

2. Klausur 12/II (Q1.2) (A)

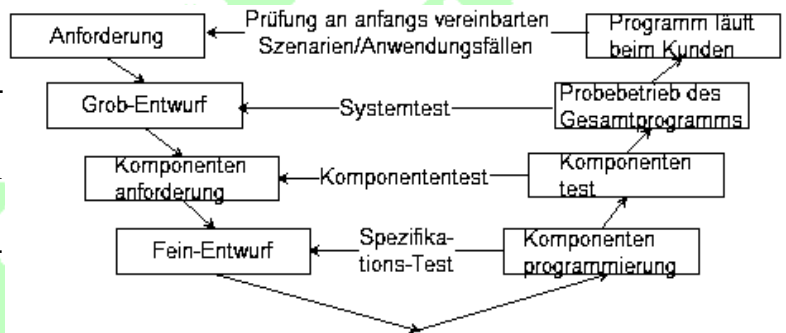
Dauer: 3 Zeitstunden (10:45 bis 13:45 Uhr)

Name: www.r-krell.de

Hilfsmittel: --

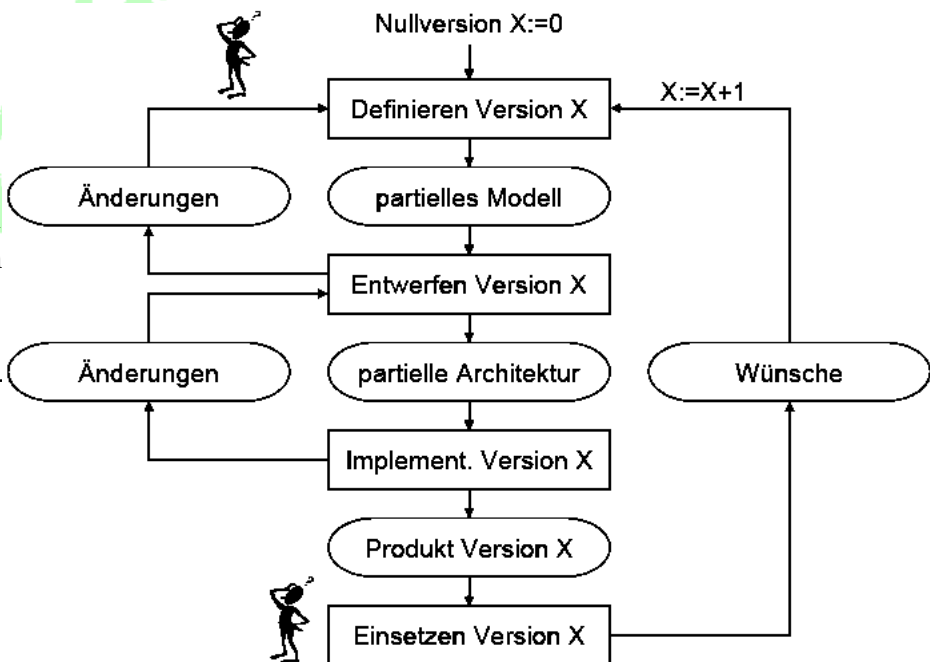
- * *Achte auf sorgfältige Darstellung mit vollständigem, nachvollziehbarem Lösungsweg!* *
- * *Kommentiere deine Programme!* *

- 1) Die Schritte bzw. Phasen von der Planung bis zur Fertigstellung eines Softwareprojekts können durch sogenannte Vorgehensmodelle (auch „Phasenkonzepte“) beschrieben werden.
- a) Im Unterricht waren zwei Modelle vorgestellt worden: Skizziere grob in je einem Diagramm die Phasen für a1) das Wasserfallmodell bzw. für a2) das Spiral-Modell.
- b) Bei den eigenen Projekten wollten viele von euch schon erste Teilerfolge sehen und haben schon frühe, vorläufige Fassungen gestartet und getestet, und erst nach und nach weitere Funktionen hinzugefügt. Erläutere begründet, mit welchem der beiden Modelle aus a) dieses Vorgehen („rapid prototyping“) am besten zu beschreiben ist!



- c) Beim V-Modell sollen die rechts erhaltenen Ergebnisse erst durch Tests mit den Planungen links verglichen werden (ggf. muss rechts nachgebessert werden), bevor man rechts weiter hoch geht.
- c1) Das V-Modell gilt als Weiterentwicklung eines der beiden Modelle aus a). Nenne das passende Modell und erkläre, wo/wie sich das V-Modell davon unterscheidet.
- c2) Bei der Softwareentwicklung sind zwei Fragen wichtig: 1.) Entspricht das Programm den notierten Anforderungen? (Verifikation) sowie 2.) Ist das Programm für den beabsichtigten Einsatzzweck wirklich geeignet? (Validation). Erläutere begründet, ob das V-Modell deiner Meinung nach beide Fragen gleich stark bzw. ausreichend berücksichtigt.

