



## Softwaregüte

WS 2012/13 – Übungsblatt 7

Ausgabe: 12. Dezember 2012

Abgabe: bis spätestens 20. Dezember 2012  
an [Christian.Seepold@uni-wuppertal.de](mailto:Christian.Seepold@uni-wuppertal.de)

### Aufgabe 1. *Quantoren*

Lesen Sie in

<http://informatik.karlheinz-hug.de/artikel/ForumWI01%20SdV.pdf>

über die Benutzung von Quantoren (was ist das?) nach. Welchen Einsatz sehen Sie für diese (Prädikatenlogik)? Was kann man ohne sie nicht spezifizieren?

Beschreiben Sie in diesem Zusammenhang folgende neue `algorithm`-Funktionen

```
#include <algorithm>
//C++11 code
//are all of the elements positive?
all_of(first, first+n, ispositive()); //false
//is there at least one positive element?
any_of(first, first+n, ispositive()); //true
// are none of the elements positive?
none_of(first, first+n, ispositive()); //false
```

und deren Anwendung (vergleiche [The Biggest Changes in C++11](#)).

### Aufgabe 2. *Spezifikation des arithmetischen Mittelwerts*

Schreiben Sie eine Spezifikation einer Funktion zur Berechnung des arithmetischen Mittelwerts zweier `double`-Argumente. Wo auf dem Zahlenstrahl sollte das Ergebnis in Bezug auf die beiden Argumente liegen? Welche anderen Eigenschaften sollte das Ergebnis mit den beiden Argumenten verknüpfen?

**Aufgabe 3.** *Implementierungsversuch: arithmetischen Mittelwert*

Es werde der arithmetische Mittelwert durch

$$Z := (X + Y)/2$$

beziehungsweise durch

$$Z := X + (Y - X)/2$$

berechnet.

Wie unterscheiden sich die Ergebnisse dieser beiden Algorithmen voneinander? (Wann liefert jeder der beiden Algorithmen einen Wert ungleich Unendlich, wann den Wert NaN?)

Wann sollte deshalb der Algorithmus 1, wann der Algorithmus 2 benutzt werden?

**Aufgabe 4.** *eine Fallstudie zur Implementierbarkeit von DbC in C++*

Lesen Sie in

<http://contractpp.sourceforge.net/>

über die Sourceforge-Fallstudie `Contract++`, DbC als C++-Bibliothek möglichst vollständig bereitzustellen. Beurteilen Sie die Umständlichkeit, mit der bei direkter Benutzung dieser Bibliothek Software geschrieben werden muß.

Schreiben Sie dazu eine zur Klasse `stack` in der gemäß Ottosen vorgeschlagenen C++-Erweiterung

<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2006/n1962.html>

analoge Klasse und vergleichen Sie nach Lesbarkeit, technischem Aufwand beim Programmieren, ...