

CES Softwareentwicklungspraktikum

Analyse- und Entwurfsdokument

Uschi Musterfrau, Detlef Mustermann und Ralf Auchmustermann

Matr.-Nr. 0815, 0816 und 0817

email: [uschi|detlef|ralf]@rwth-aachen.de

Inhaltsverzeichnis

Kapitel 1

Vorwort

1.1 Aufgabenstellung und Struktur des Dokuments

Kurzversion der Aufgabenstellung und Organisation des Dokuments (Wo steht was?)

1.2 Projektmanagement

Arbeitsaufteilung innerhalb der Gruppe (Wer macht was?)

1.3 Lob und Kritik

z.B. Danksagung an Betreuer

Kapitel 2

Analyse

2.1 Anforderungsanalyse

2.1.1 Benutzeranforderungen

Ausführliche Beschreibung der Aufgabenstellung (Ziel: Konsens zwischen Auftraggeber und Auftragnehmer)

Lösung nichtlinearer Gleichungssysteme resultierend aus Anwendung mit Newton's Methode [?, ?]; Problemspezifikation; Weiterverarbeitung der Resultate

2.1.2 Anwendungsfallanalyse

Anwendungsfälle (Statik: Anwendungsfalldiagramme; Dynamik: Aktivitätsdiagramme; Textuelle Beschreibungen laut Vorlage)

Systemanforderungen

Beschreibung funktionaler und nichtfunktionaler Anforderungen basierend auf Anwendungsfällen (Ziel: Konsens zwischen Auftraggeber und Auftragnehmer)

Benutzerdokumentation

kann basierend auf den Anwendungsfällen erstellt werden; siehe Kapitel ??

2.2 Begriffsanalyse

Identifikation von Klassenkandidaten basierend auf Anforderungen; Assoziationen zwischen Klassenkandidaten (Aggregation, Komposition, Vererbung) und Kardinalitäten

Kapitel 3

Entwurf

3.1 Grobentwurf: Subsysteme

Grobe Strukturierung in Subsysteme (evtl. Namensbereiche), z.B. Daten, Numerik, Grafik

3.1.1 Statik

3.1.2 Dynamik

3.2 Detailentwurf: Klassen

Klassenmodell (Statik: Klassendiagramm; Dynamik: Sequenzdiagramm, CRC Karten) inklusive (Klassen- bzw. Objekt-)Daten und (Klassen- bzw. Objekt-)Funktionen; Navigierbarkeit; Zugriffsrechte; Abstrakte Datentypen: vollständig spezifizierte fundamentale Datentypen, z.B. `vektor`, `matrix`; Axiome, Bedingungen, Testfunktionen

- Parser basierend auf `flex`¹ und `bison`² [?, ?] zur Eingabe der Daten (Problemspezifikation)
- Newton's Methode zur Lösung des nichtlinearen Systems [?]
- Direkte [?] bzw. indirekte Verfahren [?] zur Lösung des linearen Systems
- Automatisches Differenzieren [?, ?] zur Berechnung der Jacobimatrix bzw. des Produkts der Jacobimatrix mit einem Vektor

¹<http://www.gnu.org/software/flex/manual/>

²<http://www.gnu.org/software/bison/manual/>

3.2.1 Statik**3.2.2 Dynamik**

Kapitel 4

Benutzerdokumentation

wohlstrukturierte und gut lesbare Dokumentation basierend auf den Anwendungsfällen

Kapitel 5

Entwicklerdokumentation

wohlstrukturierte und gut lesbare Dokumentation der Software aus Entwickler-sicht; zahlreiche Referenzen nach Kapitel ?? und in den Quellcode in Kapitel ??

Die Signatur der Funktion `foo` finden Sie in Zeile ?? des Quellcodes ?? in Sektion ??.

Anhang A

Quellcode

einfach referenzierbare Version des Quelltexts

A.1 Paket 1

A.1.1 Klasse 1.1

Listing A.1: Dokumentierter Quellcode in `klasse1.1.hpp`

```
1 class foo {  
2     public:  
3         int bar(const float& f);  
4 }
```

A.1.2 Klasse 1.2

A.2 Paket 2

A.3 Paket 3