

# Mögliche Prüfungsfragen

## Informatik II - D-BAUG

Felix Friedrich, Lars Widmer

13. Juni 2014

### Vorwort

Diese Liste soll bei der Vorbereitung auf die kommende Prüfung helfen. Eine aktualisierte Version werden wir zu Beginn der unterrichtsfreien Zeit auf der Vorlesungswebsite veröffentlichen. Es ist möglich dass teils Fragen aus dieser Liste in ergänzter, kombinierter oder abgewandelter Form Eingang in die Leistungskontrolle finden. Die Liste erhebt jedoch keinen Anspruch auf Vollständigkeit. Für eine zweistündige Probeprüfung wäre diese Liste zu lang, zumal die Lösung mancher Aufgaben viel (hand)schriftliche Arbeit erfordern. Voraussichtlich wird die Klausur zu Informatik II aus maximal **10 Aufgaben** bestehen.

Die Leistungskontrolle findet am **Montag 11. August** statt. Ort und Zeit stehen noch nicht fest. Informatik I (*Teil 1*) und Informatik II (*Teil 2*) werden am gleichen Tag geprüft. Total dauert die Klausur für beide Teile 4 Stunden. Die Prüfung für Informatik II (*Teil 2*) alleine dauert **2 Stunden**. Für Teil 1 dürfen Sie eine selbstverfasste, handgeschriebene Zusammenfassung auf 2 A4-Seiten (*resp. 1 Blatt doppelseitig*) verwenden. Für Teil 2 dürfen Sie selbstverfasste Notizen auf **8 A4-Seiten** (*resp. 4 Blätter doppelseitig*) verwenden.

Kopieren Sie bitte keine Zusammenfassungen (*Spickzettel*). Die sehen zwar meistens schön aus, helfen Ihnen aber **nicht** viel weiter. Legen Sie Wert darauf, Ihren Spick selbst zu verfassen. Jeder Studentin und jedem Student sind andere Dinge unklar. Es hat sich bewährt, direkt beim Durchgehen des Stoffs, das auf den Spick zu schreiben, was Ihnen (noch) nicht in den Kopf will.

Die Prüfung in Informatik II wird zweisprachig abgefasst. Das heisst auf den Blättern steht jeweils die gleiche Aufgabenstellung, sowohl in deutsch, als auch englisch. Was Sie dann lesen wollen, dürfen Sie entscheiden. Bisweilen hilft es, bei Verständnisproblemen, beide Sprachen zu lesen. Die Aufgabenstellungen in den beiden Sprachen werden sich in Schriftbild und Farbe unterscheiden.

Gerne halten wir für alle Interessierten ein paar Wochen vor der Leistungskontrolle eine Fragestunde ab. Ort und Zeit werden so schnell wie möglich auf der Website publiziert. Sie können uns bis eine Woche vor der Fragestunde Ihre Fragen mailen und wir versuchen die dann im Plenum zu beantworten. Weiter können wir bei dieser Gelegenheit mit Ihnen die Lösungen zu nachfolgenden möglichen Prüfungsfragen besprechen. Allerdings wird es keine Vorlesung im Sinn, dass wir die Lösung einfach präsentieren. Eher werden wir gemeinsam die Aufgaben durchgehen und jede Studentin und jeder Student beantwortet eine Teilfrage.

## 1 Programmverständnis Schleifen

Bestimmen Sie die Ausgabe von folgendem Programm.

```
1 public static void main(String[] args) {
2     int x = 7, y = 1;
3     while (x > -3) {
4         System.out.println("x="+x);
5         x = x - 2 * y;
6         y++;
7     }
8 }
```

## 2 Programmverständnis Rekursion

Bestimmen Sie die Ausgabe von folgendem Programm.

```
1 public static void tmp(int n) {
2     System.out.println(n);
3     if (n > 3) {
4         tmp(n-3);
5     }
6 }
7
8 public static void main(String[] args) {
9     tmp(10);
10 }
```

## 3 Programmverständnis Referenzen und Objekte

Bestimmen Sie die Ausgabe von folgendem Programm.

```
1 public class Tiny {
2     public String a;
3 }
4
5 public static void main(String[] args) {
6     Tiny t1, t2;
7     t1 = new Tiny();
8     t2 = new Tiny();
9     t1.a = "Hund";
10    t2 = t1;
11    t2.a = "Katze";
12    System.out.print(t1.a);
13 }
```

## 4 Fehlersuche Vererbung

Im folgenden Code, haben sich zwei Fehler eingeschlichen. Finden und beheben Sie diese.

