

Aufgabe 1: Iteration versus Rekursion in Java, JUnit-Test

Abgabetermin: 06. Okt. 2014

Gegeben ist das Interface `SumUp` und ein JUnit4-Testtreiber dazu.

Implementieren Sie das Interface durch 2 Klassen: `IterativeSumUpImpl`, `RecursiveSumUpImpl`

Dabei sollen Sie die Methode

```
public long sumUp(final int[] numbers, int fromIndex)
```

verschieden implementieren:

1. **iterativ**: Mit einer Schleife die Reihung ab `fromIndex` durchlaufen und dabei alle Elemente aufsummieren.
2. **rekursiv**: Die Summe aus dem Element an der Stelle `fromIndex` und dem Rest der Reihung liefern.

Lassen Sie den Testtreiber in Ihrer Entwicklungsumgebung laufen, ohne ihn zu verändern.

Wie verhalten sich die Laufzeiten der beiden Testmethoden `iterativeSumUp` und `recursiveSumUp` zueinander?

Für die Ehre:

- Wie verhält sich eine rekursive Lösung nach dem algorithmischen Prinzip "Teile und Herrsche" dazu? Dazu bräuchte man allerdings zwei Indexparameter `fromIndex` und `toIndex`.
- Können Sie auch mit JVisualVM aus dem JDK oder einem anderen Werkzeug den Speicherverbrauch beider Lösungen bestimmen?

Hinweise

- Der JUnit-Runner in Eclipse zeigt die Laufzeit jeder einzelnen Testmethode an.
- Bei Rekursion müssen Sie immer eine geeignete Abbruchbedingung implementieren.
- Wenn Sie einen `StackOverflowError` erhalten, müssen Sie im Testtreiber `SumUpTest` die Konstante `_quantity` etwas geringer initialisieren.