

Inhaltsverzeichnis

Metadokumente

- *TeachSWT – Status, History and License*

Konzeptpapiere und Übersichten

- *Softwaretechnik 2002 – Anleitung für den Assistenten und Erfahrungsbericht*
- *Softwaretechnik 2002 – Konzept zur Durchführung der Übungen*
- *Ablauf der Übungen und Scheinkriterien*

Runde 1

- *VDMTools am WSI*
- *Spezifikation eines assoziativen Arrays in VDM-SL*
- *VDM-SL-Beispiele aus dem Editor EFASS*
- *Übungsblatt 1: Spezifikation eines FiFo in VDM-SL*
- *Lösungsskizze 1: Spezifikation eines FIFO*

Runde 2

- *Module und Namensräume*
- *Von der Spezifikation zur Implementation*
- *Übungsblatt 2: Implementation eines FiFo*
- *Quelltext für eine Prioritätswarteschlange in Modula-2*
- *Lösungsskizze 2: Implementation eines FiFo*
- *Quelltext für einen FIFO in C++*

Runde 3

- *Spezifikation durch Funktionen und die Behandlung von Speicher*
- *Übungsblatt 3: Modellierung eines Heap in VDM-SL*
- *Lösungsskizze 3: Spezifikation eines Heap in VDM-SL*

Runde 4

- *Übungsblatt 4: Theatersitze, Make*
- *Quelltext für einen FIFO in C++*
- *Lösungsskizze 4: Saalbestuhlung, Makefile*

Runde 5

- *Das EVA-Paradigma*
- *Beschreibung von Modulen in natürlicher Sprache*

- *Übungsblatt 5: Entwurf eines Disassemblers*
- *Modulschnittstellen (Headerdateien) für einen 8-Bit-Disassembler in C++*
- *Lösungsskizze 5: Entwurf eines Disassembler*

Runde 6

- *Übungsblatt 6: Statische OO-Modelle*
- *Lösungsskizze 6: Statische OO-Modelle mit UML*

Runde 7

- *Übungsblatt 7: Dynamische Modellierung mit UML*
- *Lösungsskizze 7: Dynamische Modellierung mit UML*

Runde 8

- *Übungsblatt 8: Strukturierte Analyse, MVC-Pattern*
- *Lösungsskizze 8a: Strukturierte Analyse*

EFASS

- *EFASS (An Editor for A Small System) – Fallstudie zu modularem Entwurf und modularer Programmierung für die Softwaretechnikvorlesung 2002 in Tübingen*
- *EFASS (An Editor for A Small System) – Beschreibung und VDM-SL-Spezifikation*

TeachSWT

Übungsmaterial, Fallbeispiele und Ergänzungen zur Softwaretechnikvorlesung 2002 in Tübingen

The art of programming is the art of organizing complexity.

– EDSGER W. DIJKSTRA

The purpose of software engineering is to control complexity, not to create it.

– PAMELA ZAVE

Beauty is more important in computing than anywhere else in technology. [...] Beauty is important in engineering terms because software is so complicated. Complexity makes programs hard to build and potentially hard to use; beauty is the ultimate defense against complexity.

– DAVID GELERNTER

Beauty is truth, truth beauty, – that is all Ye know on earth, and all ye need to know.

– JOHN KEATS

Il semble que la perfection soit atteinte non quand il n'y a plus rien à ajouter, mais quand il n'y a plus rien à retrancher.

(Vollkommenheit entsteht offensichtlich nicht dann, wenn man nichts mehr hinzuzufügen hat, sondern wenn man nichts mehr wegnehmen kann.)

– ANTOINE DE SAINT-EXUPÉRY, TERRES DES HOMMES

Konzeptpapiere und Übersichten

- *Konzept zur Durchführung der Übungen*
- *Anleitung für den Assistenten und Erfahrungsbericht*

When you plan your next project, leave time at the end to administer a brief evaluation to the people involved. Remember to include these basic questions: What worked? What didn't? What could have been done better? What are some specific suggestions for improvement? Then take time to read over the answers, asking what you learned from the experience. Before beginning your next project, reflect on these lessons and include them in your plans.

– DIANE DREHER

Runde 1

- *VDMTools am WSI*
- *Spezifikation eines assoziativen Arrays in VDM-SL*
 - *VDM-SL-Beispiele aus dem Editor EFASS*
- *Spezifikation eines FiFo in VDM-SL (Übungsblatt 1)*
 - *Spezifikation eines FIFO (Lösungsskizze 1)*

There's no sense in being precise about something when you don't even know what you're talking about.

– JOHN VON NEUMANN

If a system doesn't have to be reliable, it can do anything else.

