Learning by Example: checksum

In this lesson, you will learn how to generate the checksum of a string.

We'll cover the following

- ChecksumAccumulator
  - Checksum
  - The ChecksumAccumulator Class

`ChecksumAccumulator`

To understand the concepts of the next lesson, we will be using Martin Odersky's (inventor of Scala) `ChecksumAccumulator` program. Before we move on, let's briefly go over checksum.

Checksum

A checksum is a string of characters consisting of numbers and letters. It acts as a fingerprint for files and messages and is used to ensure the file or message you have received is genuine and error free. To do so, you create a checksum of the file you have received and compare it to the checksum of the source file. Even the most minor of changes will result in a completely different checksum.

> For the scope of this course, we will be generating a checksum of a string.

The `ChecksumAccumulator` Class

The `ChecksumAccumulator` class has a single private field value `sum`, which is initialized to **0**. It further has two methods, namely `add`, which is adding the bytes of a string, and `checksum` which is calculating the final checksum. `checksum` first uses the bitwise operator `&` with the final sum and the hexadecimal `0xFF`. It then takes the result of the `&` operator and applies another bitwise operator to it `~`. Finally, it adds a **1** to the result of the `~` operator, resulting in the final checksum value.

```
class ChecksumAccumulator {
  private var sum = 0
  def add(b: Byte) = sum += b
  def checksum() = ~(sum & 0xFF) + 1
}
```

This is a predefined algorithm which we are implementing. If you do not understand the algorithm, it's okay. You just need to know that it is used to generate a checksum.

In the next lesson, we will use `ChecksumAccumulator` to introduce singleton objects.