P. Thiemann, L. Fennell Sommersemester 2017

Programmieren in Java

`http://proglang.informatik.uni-freiburg.de/teaching/java/2017/`

rainfall

Niederschlagsmessungen

Woche 10 Aufgabe 1/4

Herausgabe: 2017-07-04
Abgabe: 2017-07-22

Achtung: beachten Sie unbedingt die allgemeinen Hinweise zur Abgabe auf der Homepage.

Project `rainfall`
Package `rainfall`
Klassen

Main
`public static List< Optional<Double> > rainfall(List<Double> readings)`

Die Funktion `rainfall` berechnet durchschnittliche Niederschlagsmengen aus einer Liste `readings`. Die Liste `readings` enthält Serien von Messergebnisse als `double`-Werte (≥ 0). Eine Serie wird durch den speziellen Markierungswert -999 abgeschlossen. Wenn in den Serien andere negative Werte auftauchen, so sind das Messfehler; diese sollen ignoriert werden. Messwerte, die nach der letzten Serie auftreten, sollen ebenfalls verworfen werden.

Das Ergebnis von `rainfall` ist eine Liste von `Optional<Double>` Werten, die für jede Serie in `readings` den Durchschnitt angeben, sofern dieser existiert. Die Klasse `java.util.Optional` ist in der Java-API Dokumentation beschrieben.

`https://docs.oracle.com/javase/8/docs/api/java/util/Optional.html`

Sie wird verwendet wenn, wie hier, eine Funktion oder Methode nicht immer Ergebnisse liefern kann, aber auch nicht gleich mit einer `Exception` abbrechen soll. Die Klasse `Optional` enthält dazu zwei Factorymethoden:

1. `Optional.of` wird verwendet, wenn ein Ergebnis berechnet wurde. Das Ergebnis kann dann mit `Optional.get` abgerufen werden.

2. `Optional.empty` wird verwendet, wenn kein Ergebnis berechnet wurde. `Optional.get` wirft in diesem Fall eine `NoSuchElementException`.

Ob ein `Optional` Wert ein Ergebnis enthält, kann mit `Optional.isPresent` geprüft werden.

Zu dieser Aufgabe gibt es keine weiteren Hinweise.

Beispieltests

```
package rainfall;

import org.junit.Test;

import java.util.Arrays;
import java.util.List;
import java.util.Optional;

import static org.junit.Assert.assertEquals;
import static org.junit.Assert.assertTrue;

public class ExampleTests {

    @Test
    public void rainfallTest1() {
        List<Optional<Double>> readings =
                Main.rainfall(Arrays.asList(
                        3.0, 0.0, -1.0, 7.0, -999.0, -999.0, 5.0,
                        3.0, -999.0, -1.0));
        List<Optional<Double>> expected = Arrays.asList(
                Optional.of(3.333), Optional.empty(), Optional.of(4.0));
        assertRainfallEquals(expected, readings, 0.001);
    }

    public static void assertRainfallEquals(List<Optional<Double>> expected,
                                            List<Optional<Double>> actual,
                                            double delta) {
        String listMessage = " Expected: " + expected + "\n Actual: " + actual;
        assertTrue("Sizes differ. \n" + listMessage,
                   expected.size() == actual.size());
        for (int i = 0; i < expected.size(); i++) {
            String message = "Difference at index " + i + "\n" + listMessage;
            Optional<Double> expectedElem = expected.get(i);
            Optional<Double> actualElem = actual.get(i);
            assertTrue(expectedElem.isPresent() == actualElem.isPresent());
            if (actualElem.isPresent()) {
                assertEquals(message,
                             expectedElem.get(), actualElem.get(), delta);
            }
        }
    }


}
```