EI Info J2 2011-12	*INFORMATIK* 1. Klausur (05.12.11) - Lösung	

1. Aufgabe **(2 Punkte)**

Erläutere, was objektorientierte Programmierung ist.

Bei der objektorientierten Programmierung wird eine allgemeine Problemstellung unterteilt (Objekte). Diese Unterteilungen werden getrennt behandelt (Klassen mit Methoden und einem Konstruktor) und können aufeinander zugreifen (get/set).

Dadurch wird erreicht, dass ähnliche Modelle (Pony, Zebra) in einer einzigen Klasse (Pferd) stehen und dort einfache Spezialisierungen sind.

So richtig einfach oder eindeutig ist A1 aber nicht zu beantworten…

2. Aufgabe **(12 Punkte)**

Gegeben ist folgender Programmcode:

```
public class BesondereZahlen
{
        private int [] zahl;

        public BesondereZahlen()
        {
                zahl = new int [10];

                for (int i=1; i<10; i=i++) {zahl[i] =3*i;}
        }

        public int EintragAusgeben(int welcheStelle)
        {
                return zahl[welcheStelle];
        }
}
```

a) Erläutere anhand des Codes den grundlegenden Aufbau eines Java-Programmes.

Ggf. Import wichtiger Bibliotheken (hier keine), Definition einer Klasse. In dieser dann Def. von Variablen, ein Konstruktor und Eigenschaften (=Methoden) dieser Klasse (die dann auch alle Unterklassen können).

b) Was unterscheidet „private int [] zahl" von einem normalen „private int zahl"?

Damit wird ein Array definiert, also eine Liste, in der wiederum INT-Zahlen stecken.

c) Ändere die Methode „EintragAusgeben()" so um, dass sie keinen Wert mehr zurückgibt, sondern den Wert des Listeneintrags ausdruckt (Pseudocode erlaubt).

```
public void EintragAusgeben(int welcheStelle)
        {
                println(“”+zahl[welcheStelle]);
        }
```

Das ist halber Pseudocode. Wichtig ist aber das VOID! Denn ihr wollt keine Rückgabe mehr.

d) Erläutere anhand der for-Schleife, was deren Vorteile in der Programmierung sind.

For-Schleifen übernehmen das Ausführen einer mehrfach wiederholten Routine! Das spart immense Programmierarbeit und erst so wird ein PC wirklich „mächtiger" für uns, da er solche Iterationen sehr schnell ausführt, wir sie ihm aber sehr kompakt „beibringen" können, sprich, nicht viel zu tippen haben…

e) Welchen Wert hat „zahl[1]" und welchen Wert hat „zahl[10]"?

Hier ist i=1 bzw. i=10. i=1 ist die ZWEITE Stelle im Array (der ist 10 lang und läuft von zahl[0] bis zahl[9]. Es ist zahl[0]=0 (weil nicht in der Schleife beschrieben, das startet ja bei i=1) und zahl[1]=3. zahl[10] ist gar nicht definiert, da der Array mit zahl[9] endet…

f) Wie müsste man das Programm modifizieren, damit der Benutzer die Listenlänge vorgeben kann und diese vollständig mit den ersten geraden Zahlen beschrieben wird (Pseudocode erlaubt)?

Jetzt müssen wir den Konstruktor ändern:

```
public BesondereZahlen(int wieLang)
{
        zahl = new int [wieLang];

        for (int i=0; i<wieLang; i=i++) {zahl[i] =2*i;}
}
```

Wichtige Änderungen: „int wie Lang" verlangt beim Ausführen des Konstruktors eine Eingabe der Listenlänge. Die wird dann auch weiterverwendet (ersetzt die statische 10). In der For-Schleife korrigieren wir noch i=0, damit ab 0 beschrieben wird und 3*i wird durch 2*i ersetzt. Wobei wir dann 0 als gerade Zahl ansehen.

3. Aufgabe **(6 Punkte)**

Die Klasse **WuerfelSim** eines Java-Programmes sieht so aus:

```
import java.util.Random;

public class WuerfelSim
{
        private Random zufallsgenerator;

        public WuerfelSim()
        {
                zufallsgenerator = new Random();
        }

        public int Wuerfel(int max)
        {
                int ergebnis = zufallsgenerator.nextInt(max) + 1;
                return ergebnis;
        }
}
```

a) Welchen Sinn hat die erste Zeile, also der Ausdruck „import java.util.Random"?

Hier wird eine Bibliothek importiert, in der bereits die Klasse „Random" bekannt ist und auf die jetzt zugegriffen werden kann.

b) Der Befehl „Random.nextInt(6)" erzeugt ganze („Zufalls")Zahlen von 0 bis 5. Durch „+1" erhält man die üblichen Zahlen von 1 bis 6. Wieso kann das Programm „WuerfelSim" mehr als ein gewöhnlicher sechsseitiger Würfel?

Man kann die Zahl 6 ersetzen! So kann man beliebigseitige Würfel erzeugen!

c) Schreibe eine Methode ZinkWuerfel, bei dem die Zahlen 1 bis 5 zu je 1/10 auftreten, die Zahl 6 jedoch zu 5/10. *Tipp: Nutze einen Array!*

Mit dem Tipp würde man einen Array anlegen, aus dem zufällig gezogen wird. Er hätte 10 Einträge: 1,2,3,4,5,6,6,6,6,6. Dann wäre die Aufgabe erfüllt. Einfacher ist aber, die 6 durch 10 zu ersetzen und mit „if ergebnis>=6" den Befehl „return 6" auszuführen und „else" das ergebnis nicht zu verändern; „return ergebnis". Das ist alles!

4. Aufgabe **(4 Punkte)**

Im Unterricht haben wir die Klassen **Uhrenanzeige** und **Nummernanzeige** besprochen, die beide zum bekannten Projekt Zeitanzeige gehören. In der Uhrenanzeige werden am Anfang folgende Befehle gegeben:

```
private Nummernanzeige stunden;
private Nummernanzeige minuten;
```

a) Was passiert mit diesen beiden Befehlen?

Es werden eine Nummernanzeige „stunden" und eine mit dem Namen „minuten" erstellt. Im Programm Uhrenanzeige kann man mit diesen beiden arbeiten (siehe b)!).

b) In der Methode **setzeUhrzeit** taucht die Anweisung **„stunden.SetzeWert(stunde)"** auf. Wieso wird diese Schreibweise verwendet und was geschieht hier?

Genau wie in a) gesagt, arbeiten wir jetzt mit den Nummernanzeigen. Hier wird auf die Anzeige „stunden" zugegriffen und der Wert „stunde" in der Anzeige neu gesetzt. Durch das „stunden." weiß Java, dass wir darauf zugreifen wollen.