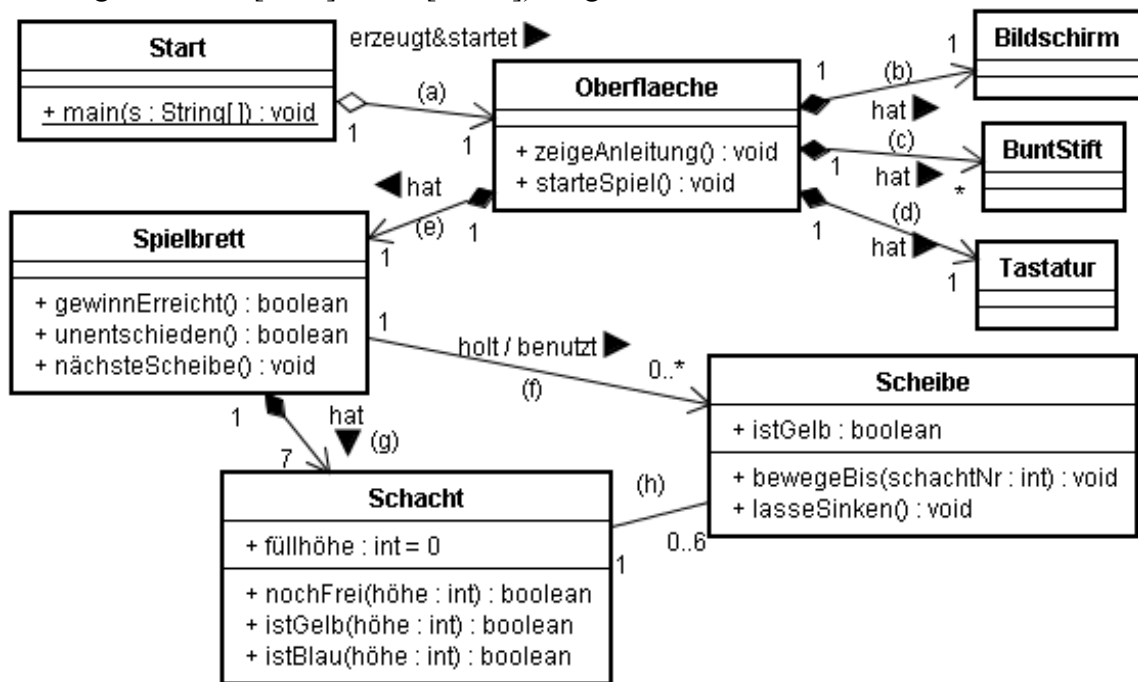
- d) Nach dem evolutionären Modell erhält der Kunde schon bald die Produkt-Version 0 der bestellten Software zum Einsatz und Test und kann diese auf Wunsch (und gegen Bezahlung) weiter verbessern lassen.



<http://www3.informatik.uni-erlangen.de/Lehre/UML-Seminar/SS2002/StateOfTheArt.pdf>

- d1) Erläutere, welchem der beiden Vorgehensmodelle aus a) dieses evolutionäre Modell eher ähnelt. Und wo sind trotzdem Unterschiede? Gibt's hier Validation?
- d2) Suche begründet je ein Beispiel, für welche Kunden bzw. für welche Software-Aufträge sich das evolutionäre Modell gut bzw. gar nicht eignet.

- 2) Bei einem 4-Gewinnt-Spiel wird abwechselnd eine blaue oder gelbe Scheibe in einen von sieben Schächten geworfen, wo sie soweit wie möglich nach unten sinkt. Maximal passen jeweils 6 Scheiben übereinander (Höhen 0 [unten] bis 5 [oben]). Der Schacht, in den die nächste Scheibe soll, wird per Tastatur (durch Eingabe von '0' [links] bis '6' [rechts]) ausgewählt.



