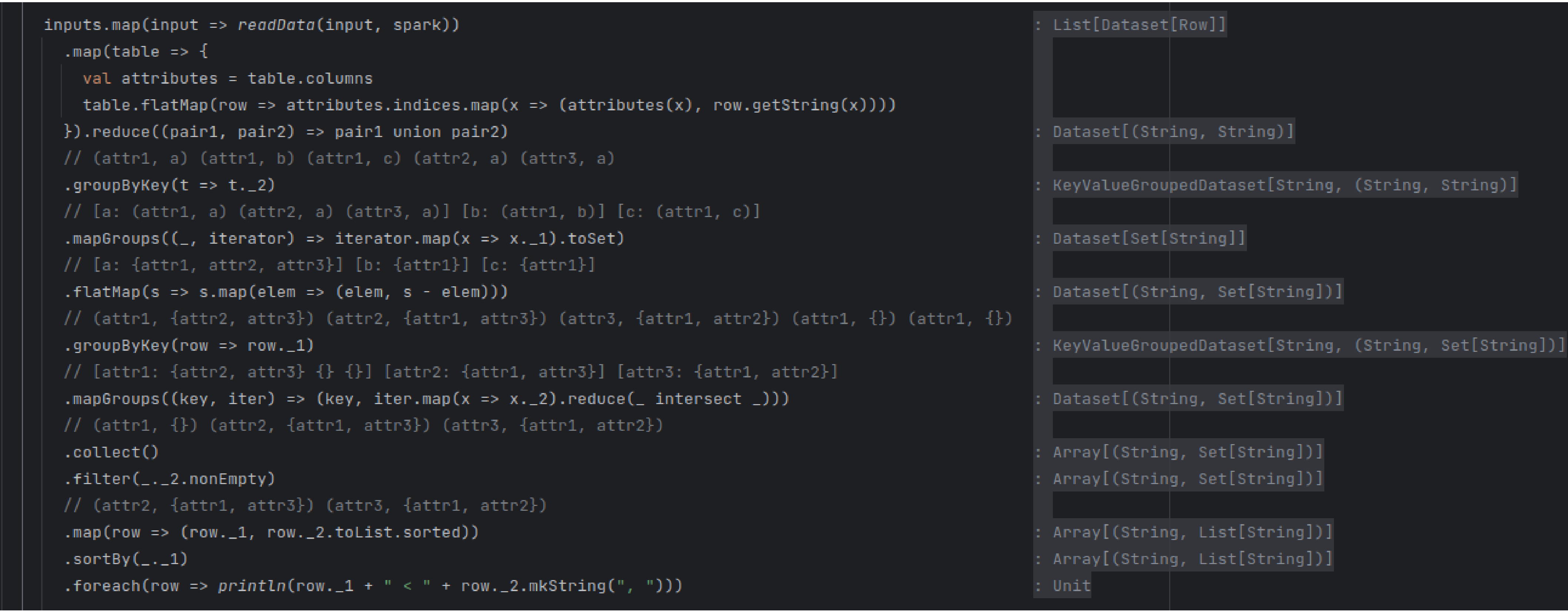
die-sich-vom-akka-machen



```
inputs.map(input => readData(input, spark))                                                              : List[Dataset[Row]]
  .map(table => {
    val attributes = table.columns
    table.flatMap(row => attributes.indices.map(x => (attributes(x), row.getString(x))))
  }).reduce((pair1, pair2) => pair1 union pair2)                                                         : Dataset[(String, String)]
  // (attr1, a) (attr1, b) (attr1, c) (attr2, a) (attr3, a)
  .groupByKey(t => t._2)                                                                                 : KeyValueGroupedDataset[String, (String, String)]
  // [a: (attr1, a) (attr2, a) (attr3, a)] [b: (attr1, b)] [c: (attr1, c)]
  .mapGroups((_, iterator) => iterator.map(x => x._1).toSet)                                             : Dataset[Set[String]]
  // [a: {attr1, attr2, attr3}] [b: {attr1}] [c: {attr1}]
  .flatMap(s => s.map(elem => (elem, s - elem)))                                                         : Dataset[(String, Set[String])]
  // (attr1, {attr2, attr3}) (attr2, {attr1, attr3}) (attr3, {attr1, attr2}) (attr1, {}) (attr1, {})
  .groupByKey(row => row._1)                                                                             : KeyValueGroupedDataset[String, (String, Set[String])]
  // [attr1: {attr2, attr3} {} {}] [attr2: {attr1, attr3}] [attr3: {attr1, attr2}]
  .mapGroups((key, iter) => (key, iter.map(x => x._2).reduce(_ intersect _)))                            : Dataset[(String, Set[String])]
  // (attr1, {}) (attr2, {attr1, attr3}) (attr3, {attr1, attr2})
  .collect()                                                                                             : Array[(String, Set[String])]
  .filter(_._2.nonEmpty)                                                                                 : Array[(String, Set[String])]
  // (attr2, {attr1, attr3}) (attr3, {attr1, attr2})
  .map(row => (row._1, row._2.toList.sorted))                                                            : Array[(String, List[String])]
  .sortBy(_._1)                                                                                          : Array[(String, List[String])]
  .foreach(row => println(row._1 + " < " + row._2.mkString(", ")))                                       : Unit
```