

Aufgabenblatt 8

Aufgabe 1

Als Beispiel für den Einsatz Zustandsmonade dient oft die Erzeugung von Zufallszahlen.¹ Ein Beispiel für eine **imperative** Berechnung, die Zufallszahlen nutzt, ist `testQuicksort` in:

```
val initialSeed: Int = 1234567890
var r_n: Int = initialSeed

def nextRandom(): Int = { // ohne viel Nachdenken
  val a = 48247L // Primzahl
  val c = 11L // Primzahl
  val m = (1L << 31) - 1
  r_n = ((a*r_n + c) % m).toInt
  r_n
}

def quicksort(lst: List[Int]): List[Int] = lst match {
  case Nil => Nil
  case _ :: Nil => lst
  case first :: _ =>
    val pivot = first
    val l1 = lst.filter( _ < pivot)
    val l2 = lst.filter( _ == pivot)
    val l3 = lst.filter( _ > pivot)
    quicksort(l1) ++ l2 ++ quicksort(l3)
}

// Berechnung die Zufallszahlen nutzt
def testQuicksort(): List[Int] = {
  val a0 = nextRandom()
  val a1 = nextRandom()
  val a2 = nextRandom()
  val a3 = nextRandom()
  val lst = List(a0, a1, a2, a3)
  quicksort(lst)
}
```

1. Entwickeln Sie daraus eine funktionale Version mit Zustandsmonade.
2. Und wie ist es mit folgender zustandsabhängiger **imperativer** Berechnung:

```
def testQuicksortN(n: Int): List[Int] = {
  val lb = scala.collection.mutable.ListBuffer.empty[Int]
```

¹ So in der Cats-Dokumentation <https://typelevel.org/cats/datatypes/state.html>.

```
for (i <- 0 until n) {  
  lb.append(nextRandom())  
}  
quicksort(lb.toList)  
}
```