



Programming by Contract

SS 2009 – Übungsblatt 4

Ausgabe: 18. Mai 2009

Abgabe: bis spätestens 27. Mai 2009
in der Vorlesung

Aufgabe 1. *Contracting in C++*

Lesen Sie

<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2006/n1962.html#vector-example-hpp>
und beschreiben Sie in eigenen Worten die Contracts der Konstruktoren
und der Methoden `erase`, `clear` sowie `swap` der Klasse `vector`.

Wird diese Contracting-Erweiterung im nächsten C++-Standard
realisiert werden?

(vgl. <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2008/n2705.html>)

Aufgabe 2. *Anti-structured Programming*

Welche Auswirkungen auf die Software-Güte hat

Spaghetticode (http://en.wikipedia.org/wiki/Spaghetti_code),
welche Raviolicode (http://en.wikipedia.org/wiki/Ravioli_code)?

Stellen Sie diese beiden Anti-Pattern in Zusammenhang mit strukturiert
er und unstrukturierter Programmierung!

Aufgabe 3. *Verwirrter Geldautomat*

Erläutern Sie, welche Software-Gütekriterien-Mißachtungen zum Problem
„*Verwirrter Geldautomat zahlt doppelt aus*“

<http://www.welt.de/die-welt/article3735762/Welt.html>

(Seite 5 der Materialsammlung) geführt haben (Erstellung einer
Relevanz-Matrix). Welche Unterschiede gibt es hier im Vergleich zur
Aufgabe 4 von Übungsblatt 3?

Aufgabe 4. *RISKS Digest*

Suchen Sie im Archiv des „RISKS Digest“ nach dem „great bankcard network breakdown“ und beantworten Sie auch für diesen Fall einer „Computer-Katastrophe“ die Fragen von Aufgabe 3.

Welche konstruktiven Maßnahmen in der Design- und Implementierungsphase der Software hätten das auf Seite 6 der Materialsammlung geschilderte Problem (THERAC 25) vermeiden helfen können?