BERGISCHE UNIVERSITÄT
GESAMTHOCHSCHULE WUPPERTAL
GAUSS-STRASSE 20
42097 WUPPERTAL
(Korrespondenzanschrift)
42119 WUPPERTAL
(Lieferanschrift)
TELEX 8 592 262 bughw
TELEFAX (0202) 439-2901
TELEFON (0202) 439-1

*Fachbereich 7*
MATHEMATIK

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

e-mail: Juergen.Buhl@math.uni-wuppertal.de

# Programming by Contract

## WS 2003/2004 – Übungsblatt 8

15. Januar 2004
Ausgabe: 8. Januar 2004

**Aufgabe 1.** *nana-Modifikationen*

Vergleichen Sie Ihre gemäß Aufgabe 4 von Übungsblatt 7 durchgeführten Modifikationen Ihres Linux-Accounts mit denen der Datei

http://www.math.uni-wuppertal.de/~buhl/teach/exercises/PgmByContr0304/nana.tar.gz

und beschreiben Sie den Sinn der jeweiligen Modifikationen zu den Originaldateien in `/usr/bin` bzw. `/usr/include` (welche Fehler wurden jeweils behoben?).

**Aufgabe 2.** *Quicksort*

Die folgende Version von Quicksort wurde schon mit einigen Zusicherungen in nana versehen:

http://www.math.uni-wuppertal.de/~buhl/teach/exercises/PgmByContr0304/Quicksort6.cc

Testen Sie das Programm (bauen Sie dazu absichtlich Fehler in die Algorithmen ein, um die Zusicherungen greifen zu sehen).

Ergänzen Sie — falls nötig — weitere »Programming by Contract«-Klauseln.

**Aufgabe 3.** *Quicksort - Fortsetzung*

Welche Nachbedingungen fehlen noch in den Spezifikationen der beiden Funktionen? Versuchen Sie diese mit Hilfe des `C(...)`-Makros zu realisieren.

**Aufgabe 4.** *vektor4.cc*

Die folgende Klasse `vektor`

http://www.math.uni-wuppertal.de/~buhl/teach/exercises/PgmByContr0304/vektor4.cc

versucht schon an vielen Stellen sinnvolle »Contracts« und Zusicherungen einzusetzen. Erklären Sie in eigenen Worten, warum in welchen Methodenkategorien Vorbedingungen, Nachbedingungen, Klassen- und

Schleifen-Invariantenüberprüfungen sowie einfache Zusicherungen sinnvoll sind.

Bearbeiten oder erläutern Sie mindestens die mit „`<---`“ markierten Codestellen. Ersetzen Sie den Referenz-Komponenten-Zugriffsoperator durch eine Methode `changeValueAt(int i, double x)` und nutzen Sie dabei das volle Arsenal der nana-Hilfsmittel. Berichtigen Sie den Fehler in der Methode `ei()`.

Testen Sie auch hier das Programm durch vorübergehendes absichtliches Einbauen von Fehlern in die Algorithmen, um die Zusicherungen greifen zu sehen.