



Programming by Contract

SS 2009 – Übungsblatt 9

Ausgabe: 30. Juni 2009

Abgabe: bis spätestens 8. Juli 2009
in der Vorlesung

Aufgabe 1. *nana-dir*

Benutzen Sie `nana-dir`, um alle Quelldateien im `src`-Verzeichnis in dokumentierende ShortForm (Deklarationen inklusive Contracts) umzuwandeln.

Schreiben Sie eine kurze Dokumentation zur Konfiguration von `eclipse` mit zwei `make` Targets

- `nana-dir`
- `clean`

`nana-sfg` ist nur für unter sehr restriktiven Stil-Richtlinien erzeugten Quellcode nutzbar:

- Funktionsdefinitionen müssen direkt am Zeilenbeginn starten
- ...

Vervollständigen Sie diese Stilrichtlinien. Sollte Ihr Programmierstil nicht dem von `nana-sfg` erwarteten Stilrichtlinien entsprechen, so haben Sie das Shell-Skript `nana-sfg` entsprechend anzupassen: „keep a version of this file and modify it for local taste.“

Aufgabe 2. *relative/absolute Abweichung*

Beim Arbeiten mit Gleitkommazahlen ist die Benutzung der relativen Abweichung empfehlenswert. Warum? (Lesen Sie dazu die auf Seite 54 der Materialsammlung zitierten Quellen.)

Wie ist die Maschinengenauigkeit ϵ für z.B. den Datentyp `double` definiert? Und wie für den Datentyp `long double`? (Überprüfen Sie die Werte in einem kurzen C++-Programm.)

In der Technik ist die alleinige Benutzung der relativen Abweichung als Kleinheitskriterium nicht immer geeignet. Warum? Was sollte gemäß <http://realtimecollisiondetection.net/blog/?p=89> als Alternative benutzt werden? Wie ist dazu die Hilfsfunktion `approximatelyEqualTo()` abzuändern?

Aufgabe 3. *Vervollständigung der Wuerfel-Contracts*

Die Contracts der Klasse `Wuerfel` verbieten den Methoden `Oberflaeche()`, ... nicht, den Wert des Attributs `Laenge` zu ändern.

Ergänzen Sie die Contracts der entsprechenden Methoden gemäß dem Beispiel auf Seite 25 der Materialsammlung, um die entsprechende Frame-Bedingung (Benutzung von `ID()` und `ENSURE()`) und testen Sie.

Aufgabe 4. *const-Methoden*

Ändern Sie die Klasse `Wuerfel` des letzten Übungsblatts alternativ zu Aufgabe 3 dadurch ab, dass Sie die abgeleiteten Abfragen zu `const`-Methoden machen und testen Sie wiederum.

Warum ist diese Vorgehensweise sinnvoller? Warum sollte auch in `const`-Methoden die Gültigkeit der Klassen-Invariante bei Methodenbeginn überprüft werden?