Di., 9.3.2010 Informatik 12 M (Kr)

1. Klausur 12/II

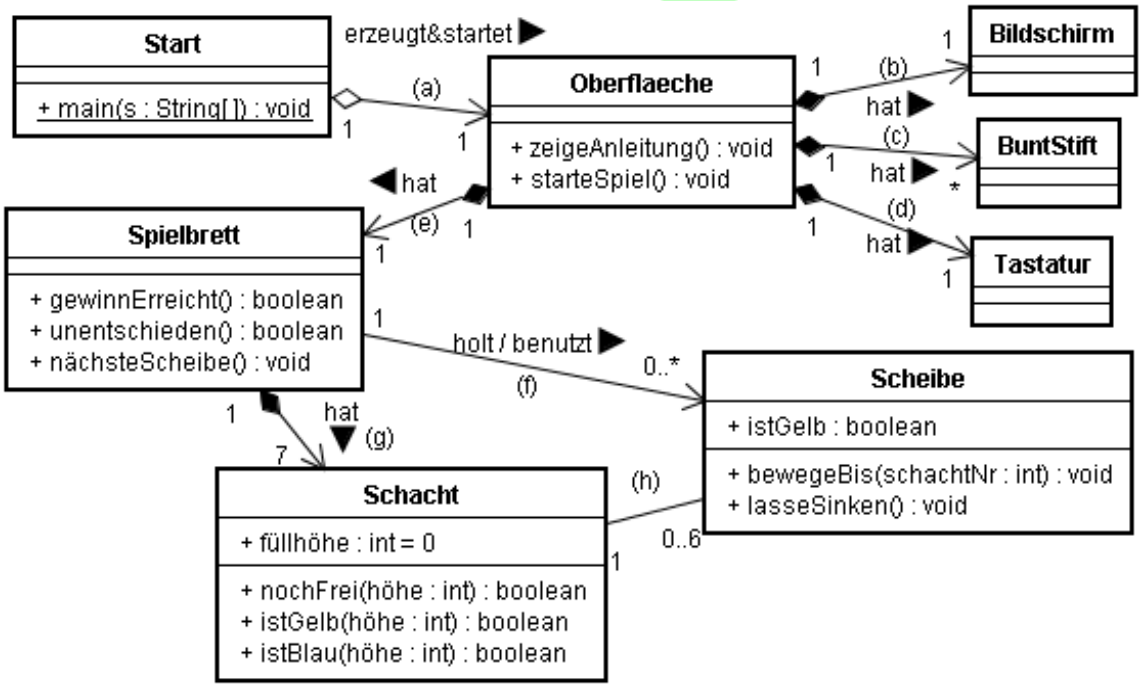
Dauer: 4 Schulstunden (Abgabe 11:10 Uhr) Name: ___www.r-krell.de__

Hilfsmittel: --

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

1 Spiel

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 bis 5). Der Schacht, in den die nächste Scheibe soll, wird per Tastatur (durch Eingabe von '0' bis '6') ausgewählt.



a) Erkläre/Begründe für jeden(!) der acht Verbindungsstrichen (a) bis (h) jeweils in ein bis zwei Sätzen die im Diagramm notierten Multiplizitäten.

b) Manche Verbindungen sind einfache Assoziationen, andere Aggregationen oder Kompositionen. Ordne (a) bis (h) den drei Typen zu und begründe jeweils für eine Verbindung von jedem Typ, warum gerade dieser Typ gewählt wurde.

c) Die Klassen wurden im Diagramm unvollständig notiert. So fehlen z.B. in der *Oberflaeche* die globalen Variablen, obwohl es offensichtlich mindestens 4 geben muss. Welche? (Typ?)

d) Schreibe die Klasse *Spielbrett* mit leeren Methodenrahmen in Java. Definiere und erzeuge aber eine geeignete globale Variable für die Verwaltung der 7 Schächte und erläutere begründet, warum du Reihung, Keller, Schlange oder Baum genommen bzw. nicht 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.

f) Bevor ein neuer Stein im Spiel bereitgestellt wird, fragt *starteSpiel* aus der Oberfläche erst beim Spielbrett-Objekt nach, ob nicht schon ein *gewinnErreicht* wurde. Gibt es noch keinen Gewinner, wird gefragt, ob das Spiel dann vielleicht durch *unentschieden* zu Ende ist. Beschreibe zunächst auf Grund deiner Kenntnis des Spiels in Deutsch, wann 4-Gewinnt unentschieden endet und schreibe dann für die Klasse *Spielbrett* (wie in d)) die Methode *unentschieden* möglichst in Java (oder als Struktogramm).

