BERGISCHE
UNIVERSITÄT
WUPPERTAL

Prof. Dr. Hans-Jürgen Buhl
Praktische Informatik/Numerik

Fachbereich C
Mathematik und Naturwissenschaften,
Mathematik und Informatik

E-MAIL buhl@math.uni-wuppertal.de
WWW www.math.uni-wuppertal.de/~buhl

DATUM 15. Januar 2014

Softwarequalität

WS 2013/2014 – Übungsblatt 11
Ausgabe: 14. Januar 2014
Abgabe bis 23. Januar 2014 an: *simon.wyrostek@googlemail.com*

Aufgabe 1. *DbC in D*

Lesen Sie

http://dlang.org/dbc.html

und erläutern Sie in eigenen Worten, wie die Programmiersprache D SdV (=DbC) unterstützt. Nutzen Sie dabei zusätzlich die Tabelle auf Seite 22 der Materialsammlung.

Aufgabe 2. *Umbrello als externes Eclipse-Tool*

Schreiben Sie eine Benutzeranleitung zur Konfiguration von Eclipse Juno, so dass `umbrello` als externes Tool verfügbar wird.

Testen Sie die `umbrello`-Integration:

- Erzeugen eines UML-Klassendiagramms aus Eclipse heraus
- C++-Code-Generierung in den Eclipse-src-Ordner hinein
- Eclipse-Build des generierten Codes
- Debugging des Codes
- ...

Aufgabe 3. *simple_stack0*

Übersetzen Sie das folgende Programm

http://www.math.uni-wuppertal.de/~buhl/teach/exercises/PbC09/simple_stack0.cpp

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 4. *Konstruktor/Destruktor*

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, einen Contract für diesen Konstruktor zu spezifizieren.

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

Aufgabe 5. *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.