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 2. Juli 2015

Softwarequalität

SS 2015 – Übungsblatt 12
Ausgabe: 8. Juli 2015
Abgabe bis 15. Juli 2015 an: *mailto:125319@uni-wuppertal.de*

Aufgabe 1. *doxywizard-Integration in Eclipse*

Schreiben Sie eine Schritt-für-Schritt-Anleitung zum Einbinden des `doxywizard` als externes Tool in `eclipse`-Luna/Mars. Vergessen Sie dabei die Beschreibung der Aktivierung der Eclipse-Unterstützung des „documentation tool"s `doxygen` nicht (was wird hier zugeschaltet?).

Beschreiben Sie in je einer Fallstudie das Anlegen (und Dokumentieren) einer neuen Methode beziehungsweise eines neuen Attributs.

Beschreiben Sie gemäß

http://www.stack.nl/~dimitri/doxygen/docblocks.html

die notwendige weitere Präzisierung einer `doxygen`-Dokumentation eigener Quelltexte. Beachten Sie dabei insbesondere die Dokumentation der Vorbedingungen, der Nachbedingungen und der Klasseninvarianten gemäß Seite 103f. der Materialsammlung.

Aufgabe 2. *Wertzuweisungsoperator*

Spezifizieren Sie für die Klasse `simple_stack0.cpp` (des letzten Übungsblattes) einen Wertzuweisungsoperator.

Implementieren und testen Sie ihn. Welche Vorteile hat die Existenz dieses Operators für Contracts?

Aufgabe 3. *grundlegende Observatoren*

Was sind *grundlegende Observatoren*? Wie sollten sie spezifiziert werden (Notwendigkeit der Angabe von Vorbedingungen, ...)?

Schreiben Sie eine C++-Klasse `Polarkoordinaten`, die Oberservatoren für die x- und y-Koordinate sowie für Winkel und Länge enthält. Ergänzen Sie in Form von nana-Constructs die Spezifikationen für „basic queries" und für „derived queries".

Aufgabe 4. *Python DbC*

Lesen Sie im Link WiederverwendbareSoftware-Teil2.pdf des Abschnitts 1.12 der Materialsammlung die Seiten 26ff.

Wie sind Vorbedingungen, Nachbedingungen und Klasseninvarianten bei Benutzung von Pythons `contract`-Modul syntaktisch benutzbar? Wie greift man in Nachbedingungen auf die Werte von Parametern einer Methode zu (siehe auch pycontract)? Welches Problem gibt es nach "'PyContract PostCondition"' in Nachbedingungen?

Aufgabe 5. *Parkplatzproblem*

Schreiben Sie eine Funktion mit zwei Input-Parametern (m, n) und zwei Output-Parametern (P, M) zum in Abschnitt 1.9 gesprochenen *Parkplatzproblem*.

Ergänzen Sie diese Funktion um einen `nana`–Codevertrag und testen Sie.