**Tipp:** Es handelt nicht um Syntaxfehler.

```

1 public class Hello {
2     protected int x;
3     public void outro() {
4         System.out.println("See_Ya");
5     }
6     private void test() {
7         System.out.println("Hello");
8     }
9 }
10
11 public class ByeBye extends Hello {
12     public void intro() {
13         System.out.println(x);
14         test();
15     }
16 }
17
18 public class Test {
19     public static void main(String[] args) {
20         Hello h = new ByeBye();
21         h.intro();
22         h.outro();
23     }
24 }

```

## 5 Arrays

In einer Anwendung werden extrem viele Sinus-Werte berechnet. Eingangswerte sind immer 0 bis  $2\pi$  (**double**). Die Ausgangswerte sind  $-1$  bis  $+1$  (**double**). Das Programm ist wegen den aufwändigen Sinus-Berechnungen sehr langsam.

Jetzt wurde herausgefunden, dass 1% Ungenauigkeit in dieser Anwendung keinerlei Problem darstellt. Deswegen wollen wir nun 100 Sinus-Werte für den Bereich von 0 bis  $2\pi$  vorberechnen. Die Eingangswerte werden dazu mit 100 multipliziert und als **int**-Zahlen verwendet. Anstelle einer Neuberechnung können die gesuchten Sinuswerte fortan aus der Tabelle gelesen werden.

- a) Schreiben Sie den Code, um das Array zu erstellen und mit den entsprechenden Sinus-Werten zu füllen. Verwenden Sie `Math.sin(double a)`, um die Sinuswerte zu berechnen.
- b) Sehen Sie eine Möglichkeit mit einer kleineren Tabelle die gleiche Genauigkeit zu erreichen?

## 6 Programmverständnis und Tools

Betrachten Sie folgende zwei Java-Programme

```

1 public class Aus {
2     public static void main(String[] args) {
3         int len = 0;
4         if (args.length > 0)
5             len = Integer.parseInt(args[0]);
6         System.out.print(len);

```

```

7   for (int i = 0; i < len; ++i)
8       System.out.print("_"+ i*i);
9   } }

1  import java.util.Scanner;
2  public class Ein {
3  public static void main(String[] args) {
4      Scanner scanner = new Scanner(System.in);
5      if (scanner.hasNextInt()) {
6          int len = scanner.nextInt();
7          int sum = 0;
8          while (len > 0 && scanner.hasNextInt())
9              sum += scanner.nextInt();
10         System.out.println("sum=="+sum);
11     }
12     scanner.close();
13 } }

```

Welche Ausgabe erwarten Sie bei den folgenden zwei Programmaufrufen

```

java Aus 3
java Aus 4 | java Ein 3

```

## 7 Zufallszahlen

Das folgende Programmfragment soll zehn Zufallszahlen erzeugen und ausgeben

```

1   double[] p = {0.2, 0.1, 0.25, 0.25, 0.2};
2   for (int t = 0; t < 10; ++t)
3       System.out.println(Simulate(p));

```

Ergänzen Sie folgende Funktion so, dass sie eine Zufallszahl gemäss der gegebenen Wahrscheinlichkeitsverteilung  $p$  zurückgibt.

```

1  static int Simulate (double[] p)
2  {
3      int res=0;
4      double r = Math.random();
5
6
7
8      return res;
9  }

```

## 8 Listen

Eine LinkedList sei aus den folgenden Elementen aufgebaut.

```

1  public class ListElement
2  {
3      public int x;
4      public ListElement next;
5  }

```

Das erste Element der Liste stehe in der Variable "ListElement root;" zur Verfügung. Implementieren Sie untenstehend die Methode `delete`, die das erste Vorkommen der als Parameter gegebenen `int`-Zahl `d` in der Liste entfernt. Bei erfolgreichem Entfernen soll der Boolean-Wert `true` zurückgegeben werden, wenn kein Element mit `x==d` gefunden wurde, soll `false` zurückgegeben werden.

```

1 public Boolean delete(int d)
2 { ...

```

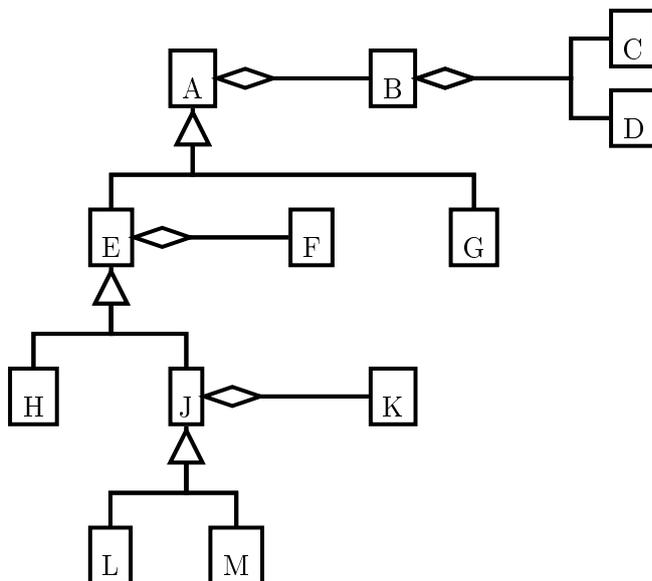
## 9 Klassendiagramm 1

Zeichnen Sie ein Klassendiagramm, das folgende Beziehungen abbildet.