– H. H. WILLIAMS, *Oakland*

Runde 2

- *Module und Namensräume*
- *Von der Spezifikation zur Implementation*
- *Implementation eines FiFo (Übungsblatt 2)*
- *Quelltext für eine Prioritätswarteschlange in Modula-2*
- *Implementation eines FiFo (Lösungsskizze 2)*
- *Quelltext für einen FIFO in C++*

I invented the term Object-Oriented, and I can tell you I did not have C++ in mind.

– ALAN KAY

Auf Altnordaustralien waren die Insekten klein, fortlaufend nummeriert und genehmigungspflichtig.

– CORDWAINER SMITH in *Der Planetenkäufer*

Runde 3

- *Spezifikation durch Funktionen und die
Behandlung von Speicher*
- *Modellierung eines Heap in VDM-SL (Übungsblatt 3)*
- *Spezifikation eines Heap in VDM-SL (Lösungsskizze 3)*

Runde 4

- *Theatersitze, Make (Übungsblatt 4)*
- *Saalbestuhlung, Makefile (Lösungsskizze 4)*

Was der Autor mit vielen schönen Worten (Betonung auf viele) umschreibt, ist nichts anderes, als ein ungerichteter, [...] Graph:

- *Die Sitzplätze werden durch die Knoten des Graphen repräsentiert*
- *Hinter der Aussage „zwei Plaetze sind nebeneinander“ verbirgt sich: Es gibt eine zu beiden Knoten adjazente Kante.*
- *Die ominöse Operation suche-plaetze ist als Finden eines zusammenhängenden Teilgraphen mit n Knoten zu interpretieren.*

– *aus der Abgabe einer ÜBUNGSTEILNEHMERIN*

In the development of our understanding of complex phenomena, the most powerful tool available to the human intellect is abstraction.

– C.A.R. HOARE

Runde 5

- *Das EVA-Paradigma*
- *Beschreibung von Modulen in natürliche Sprache*
 - *Entwurf eines Disassemblers (Übungsblatt 5)*
 - *Entwurf eines Disassembler (Lösungsskizze 5)*

There are two ways of constructing a software design: One way is to make it so simple that there are obviously no deficiencies and the other way is to make it so complicated that there are no obvious deficiencies.

– C.A.R. HOARE

Clutter and confusion are failures of design, not attributes of information. There's no such thing as information overload.

– EDWARD TUFTE

Runde 6

- *Statische OO-Modelle (Übungsblatt 6)*
- *Statische OO-Modelle mit UML (Lösungsskizze 6)*

That is what modeling, Z for example, is all about. Abstract models of systems behaviors and properties that allow the use of mechanized analysis. UML is a contender put forward by OMG, however it requires you to think like OMG. When designing systems you should be free to think in terms of the system, not in terms of someone with a bridge to sell.

– BILL MILAM in comp.specification.z

The tools we use have a profound (and devious) influence on our thinking habits, and therefore, on our thinking abilities.

– EDSGER W. DIJKSTRA

Language shapes the way we think and determines what we can think about.

– BENJAMIN LEE WHORF, LANGUAGE, THOUGHT, AND REALITY

We have heard much discussion recently on the difference between a component and an object; and I have done so mostly with gasps of disbelief. [...] Components are just those objects that have been placed in the library under reuse management.

– IAN GRAHAM in Requirements Engineering and Rapid Development

Runde 7

- *Dynamische Modellierung mit UML (Übungsblatt 7)*
- *Dynamische Modellierung mit UML (Lösungsskizze 7)*

Time, if we can intuitively grasp such an identity, is a delusion: the difference and inseparability of one moment belonging to its apparent past from another belonging to its apparent present is sufficient to disintegrate it.

– JORGE LUIS BORGES in A New Refutation of Time

The curious thing is that this principle of encapsulation is regularly ignored by creators of object-oriented analysis methods, as we shall see later. So [...] we can only wonder at the extent of conservatism in most current object-oriented software engineering practice.

– IAN GRAHAM in Requirements Engineering and Rapid Development

Runde 8

- *Strukturierte Analyse, MVC-Pattern (Übungsblatt 8)*
- *Strukturierte Analyse (Lösungsskizze 8a)*

Get the data structures correct first, and the rest of the program will write itself.

– DAVID JONES, Assen, The Netherlands

EFASS

- *EFASS (An Editor for A Small System) – Beschreibung und VDM-SL-Spezifikation*
- *Auszüge aus den Quelltexten von EFASS*

Inside every large program there is a small program trying to get out.

– C.A.R. HOARE

Inside every EMACS, there is EFASS trying to get out.

– M E LEYPOLD

The End

There is a great danger associated with people's perception of new concepts. If improved methods are used to tackle the same sort of problems previously handled by ad hoc methods, the systems created could be far safer. If, on the other hand, the improved methods are used to justify tackling systems of even greater complexity, no progress will have been made.

– CLIFF B. JONES in *Systematic Software Development using VDM*