



## Programming by Contract

SS 2007 – Übungsblatt 9

Ausgabe: 21. Juni 2007

Abgabe: bis spätestens 28. Juni 07  
in der Vorlesung

### Aufgabe 1. *simple\_stack0*

Übersetzen Sie das folgende Programm

[http://www.math.uni-wuppertal.de/~buhl/teach/exercises/PbC07/simple\\_stack0.cc](http://www.math.uni-wuppertal.de/~buhl/teach/exercises/PbC07/simple_stack0.cc)

und führen Sie es aus.

Ergänzen Sie genügend viele Testfälle, um alle Zusicherungen greifen gesehen zu haben.

Provozieren Sie durch absichtliche Implementierungsfehler die Verletzung jeder einzelnen Nachbedingung.

Welche (wünschenswerten) Nachbedingungen werden noch nicht spezifiziert?

Warum haben die Methoden `get_count()` und `item()` keine Nachbedingungen?

Erläutern Sie die Invariante der Klasse!

### Aufgabe 2. *Konstruktor*

Ergänzen Sie `simple_stack0.cc` durch einen weiteren Konstruktor mit zwei Parametern, einem Feld `G[]` und einem `int` für die Länge dieses Feldes. Das durch diesen Konstruktor erzeugte Exemplar soll durch die Elemente des Feldes vorgefüllt werden.

Vergessen Sie nicht, den Contract für diesen Konstruktor zu spezifizieren.

### Aufgabe 3. *Destruktor*

Ergänzen Sie analog den nötigen (virtuellen) Destruktor. Wie sieht hier der Contract aus?

**Aufgabe 4.** *Kopierkonstruktor*

Spezifizieren Sie für die Klasse `simple_stack0.cc` einen Kopierkonstruktor. Implementieren Sie ihn. Testen Sie mit genügend vielen Testdaten und benutzen Sie dann den Kopierkonstruktor zur Verbesserung der Lesbarkeit der Nachbedingungen der Klasse.

**Aufgabe 5.** *Wertzuweisungsoperator*

Spezifizieren Sie für die Klasse `simple_stack0.cc` einen Wertzuweisungsoperator.

Implementieren und testen Sie ihn. Welche Vorteile hat die Existenz dieses Operators für Contracts?