



Programming by Contract

SS 2007 – Übungsblatt 8

Ausgabe: 14. Juni 2007

Abgabe: bis spätestens 21. Juni 07
in der Vorlesung

Aufgabe 1. *Abstraktionen*

Verschaffen Sie sich einen Überblick zu Kapitel 37.5 von

<http://research.sun.com/projects/plrg/fortress.pdf>

und erläutern Sie kurz die abstrakten Datenkapseln Monoid, Gruppe, Ring und Körper, die in Fortress für die Bibliothekprogrammierung zur Verfügung stehen.

Zeichnen Sie ein Hierarchiediagramm (*is-a*).

Wo werden diese abstrakten Datenkapseln zum Beispiel im Kapitel 38 (Zahlen) benutzt?

Für welche Zwecke kann der Fortress-Compiler die Kenntnis dieser Strukturen bei den Zahlentypen einsetzen?

Aufgabe 2. *Abstraktionen für lesbare Programme*

Übersetzen Sie das Programm

lu-gauss.cc

und führen Sie eine größere Anzahl von Testläufen durch.

Wie unterscheidet es sich von einer Version ohne Benutzung der Bibliothek `newmat10`? Wie können Sie einen Kollegen in einem programmierten Team von der Richtigkeit Ihres Codes überzeugen?

Übersetzen Sie das Programm erneut ohne die `-O 2` und statt dessen mit der `-g` Option. Führen Sie es sodann schrittweise im Debugger `ddd` aus (Next), nachdem Sie einen Breakpoint auf die Funktion `main()` gesetzt haben.

Aufgabe 3. *asserts zur Qualitätssicherung*

Übersetzen Sie das Programm

[lu-gauss-assert.cc](#)

und führen Sie die Testläufe erneut durch (jeweils mit dem Compiler mit der Option `-DNDEBUG` und ohne diese). Bauen sie absichtlich Fehler ein, übersetzen und testen Sie erneut.

Das Programm

[lu-gauss-nana.cc](#)

benutzt die `nana`-Sprachmittel `CHECK()` und `INVARIANT()`. Übersetzen und testen Sie es ebenfalls. Welche Vor- oder Nachteile hat es gegenüber der `assert`-Version? Übersetzen Sie mit jeder der möglichen Kombinationen der drei Compileroptionen `-DWITHOUT_NANA`, `-DNDEBUG` und `-DEIFFEL_CHECK=CHECK_LOOP` und vergleichen Sie die Längen der Binaries.

Welche anderen `EIFFEL_CHECK`-Optionen sind hier noch sinnvoll?

Aufgabe 4. *Contract*

Erzeugen Sie aus `lu-gauss-nana.c` eine neue Variante mit einer Funktion `zerlegeLU()` mit zugehörigem Contract (`REQUIRE()`, `ENSURE()`), Benutzung einer Exception sowie einem Test-Hauptprogramm. Testen Sie!

Aufgabe 5. *MFC mit Contracts*

Lesen Sie den Bericht

<http://archives.cs.iastate.edu/documents/disk0/00/00/01/04/00000104-01/TR95-17.pdf>

über den Einsatz von Contracts als Dokumentationshilfsmittel der „Microsoft Foundation Class Library“.

Referieren Sie in eigenen Worten den Inhalt des Abschnitts 9 (Conclusions).