2 Schule

a) Bei Klausurversäumnis an der fiktiven XY-Schule gilt:

Wenn ein Schüler am Klausurtag erkrankt ist, muss er bis 9:00 Uhr im Sekretariat anrufen. Die Sekretärin fragt dann nach Name, Klasse und Fach und bereitet ein Formular vor. Ist der Schüler wieder gesund, kann er das vorbereitete Formular im Sekretariat abholen und zusammen mit dem Attest dem Fachlehrer vorlegen. Dieser bestätigt auf dem Formular, ob Formular und Attest rechtzeitig (=binnen 3 Tagen nach Wiedererscheinen) gezeigt wurden oder nicht. Der Schüler übergibt das Formular mit der positiven Bestätigung dem Oberstufenkoordinator. Dieser prüft, ob das Formular vollständig ausgefüllt wurde und binnen einer Woche nach Wiedererscheinen abgeben wurde. Ist alles okay, kommt der Schüler auf die Nachschreibeliste, sonst gilt die Klausur als unentschuldigt/ungenügend.

a1) Zeichne ein eEPK für den beschriebenen „Geschäftsprozess“! Trenne dabei möglichst in vier Sichten (Spalten), nämlich Daten-, Steuerungs-, Funktions- und Organisations-Sicht.

a2) Im eEPK aus a) wurden XOR- oder ODER(∨)-Konnektoren verwendet, um den Kontrollfluss zu verzweigen. Erläutere begründet, ob eine solche Verzweigung sowohl nach Ereignissen als auch nach Funktionen (=Handlungen/Prozessen) erlaubt ist. Erläutere außerdem (ggf. mit eignem Beispiel/Gegenbeispiel), ob eine Verzweigung mit UND (∧)sowohl nach Ereignissen wie auch nach Funktionen erlaubt wäre.

a3) Auch bei der Zusammenführung von Kontrollflüssen müssen die Konnektoren UND, ODER oder XOR verwendet werden. Erkläre, ob/welche Zusammenführungen jeweils vor Ereignissen und/oder vor Funktionen erlaubt sind.

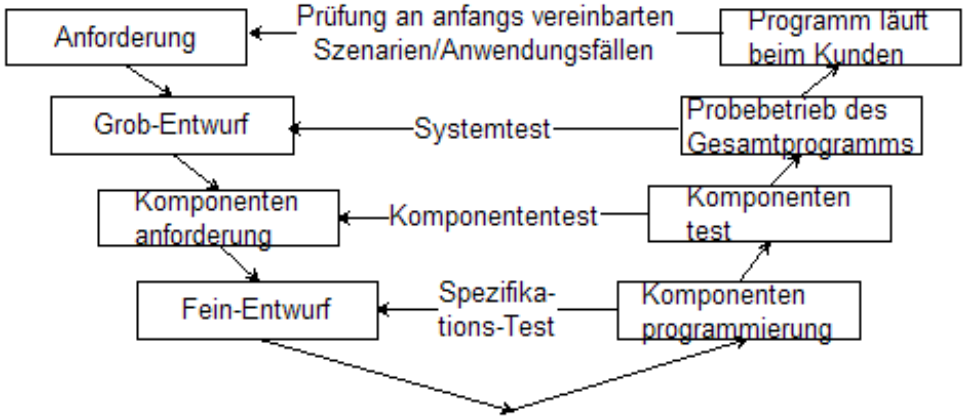
b) Für das Erzeugen der Nachschreibeliste soll ein Programm entworfen werden. Dabei gilt:

