Object-Oriented Programming 2

Rolf Haenni & Andres Scheidegger

Exercises 4

1. Stream Implementation

Take the solution of Exercise 1 of Topic 1 about series of values. Extend the class `Series<V>` with the following two methods:

- `Series<V> limit(long maxLength)`: limits the length of a given series to a maximal value
- `Series<V> map(Function<V> function)`: applies a function to every element of a given series

With these methods, your series implementation should behave like Java 8 streams:

```
1 public static void main(String[] args) {
2
3   Series<Integer> s1 = new Series<>(0, x -> x + 1, x -> x <= 10);
4   Series<Integer> s2 = s1.map(x -> x*x).limit(5);
5
6   System.out.println(s2.toList());
7   // prints [0, 1, 4, 9, 16]
8 }
```

2. F1 Car Racing Application

Consider the given classes `F1Car` and `F1Driver`. You can use them to generate a stream of 50 F1 cars by calling `Stream.generate(F1Car::new).limit(50)`.

Given such a stream of F1 cars, perform the following computations:

1. Print the three teams with the fastest cars.
2. Print the average speed of the cars with a driver who has won at least 8 races.

3. Print the first and last names of the 3 drivers who have won the smallest amount of races.