Prüfungsprotokoll

Art der Prüfung:	mündlich
Dauer:	ca. 45 Minuten
Studiengang:	TK/Informatik
Fächer:	Objektorientierte Softwarekonstruktion (**OOSK**), Software-Technik (**SWT**), Systemprogrammierung (**Syspro**)
Prüfer:	Prof. Horst Lichter
Note:	1,0

SWT:

1. Was ist ein Prozessmodell?
   → erklärt
2. Kennen Sie ein konkretes Modell?
   → Wasserfallmodell, Vor- und Nachteile
3. Kennen Sie bessere Modelle?
   → Inkrementelles Modell, Vorteile
4. Kann man immer inkrementell entwickeln? Wann geht das nicht? Beispiel?
   → Kam auf kein Beispiel….er löste das Rätsel: man kann nicht bei Systemen inkrementell entwickeln, die in Kompaktsysteme eingebaut werden, wie z.B. eine Zugsteuerung oder die Elektronik in einem Auto; System entweder ganz oder gar nicht
5. Es gibt dann ja das Requirements Engineering. Da gibt es verschiedene Phasen bzw. Abschnitte…Welche?
   → Durchführbarkeitsstudie
   Analyse und Bestimmung von Anforderungen
   Spezifikation von Anforderungen
   Validierung der Anforderungen
6. Was gibt es denn für Techniken bei der Analyse?
   → Ethnografischer, Viewpoint- und Szenarienbasierter Ansatz
7. Was sind denn Use Cases?
   → erklärt
8. Und welche Use Cases gibt es zum Beispiel bei einem Online-Flugbuchsystem?
   → Suchen, Registrieren, Buchen
9. Ja, man kann Use Cases ja auch in einem Diagramm darstellen. Zeigen Sie das mal mit den genannten Use Cases.
   → Use Case-Diagramm gemalt: Systemgrenzen, Akteur (Kunde), Use Cases, include-Beziehung
10. Und gibt es noch andere Beziehungen?
    → extend, Generalisierung
11. Ok, wie kann man denn Anforderungen unterteilen?
    → funktional, nichtfunktional
12. Und wo wären die funktionalen und nichtfunktionalen Anforderungen im Diagramm?
    → funktional: die Use Cases
    nichtfunktional: z.B. Hardware-Ressourcen, Bedienoberflächendesign
13. Gut, kommen wir nun zu Entwurfsprinzipien. Kennen Sie welche?
    → Minimale Kopplung, Maximale Kohäsion
14. Was heißt denn maximale Kohäsion?
    → Methoden & Daten innerhalb einer Klasse sollen gut zusammenarbeiten/aufeinander abgestimmt sein

15. Und wieso minimale Kopplung?
    → weil System flexibler und leichter änderbar
16. Und kennen Sie auch das Open/Close-Prinzip?
    → erklärt
17. Kennen Sie auch Entwurfsmuster? Welche?
    → Observer Pattern, Template Method, Factory Method, Adapter, Singleton, etc.
18. Welches Problem liegt denn beim Observer Pattern zugrunde?
    → Mehrere Klassen abhängig von einer, Muster erklärt und aufgemalt.

OOSK:

1. Objektorientiert gesehen gibt es ja Klassen und Objekte. Wo ist denn da der Unterschied?
   → erklärt
2. Abstrakte Klassen, was ist das denn?
   → wichtig hier: Schnittstelle!
3. Was ist denn eine abstrakte Methode? Warum macht man das denn so? Man könnte doch auch einfach in der Unterklasse redefinieren?
   → mit abstrakter Klasse können Methoden je nach Unterklasse völlig unterschiedlich definiert werden, Möglichkeit der Abstraktion; Beispiel: Oberklasse „Geometrische Form“, Unterklassen „Kreis“ und „Rechteck“, abstrakte Methode „Fläche berechnen“ in jeder Unterklasse ganz anders, Beispiel aufgemalt
4. Was heißt denn Vererbung? Diskutieren Sie doch mal die Vor- und Nachteile?
   → Vorteil: Vererbung kann Polymorphie realisieren, macht das System flexibler und unterstützt Wiederverwendung
   Nachteil: schnell unübersichtlich
5. Was ist Polymorphie? Woran sieht man die Polymorphie an dem gemalten Beispiel? Denken Sie an polymorphe Zuweisung.
   → ein statischer Bezeichner vom Typ der Oberklasse kann Objekte von verschiedenem Typ (Typ der Oberklasse oder der Unterklassen) zugewiesen bekommen; Bezeichner „Geo_Form“ vom Typ der Oberklasse „Geometrische Form“ kann ein Objekt vom Typ der Oberklasse, vom Typ der Unterklasse „Kreis“ oder vom Typ der Unterklasse „Rechteck“ zugewiesen bekommen.
6. Warum ist Übersicht so wichtig?
   → bei späteren Änderungen, wenn ein Entwickler nach einem halben Jahr noch mal auf das System schaut oder ein neuer Entwickler mit einem bereits bestehenden System arbeiten muss, dann dauert es lange bei häufiger Verwendung von Vererbung bis ein Verständnis für das System erlangt wird; vor allem bei Mehrfachvererbung!
7. Kommen wir nun zu Test. Warum macht man das?
   → um Fehler zu finden
   um nach eingearbeiteten Änderungen zu prüfen, ob alles funktioniert
8. Braucht man denn Änderungen?
   → Ja klar, denn die Welt ändert sich und die Wünsche/Bedürfnisse der Kunden und Nutzer somit auch.
9. Was gibt es denn für Test?
   → zwei Arten: Fehlertests und statistische Tests
10. Fehlertests? Was ist das denn?
    → Diese Tests testen auf Fehler, lokalisieren sie, aber beheben sie nicht. Statistische Tests testen auf Leistungsfähigkeit, Effizienz und Zuverlässigkeit.
11. Ja, bleiben wir bei den Fehlertests.
    → Da gibt es wiederum zwei Arten: Blackbox und Whitebox-Test; Vor- und

Nachteile erklärt, bei Whitebox-Test noch die Pfad-, Anweisungs- und Zweigüberdeckung erwähnt

12. Was ist denn Pfadüberdeckung? Warum ist das so aufwendig?
    → Bei endlosen While-Schleifen können nicht alle Pfade getestet werden; Lösung: Eingeschränkte Pfadüberdeckung

Syspro:

1. Warum heißt das eigentlich Systemprogrammierung? Programmierung hat ja was mit Herstellen zu tun, aber eigentlich stellt man doch Anwendersoftware her und nicht Systemsoftware.
   → Systemprogrammierung beschäftigt sich mit dem Betriebssystem, was auch ein Programm ist. Es liegt zwischen Nutzer und Hardware und verwaltet Hardware-Ressourcen. Der Unterschied zwischen Anwender- und Systemsoftware liegt eben darin, dass Systemsoftware hardwarenah und Anwendersoftware nutzernah und im Schichtenmodell über der Systemsoftware liegt.
2. Sie sagten, das Betriebssystem verwaltet Hardware. Welche denn?
   → E/A-Geräte, CPU-Zeit, Speicher
3. Nehmen wir mal an: Ein Prozessor und mehrere Programme. Was muss da gewährleistet werden?
   → Zeitverwaltung, CPU-Scheduling
4. Welche Scheduling-Strategien gibt es denn da?
   → FIFO, LIFO, SJF (preemptive und non-preemptive) erklärt
5. Kennen Sie auch Round-Robin?
   → ja, ist preemptiv und geht so….erklärt
6. Kennen Sie auch ein Beispiel für ein System mit einem Prozessor und nur einem Programm?
   → Shit, kalt erwischt! Ich kam nicht drauf….Er wollte auf so was hinaus wie ein Programm in einer Waschmaschine, das solange läuft bis es fertig ist und das kein anderes Programm in der Zwischenzeit stören kann.

   Das hat ein bisschen länger gedauert und so hatten wir uns „verquatscht“ – daher hatten wir für nichts anderes mehr Zeit!

Fazit:

Die Atmosphäre war super locker und Prof. Lichter unheimlich freundlich und nett. Die Fragen waren oft Standard, aber teilweise gingen sie doch in die Tiefe oder quer. Doch keine Angst: Prof. Lichter hilft! ;) Ich konnte drei Detailfragen bzw. Fragen, die zu Querthemen gestellt wurden nicht (richtig) beantworten und habe dennoch diese Supernote.
Ich hatte das Gefühl, dass ich schon einen kleinen TK-Bonus hatte, denn er sagte, dass ich den Vorlesungsstoff sehr gut beherrsche, im Detail ein bisschen unsicher wirkte, aber wenn ich die Details interessant finden würde, dann hätte ich ja auch „Informatik studiert und nicht Technik-Kommunikation – und das hat er mit vollstem Verständnis und nicht herablassend gesagt!!!
Alles in allem eine tolle, relaxte Prüfung und Prof. Lichter ist absolut zu empfehlen! ;)