* Die Nachschreibeliste enthält das Datum, an dem sie erstellt wurde, sowie viele Zeilen. Zeilen können in die Nachschreibeliste aufgenommen oder wieder gestrichen werden.
* Ein Datum besteht aus Tag, Monat und Jahr.
* Jede Zeile enthält den Vor- und Nachnamen eines Schülers, seine Klasse sowie das Fach und den Lehrer der versäumten Klausur, wobei Fach und Lehrer abgekürzt werden (z.B. IF oder KR).
* Die Daten eines Schülers sollen aus der vorhanden Schülerverwaltung übernommen werden ebenso wie die Daten eines Lehrers aus der vorhandenen Lehrerverwaltung kommen.
* Dabei sind Schüler und Lehrer zunächst Personen mit Vor- und Zuname und Geburtsdatum; Schüler haben außerdem eine Klasse und Lehrer sind Personen, die zusätzlich ein Kürzel haben. Es werden nur Lehrer oder Schüler verwaltet, keine anderen Personen.
* Schüler schreiben oder versäumen Klausuren, Lehrer stellen Klausuren oder korrigieren .

Zeichne ein UML-Diagramm für das Nachschreibelisten-Programm (einschl. der Klassen der vorhandenen Schüler-/Lehrer-Verwaltungen), trage alle im Text erwähnten Attribute und Methoden (mit Sichtbarkeiten, Typ und Parametern) ein und zeichne Vererbungspfeile, Assoziationen, Aggregationen und Kompositionen mit Benennung und – soweit möglich/nötig – mit Multiplizitäten (Vielfachheiten).
Erläutere außerdem in einem zusätzlichen Text für einen einzigen von dir gewählten Assoziations-, Aggregations- oder Kompositionspfeil des Diagramms beispielhaft die Bedeutung der gezeichneten Pfeilrichtung (Navigationsrichtung).

3 Vorgehensmodelle

Für (große) Softwareprojekte, die der deutsche Staat in Auftrag gibt, wird die Verwendung des (hier etwas vereinfachten) V-Modells vorgeschrieben:



Links steigt man zum Entwurf ab; rechts steigt man zur Fertigstellung der Software auf. Auf jeder Ebene kann und soll überprüft werden, ob der rechts erstellte Entwurf bzw. das rechts entwickelte Programm den links festgeschriebenen Anforderungen genügt (waagrechte Pfeile), sonst muss rechts in der Ebene verbessert werden.

a) Das V-Modell gilt als Weiterentwicklung eines der beiden im Unterricht angesprochenen Vorgehensmodelle. Zeichne und beschreibe das passende Modell aus dem Unterricht und erläutere, wo/wie sich das V-Modell davon unterscheidet.

b) 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.

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

   c1) Erläutere, welchem der beiden Vorgehensmodelle aus dem Unterrichts dieses evolutionäre Modell eher ähnelt. Und wo sind trotzdem Unterschiede?

   c2) Suche begründet je ein Beispiel, für welche Kunden bzw. für welche Software-Aufträge sich das evolutionäre Modell gut bzw. gar nicht eignet.

In der Klausur wurde den Schülerinnen und Schülern hier das Bild eines Vorgehensmodells vorgelegt, das dem im Internet gefundenen Vorlesungsskript einer Fachhochschule entnommen war.
Um Copyrightverstöße zu vermeiden, wird auf die Veröffentlichung des fremden Bildes auf meiner Webseite verzichtet.

