



Programming by Contract

WS 2005/2006 – Übungsblatt 9

Ausgabe: 22. Dezember 2005

Abgabe: bis spätestens 12. Januar 2006
in der Vorlesung
oder per E-Mail an c.markmann@uni-wuppertal.de

Aufgabe 1. *Nana-Konfiguration*

Konfigurieren Sie Ihren User-Account auf unserem Ausbildungscluster gemäß der Konfigurationsanleitung der Vorlesung (siehe auch <http://www.math.uni-wuppertal.de/~buhl/teach/exercises/PbC0506/ConfigureNana25.txt>).

Lesen Sie <http://www.math.uni-wuppertal.de/~buhl/teach/exercises/PbC0506/nana.pdf> und führen Sie ein paar erste Tests aus.

Erweitern Sie das Test-Hauptprogramm von

<http://www.math.uni-wuppertal.de/~buhl/teach/exercises/PbC0506/Quicksort6b.cc>

um genügend viele Testfälle. Begründen Sie, warum Ihre Testfälle Ihnen ausreichend erscheinen.

Aufgabe 2. *simple_stack0*

Übersetzen Sie das folgende Programm

http://www.math.uni-wuppertal.de/~buhl/teach/exercises/PbC0506/simple_stack0.cc

und führen Sie es aus.

Ergänzen Sie auch hier 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 3. *Konstruktor*

Ergänzen Sie `simple_stack.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 4. *Destruktor*

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

Aufgabe 5. *vektor4.cc*

Übersetzen und testen Sie

<http://www.math.uni-wuppertal.de/~buhl/teach/exercises/PbC0506/vektor4.cc>,

diskutieren Sie die vorhandenen Contracts und vervollständigen Sie die Klasse.