



Programming by Contract

WS 2005/2006 – Übungsblatt 2

Ausgabe: 3. November 2005

Abgabe: bis spätestens 10. November 2005
in der Vorlesung
oder per email an c.markmann@uni-wuppertal.de

Aufgabe 1. *Software-Güte*

Lesen Sie die drei Artikel:

http://en.wikipedia.org/wiki/Coding_by_exception

http://en.wikipedia.org/wiki/Creeping_featurism

http://en.wikipedia.org/wiki/Accidental_complexity

Beschreiben Sie in eigenen Worten (in deutscher Sprache) die dort geschilderten „Anti-Pattern“.

Wie sind sie in Bezug auf die in der Vorlesung besprochenen produkt- und projektorientierten Qualitätsanforderungen einzuordnen?

Aufgabe 2. *Magische Zahlen*

Was versteht man beim Programmieren unter *magischen Zahlen*

(http://en.wikipedia.org/wiki/Magic_number_%28programming%29#Magic_numbers_in_code)?

Warum sind sie schlecht für die Software-Güte? Wie sollten sie vermieden werden?

Welche Qualitätsanforderungen der Vorlesung verletzt Code mit „magic Pushbuttons“ (http://en.wikipedia.org/wiki/Magic_pushbutton)?

Aufgabe 3. *Hard coding*

Welche Software-Gütekriterien verletzt das *hard coding*

(http://en.wikipedia.org/wiki/Hard_code)?

Aufgabe 4. *Anti-structured Programming*

Welche Auswirkungen auf die Software-Güte hat

http://en.wikipedia.org/wiki/Spaghetti_code,

welche http://en.wikipedia.org/wiki/Ravioli_code?

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

Aufgabe 5. *Abstraction Inversion*

Welchen grundsätzlichen Fehler macht ein Softwaredesign bei Abstraktionsumkehr (http://en.wikipedia.org/wiki/Abstraction_inversion)?