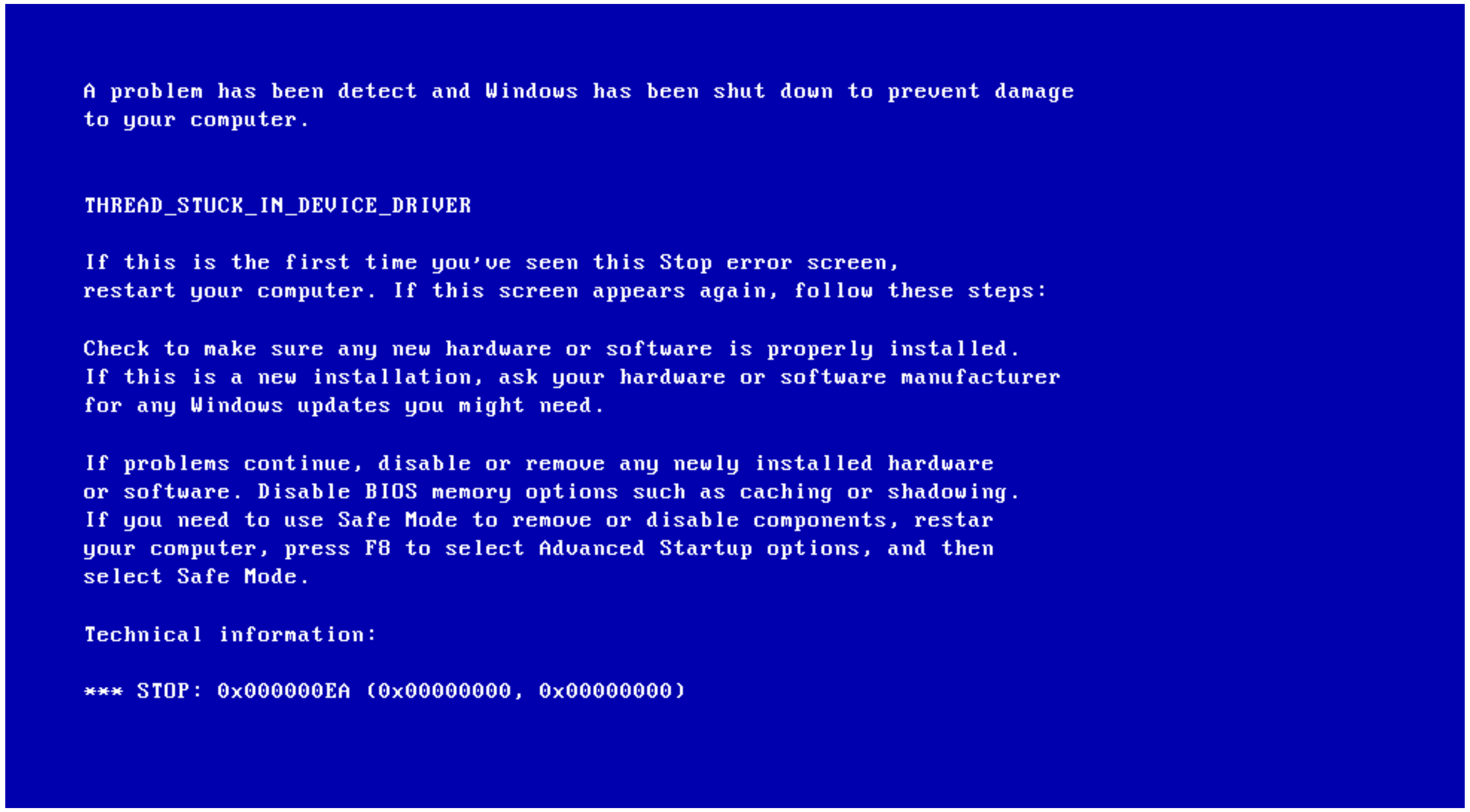
Property Based Tests

mit Scala

Ist Testen wichtig?



```
A problem has been detect and Windows has been shut down to prevent damage
to your computer.


THREAD_STUCK_IN_DEVICE_DRIVER

If this is the first time you've seen this Stop error screen,
restart your computer. If this screen appears again, follow these steps:

Check to make sure any new hardware or software is properly installed.
If this is a new installation, ask your hardware or software manufacturer
for any Windows updates you might need.

If problems continue, disable or remove any newly installed hardware
or software. Disable BIOS memory options such as caching or shadowing.
If you need to use Safe Mode to remove or disable components, restar
your computer, press F8 to select Advanced Startup options, and then
select Safe Mode.

Technical information:

*** STOP: 0x000000EA (0x00000000, 0x00000000)
```

Ein Test in Scala?

```scala
class FunctionsSpec extends WordSpec with Matchers {

  "Functions.add" should {

    "add two integers" in {
      Functions.add(2, 4) shouldBe 6
    }
  }
}
```

Ein Test in Scala?

```scala
class FunctionsSpec extends WordSpec with Matchers {

  "Functions.add" should {

    "add two integers" in new Fixture(2, 4) {
      result shouldBe 6
    }
  }

  class Fixture(a: Int, b: Int) {
    // ...
    // preparation
    // ...
    val result: Int = Functions.add(a, b)
  }

}
```