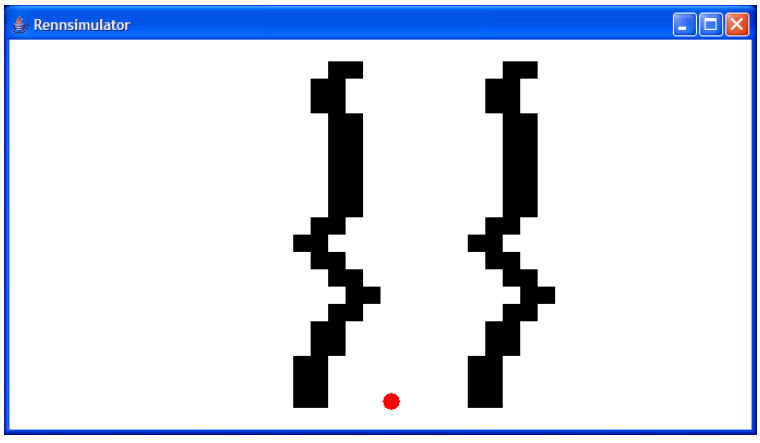
4 Urlaub

Ein Reisebüro hat sich auf den Verkauf von Last-Minute-Reisen spezialisiert. Dazu werden täglich reduzierte Angebote verschiedener Reiseveranstalter per e-Mail-Newsletter an die Newsletter-Abonnenten geschickt. Am Ende jedes Newsletters steht die Telefonnummer des Reisebüros, damit man dort buchen kann. Bisher verläuft die Telefonbuchung wie folgt:

> Ruft ein Interessent an, wird er nach der Nummer der zu buchenden Reise im Newsletter gefragt. Kennt er die Nummer nicht oder hat sich noch nicht entschieden, wird er nach seiner e-Mail-Adresse gefragt und das Telefonat abgebrochen. Der Reisebüro-Mitarbeiter schickt dann den aktuellen Newsletter an den Anrufer (damit sich dieser vorm nächsten Anruf in Ruhe eine Reise aussuchen kann) und der Vorgang ist beendet. Nennt der Anrufer eine Reise-Nummer, wird er nach der Zahl der reisenden Personen gefragt und der Mitarbeiter guckt auf der Webseite des Veranstalters nach, ob noch genügend freie Plätze da sind. Sind nicht genug freie Plätze da, wird der Anrufer gebeten, im Newsletter nach einer anderen Reise zu suchen und das Gespräch beendet. Gibt es freie Plätze, reserviert der Reisebüromitarbeiter diese beim Veranstalter, nennt dem Kunden den Gesamtpreis und fragt den Kunden nach Name, Adresse und Kreditkartennummer. Ist dem Kunden die Reise zu teuer, hat der Kunde keine Kreditkarte oder gelingt der Versuch des Mitarbeiters nicht, die Karte online bei der Bank mit dem Reisepreis zu belasten, so gibt der Mitarbeiter die reservierten Plätze wieder frei und beendet das Gespräch mit dem Hinweis, dass die Zahlung nur per gültiger Kreditkarte möglich sei. Gelingt die Bezahlung, so wird die Reise gebucht und das Gespräch beendet.

a) Zeichne ein eEPK (die 3-Sichten-Darstellung reicht).
b) In letzter Zeit häufen sich die Beschwerden von Kunden über die Art, wie sie am Telefon abgefertigt werden. Auch die Mitarbeiter des Reisebüros beklagen sich, dass viele Telefonate vorzeitig enden und oft schon reservierte Reisen wieder freigegeben werden müssen, weil keine Buchung zu Stande kommt.
   b1) Benenne jeweils die erkennbaren Ursachen für die Unzufriedenheit mancher Kunden ebenso wie die Ursachen für den Frust der Mitarbeiter.
   b2) Verbessere den beschriebenen Geschäftsprozess, sodass künftig Kunden und Mitarbeiter zufriedener sind und beschreibe die Verbesserungen im Text [und als eEPK].

5 Spiel 2



Bei einer einfachen, mit stiftUndCo programmierten Rennsimulation kann man mit den Pfeiltasten ← und → das Auto (Kreis) nach links oder rechts lenken, während einem die Straße entgegen kommt (d.h. pro Zeittakt rücken alle Zeilen um eins nach unten. Dabei verschwindet die untere Zeile, während oben eine neue Zeile dazu kommt). Das Spiel endet, wenn das Auto (das immer am unteren Bildrand bleibt) an den Straßenrand stößt (d.h. wenn es an einer Stelle steht, wo beim Runterrollen der Straße im nächsten Zeittakt ein Randstein hin kommt).
Zeichne ein UML-Klassendiagramm mit den Klassen *Oberflaeche, Bildschirm, BuntStift, Tastatur*, *Strasse* und *Auto*. Zeichne und beschrifte Assoziationen/Aggregationen/Kompositionen und notiere im Diagramm für *Strasse* und *Auto* jeweils den Namen der wichtigsten Methode. Beschreibe die Funktion dieser beiden Methoden durch zusätzlichen Text außerhalb des Diagramms.