- a) Erkläre/Begründe für jeden(!) der acht Verbindungsstrichen (a) bis (h) jeweils in ein bis zwei Sätzen die im Diagramm notierten Multiplizitäten (oder korrigiere begründet).
- b) Manche Verbindungen sind einfache Assoziationen, andere Aggregationen oder Kompositionen. Ordne (a) bis (h) den drei Typen zu und begründe jeweils für eine einzige Verbindung von jedem Typ, warum hier gerade dieser Typ gewählt wurde.
- c) Die Klassen wurden im Diagramm unvollständig notiert. So fehlen z.B. in der *Oberflaeche* die Attribute (Variablen), obwohl es offensichtlich mindestens 4 geben muss. Welche? (Typ?)
- d) Schreibe die Klasse *Spielbrett* mit leeren Methodenrahmen in Java. Definiere und erzeuge aber geeignete Attribute für die Verwaltung der 7 Schächte und erläutere begründet, warum du Reihung, Keller, Schlange und/oder Liste genommen hast.
- e) Beschreibe in Deutsch, wozu die Variable (= das Attribut) bzw. alle Methoden (= die Operationen und Funktionen) der Klasse *Schacht* wohl gedacht sind. Beachte dabei, dass allen Methoden jeweils als *höhe* eine Zahl zwischen 0 [unten] und 5 [ganz oben] als Parameter übergeben werden muss – und *nochFrei(füllhöhe-1)* das Ergebnis *false* liefert! Erläutere jeweils den Sinn, die Sichtbarkeit sowie Anfangs- oder Rückgabewert.
- 3) Der Sportverein „SV Oberbilk“ hat verschieden Abteilungen. Die Mitgliederverwaltung der Fußball-Abteilung soll demnächst per Java-Programm erfolgen. Berücksichtige bei der Planung:
- * Zur Fußball-Abt. gehören Spieler und Trainer (wobei es beim SV Oberbilk keine Spielertrainer gibt, die beides in einer Person sind). Für beide Personengruppen sollen Name, Vorname und Geburtsdatum gespeichert werden.
 - * Bei den Spielern werden zusätzlich eine Stamm- und eine Alternativposition festgehalten. Zudem wird der momentane Fitnesslevel durch eine Zahl (1, 2, 3 oder 4) ausgedrückt.
 - * Der Verein unterhält mehrere Fußball-Mannschaften. Jede Mannschaft hat eine Bezeichnung und gehört zu einer bestimmten Liga. Jede Mannschaft bestreitet Spiele.
 - * Zu jedem Spiel soll der Name der Gegners (etwa „FC Tannenhof“), das Datum des Spiels und das

Ergebnis (in Form der erzielten Tore und Gegentore) festgehalten werden.

- * Während eines Spiels kommen 11 bis 14 Spieler des Vereins zum Einsatz. Es soll nachvollziehbar sein, welche Spieler das sind bzw. in einem bestimmten Spiel waren.
- * Spieler sind einer Mannschaft zugeordnet, in der sie normalerweise spielen. Jede Mannschaft hat mindestens 11 solcher Spieler.
- * Trainer haben entweder eine Lizenz oder nicht. Jeder Trainer hat eine bestimmte Zahl von Fortbildungen besucht. Trainer sind in der Regel Mannschaften zugeordnet; dabei kann ein Trainer auch mehreren Mannschaften zugeordnet sein. Es gibt aber auch Trainer (z.B. Fitness-Experten), die keiner Mannschaft zugeordnet sind. Jede Mannschaft hat mindestens einen, maximal drei Trainer.
- * Bei Spielen gibt es immer genau einen Trainer, der die Mannschaft in diesem Spiel betreut.
- * Es sollen auch Trainingseinheiten verwaltet werden. Zu jeder Trainingseinheit sollen die Mannschaft, das Datum und der Belastungsgrad (leicht - mittel - schwer) erfasst werden.

- a) Erstelle auf der Grundlage dieser Informationen ein UML-Klassendiagramm. Berücksichtige dabei Klassen, Attribute (nur soweit sie obiger Beschreibung zu entnehmen sind) sowie die Beziehungen zwischen den Klassen. Gib bei allen Beziehungen auch die Multiplizitäten an!
[Hinweis: Methoden sollen hingegen nicht angegeben werden. Attribute, die sich aus der Umsetzung von angegebenen Beziehungen ergeben würden, brauchen nicht berücksichtigt werden. Verwaltungsklassen können ebenfalls weggelassen werden].
- b) Erkläre kurz und allgemein die folgenden Begriffe: Objekt, Klasse, Attribut, Methode.