ScalaTest - Matcher

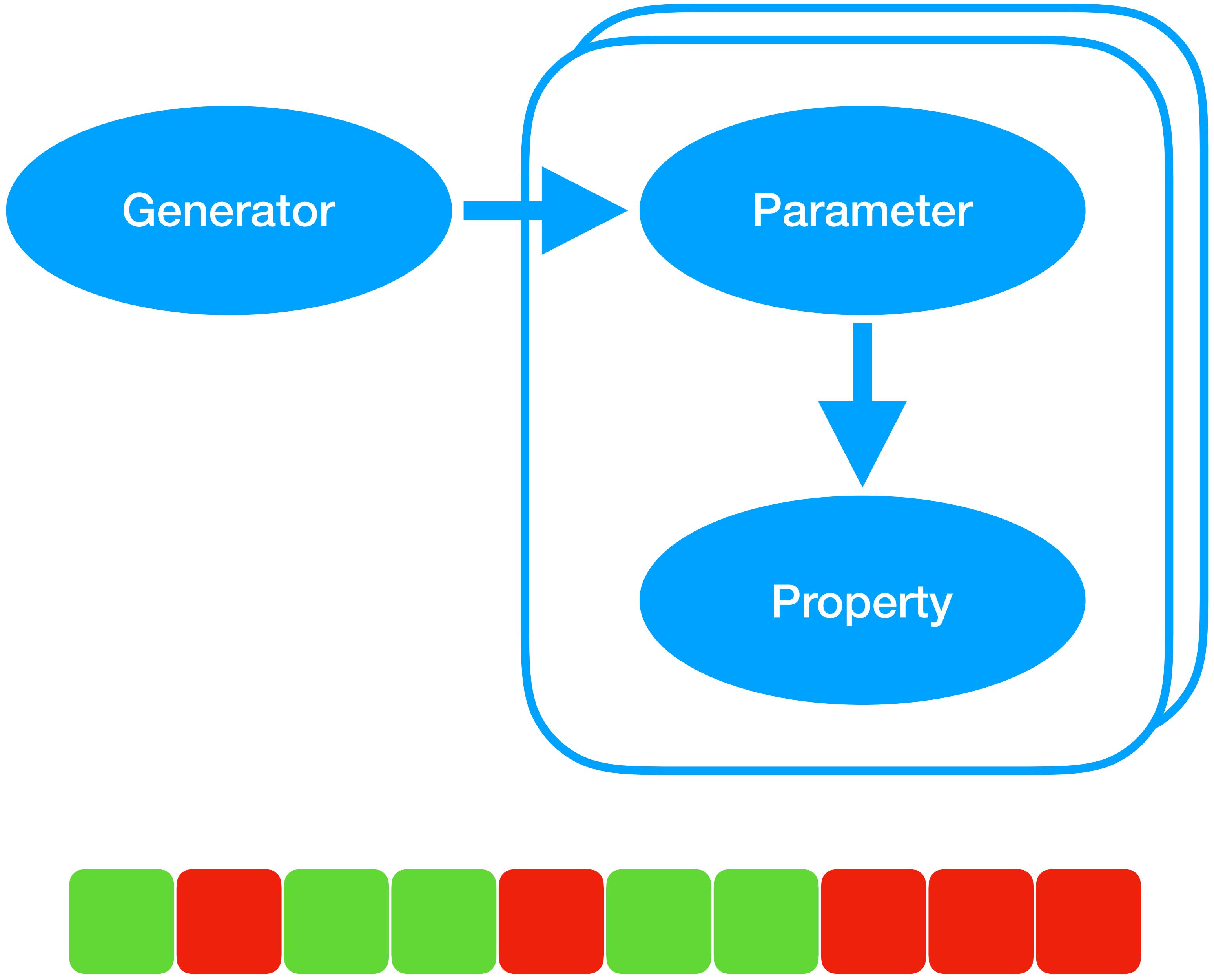
- result shouldBe 3
- result should have length 3
- result should have size 10
- string should startWith ("Hello")
- string should endWith regex "wo.ld"
- string should include regex "wo.ld"
- one should be <= 7
- one should be >= 0
- List(1, 2, 3) should contain theSameElementsInOrderAs List(…)
- "yellow" should (equal ("blue") and equal { println("hello, world!"); "green" })

Alternative?

Property Based Test

Aber was ist eine „Property“?

Wie funktioniert denn so ein Test?



PBT mit ScalaCheck

```scala
object FunctionsSpecification extends Properties("Functions") {

  property("add should form the sum of a and b") = forAll {
    (a: Int, b: Int) => Functions.add(a, b) == a + b
  }

}
```

PBT mit ScalaCheck

```scala
object FunctionsSpecification extends Properties("Functions") {

  property("add should form the sum of a and b") = forAll {
    (a: Int, b: Int) => Functions.add(a, b) == a + b
  }

  property("add is be commutative") = forAll(myGen, myGen) {
    (a: Int, b: Int) =>
      Functions.add(a, b) == Functions.add(b, a)
  }

  val myGen: Gen[Int] = Gen.choose(0, 100)
}
```

PBT mit ScalaCheck

```scala
object FunctionsSpecification extends Properties("Functions") {

  property("add and communitative“) = forAll {
    (a: Int, b: Int) => Functions.add(a, b) == a + b
  } && forAll(myGen, myGen) {
    (a: Int, b: Int) =>
      Functions.add(a, b) == Functions.add(b, a)
  }

  val myGen: Gen[Int] = Gen.choose(0, 100)
}
```

PBT mit ScalaTest

No more Slides!