

## VL1: Einführung

### *Inhalt*

1. Organisation der Vorlesung
2. Problemstellung
3. Benutzte Werkzeuge

### **1. Organisaton**

Die wichtigsten Rahmeninformationen zur Lehrveranstaltung finden sich auf der Webseite: <http://www.fbmnd.fh-frankfurt.de/~doeben/II-PPmP/ii-ppmp05-header.html>

Weiteres zusätzliches Lehrmaterial für die Vorlesungen und für die Übungen befindet sich auf dem eLearning-Server <http://elearning.fh-basys.de/> unter fb2 Informatik und Ingenieurwissenschaften, dort der Kurs II-PPmP-SS05.

Für die Teilnahme am Kurs ist es unumgänglich, dass sich jeder auf dem eLearning-Server einmalig registriert und sich das Schlüsselwort für den Kurs vom Dozenten besorgt. Dieses Schlüsselwort wird nur beim ersten Einloggen benötigt; danach wird es automatisch ergänzt (Dazu müssen die Cookies aktiv sein).

Die Vorlesung ist projektorientiert. Dies bedeutet, dass das Lernen durch aktive Teilnahme an einem Projekt geschieht. Zu diesem Zweck werden Laborgruppen von max. 20 Teilnehmern gebildet und innerhalb dieser Laborgruppen gibt es Projektteams von mind. 2, höchstens 4 Teilnehmern. Pro Laborgruppe darf es nicht mehr als 6 Teams geben. Die Teams bekommen ein Leitthema für das gesamte Semester, das sich in 5 Teilthemen untergliedert, die im 2-Wochen-Rhythmus zu bearbeiten sind. Nach zwei Wochen muss jedes Team sein Ergebnis in einem 10-min.Vortrag vorstellen. Nach einer gemeinsamen Bewertung gibt der Dozent ein qualifiziertes Feedback und es gibt eine Diskussion der gesamten Laborgruppe.

In der Vorlesung wird kein druckreifes Wissen vermittelt, sondern es werden jeweils die Aufgabenstellungen für die nächste Projektetappe erläutert.

## 2. Problemstellung

Ziel für das gesamte Semester wird es sein, einen ASCII-Text (\_\_.txt), der Befehle für ein grafisches Objekt enthält, in ein Postscript-Dokument Level 2.0 zu verwandeln (\_\_.ps), das diese Textbefehle in eine grafische Seitengestaltung übersetzt. In einem zweiten Schritt werden diese Postscriptdateien dann wiederum automatisch in ein PDF-Dokument Version 1.2 bzw. 1.3 übersetzt (\_\_.pdf).

Zwar stellen PDF-Dokumente heute das üblichere Format dar, aber aufgrund ihrer komplexen Dokumentenstruktur und der verwendeten Komprimierungen der grafischen Objekte sind sie für Lernzwecke ungeeignet. Da PDF auf dem Postscript-Format aufsetzt und sich PS-Dateien direkt in PDF-Dateien konvertieren lassen, sind die Zieldateien im Semesterprojekt Postscriptdateien Level 2.0. Im Vergleich von PS-Dateien und PDF-Dateien werden die Grundzüge von PDF miterklärt.

Gegenüber PDF weist Postscript ferner den Vorteil auf, dass es eine vollständige Programmiersprache ist, sodass sich komplexe Abläufe direkt und intuitiv verständlich mittels der Postscript-Sprache formulieren lassen. Nebenbei kann man anhand von Postscript eine stackorientierte Sprache kennenlernen.

Nachdem möglichst schnell eine Prozesskette implementiert worden ist, die es den Studenten erlaubt, texte in Postscript- und dann PDF-Dateien zu übersetzen, wird es darum gehen, eine Diagrammsprache zu vereinbaren, die implementiert werden soll.

<b>ASCII-BEFEHLS- TEXT</b>	<b>/-- Übersetzung- --&gt;</b>	<b>POSTSCRIPT -BEFEHLE</b>	<b>/-- Übersetzung- --&gt;</b>	<b>PDF- BEFEHLE</b>
test1.txt	ppmp2ps	test1.ps	ps2pdf	test1.pdf

Innerhalb des allgemeinen Themas Text-to-PS könne sich die Teams unterschiedliche Bildwelten als Untersuchungsgegenstand aussuchen (2D-CAD, Comics, Landkarten, Schaltzeichen, ...).

### **3. Benutzte Werkzeuge**

Das gesamte Projekt wird unter Linux durchgeführt werden. Dabei kommen folgende Werkzeuge in Betracht:

- Linux-Shell
- Editor: Xemacs, Kate
- IDE: KDevelop Designer
- Compiler: GNU-C++
- Build: make
- Parser/Compiler: flex + bison
- Debugger: GNU gdb
- Modellierung: umbrello
- Dokumentation: doxygen
- Testen: ...
- PDF-Reader: acroread
- gv: Postscript Reader
- ps2pdf[12|13]: Postscript zu PDF-Konverter

Aufzeigen eines Beispiels, wie man von einem Text zu einem PDF-Dokument kommt.