- 4 Die „Lessing“-Lehrerinnen und -Lehrer könnten in einer (My-)SQL-Datenbank gespeichert sein. Da jeder Lehrer ein bis drei Fächer unterrichten kann, wären verschiedene Tabellen möglich (die in Wirklichkeit immer alle Lehrer enthalten, von denen hier nur wenige beispielhaft gezeigt sind):

VersionA	
Lehrer	Fächer
El Kasmi	M, IF
Kohn	D, GL

VersionB			
Lehrer	Fach1	Fach2	Fach3
El Kasmi	M	IF	(null)
Kohn	GL	D	(null)

VersionC	
Lehrer	Fach
El Kasmi	IF
El Kasmi	M
Kohn	GL
Krell	M
Kohn	D
Krell	IF
Krell	PH

VersionD	
Lehrer	Fach
ELK	M
ELK	IF
KN	GL
KN	D
wobei in einer Extratablelle jedes Lehrer-Kürzel einmal erklärt wird	

- a) Betrachte A bis D:
- a1) Prüfe, ob Version D trotz nötiger Extratablelle (in der einmal ELK | El Kasmi, KN | Kohn, .. steht) platzsparender und weniger fehleranfällig als C ist, weil in der Lehrer/Fach-Tabelle die z.T. langen Lehrernamen nicht mehrfach ausgeschrieben werden müssen.
 - a2) Erläutere begründet, wie gut jede der Versionen A bis D geeignet ist, wenn man z.B. alle Lehrer mit einem bestimmten Fach (z.B. alle GL-Lehrer) sucht.
 - a3) Erläutere begründet, wie gut in jeder der Versionen A .. D festgestellt werden kann, welche Fächer ein bestimmter Lehrer unterrichten kann (z.B. alle Fächer von Untermann?)
 - a4) Entscheide, ob es eine Version gibt, die insgesamt am besten bzw. ein guter Kompromiss für beide Sorten von Anfragen (a2) & a3)) ist.
- b) Schreibe die 4 (My-)SQL-Befehle, um (1) eine neue Datenbank namens Schule anzulegen, (2) diese

Datenbank im Folgenden auch zu benutzen, (3) die TabelleB anzulegen und (4) in die TabelleB die erste Beispiel-Zeile (wie oben gezeigt) einzufüllen.

5) Ausgeführt werden folgende drei SQL-Anweisungen (innerhalb einer Datenbank *Luxusmarkt*, die auch benutzt wird):

1. create table *Katalog* (*BestellNr* int, *Bezeichnung* varchar(30), *Warengruppe* varchar(20), *Preis* decimal(7,2));
 2. insert into *Katalog* (*BestellNr*, *Bezeichnung*, *Preis*, *Warengruppe*) values (12, "Maus, vergoldet", 279.95, "Computerzubehör"), (13, "Kuli mit Platin-Clip", 120.00, "Schreibgeräte"), (14, "Tastatur mit Brillanten", 449.95, "Computerzubehör"), (16, "Ringbuch mit Pelzrand", 89.75, "Papierwaren");
 3. select * from *Katalog*;
- a) Zeichne/schreibe die Tabelle, die schließlich durch den Befehl 3. dargestellt wird!
- b) Notiere die Antworttabellen zu folgenden Befehlen:
- b1) select *Preis*, *BestellNr* from *Katalog*;
 - b2) select *BestellNr* from *Katalog* where *Preis* < 250;
 - b3) select *BestellNr* from *Katalog* where *Preis* > 300 and *Warengruppe* = "Computerzubehör";
 - b4) select *Preis* from *Katalog* where *BestellNr* > 13;
 - b5) select *BestellNr*, *Preis* from *Katalog* where *Warengruppe* = "Computerzubehör" or *Preis* > 100;
 - b6) select *Warengruppe*, *Preis* from *Katalog* where *BestellNr* between 12 and 14;
- c) Schreibe selbst select-Befehle, die Folgendes anzeigen:
- c1) die Bezeichnungen und Warengruppen aller Artikel zwischen 100 und 400 Euro
 - c2) Bezeichnungen und Preise aller Computerzubehör-Artikel mit Bestellnummer unter 14
 - c3) Preise, Bezeichnungen und Warengruppen aller Artikel, bei denen das Wort „mit“ irgendwo in der Bezeichnung vorkommt.
- d) Dem Katalog soll ein weiterer Papierwaren-Artikel mit der Bestell-Nummer 21 hinzugefügt werden: ein Heft mit Ledereinband zum Preis von schlappen 79.95 €. Schreibe den passenden SQL-Befehl!
- e) Dem Katalog soll eine Spalte *Lagerbestand* hinzugefügt werden, wo in jeder Zeile eingetragen werden kann, wie viele Exemplare des jeweiligen Artikels am Lager vorrätig sind. Schreibe den passenden SQL-Befehl!
- f) Wenn man demnächst auch *Crystal-Clear-Sound Lautsprecher mit Swarovski-Kristallen* aufnehmen will, reichen die 30 Buchstaben bei der Bezeichnung nicht. Ändere die maximale Buchstabenanzahl für die Bezeichnungen im Katalog auf bis zu 70 Schriftzeichen bzw. schreibe den passenden SQL-Befehl!