- *Y* "has a" *Z*
- *V* "has a" *U*
- *W* "is a" *Y*
- *Y* "is a" *X*
- *V* "is a" *Y*
- *U* "is a" *Y*

## 10 Klassendiagramm 2

Betrachten Sie nun folgendes Klassendiagramm und markieren Sie in den Statements, ob diese **wahr** oder **falsch** sind.



- A* "has a" *B*
- B* "has a" *A*
- B* "has a" *D*
- E* "has a" *K*
- A* "is a" *E*
- E* "is a" *A*
- F* "is a" *A*
- G* "is a" *A*
- M* "is a" *E*
- M* "is a" *A*

## 11 Grafikprogrammierung

- Vervollständigen Sie die untenstehende Java-Methode `drawLine`. Die Methode soll eine gerade Linie ins Grafikfenster zeichnen. Einzelne Pixel zeichnen Sie mittels dem Aufruf "`graph.set(x,y);`". Um Farben kümmern wir uns in dieser Aufgabe nicht.
- Geben Sie die Laufzeit als Formel für die Anzahl Pixeloperation in Abhängigkeit der vier `int`-Parameter.

```
1 public void drawLine(int x1, int y1, int x2, int y2, EasyGraphics graph)
2 {    ...
```

## 12 BubbleSort

- Sortieren Sie folgende Sequenz von Hand mittels BubbleSort. Schreiben Sie eine neue Zeile für jeden Schritt. Verwenden Sie Pfeile, um zu zeigen, was jeweils zwischen zwei Zeilen verändert wurde.
- Wieviele Schritte benötigen Sie?
- Wieviele Schritte wären schlimmstenfalls (Worst-Case) nötig, um 5 Element mit BubbleSort zu sortieren?

D A E C B

## 13 Numerische Integration

In der Vorlesung haben wir verschiedene Integratoren kennengelernt. Zeigen Sie, dass die Trapezregel und die Rechteckregel für lineare Funktionen dasselbe Resultat liefern.

```
1 public class RectangleIntegrator extends Integrator {
2
3     RectangleIntegrator(int pieces){
4         n= pieces;
5     }
6
7     public double Integrate(Function f, double x0, double x1)
8     {
9         double sum = 0;
10        double width=(x1-x0)/n; // Intervallbreite
11        for (int i = 0; i<n; ++i)
12        {
13            double x = x0 + (i+0.5)*width; // Mitte des Intervalls
14            double y = f.Evaluate(x); // Abgreifen des Funktionswertes
15            sum += y*width; // Rechteckinhalt addieren
16        }
17        return sum;
18    }
19 }
```

```

1 public class TrapezoidalIntegrator extends Integrator{
2     TrapezoidalIntegrator(int pieces){
3         n= pieces;
4     }
5
6     public double Integrate(Function f, double x0, double x1){
7         double sum = 0;
8         double width=(x1-x0)/n;
9         for (int i = 0; i<n; ++i)
10            {
11                double x = x0 + i*width;
12                double y = f.Evaluate(x)
13                    + f.Evaluate(x+width);
14                sum += y*width/2;
15            }
16        return sum;
17    }
18 }

```

## 14 Bäume

Gegeben sei folgende Definition von Bauelementen.

```

1 public class TreeElement
2 {   public String data;
3     public TreeElement lower;
4     public TreeElement higher;
5 }

```

- Schreiben Sie eine rekursive Methode, um alle Strings im Baum in alphabetischer Reihenfolge auszugeben. Ihre Methode wird initial mit dem Wurzelement als Parameter aufgerufen. **Tipp:** Verwenden Sie `System.out.println(...)` für Textausgaben. Suchen Sie nicht zu weit, die Lösung benötigt nur wenige Zeilen Code.
- Zeichnen Sie einen vollständig balancierten Binärbaum, der die Elemente "A", "B", "C", "D", "E", "F", "G" enthält.
- Nennen Sie eine Reihenfolge, in der die genannten Element in einen Binärbaum eingefügt werden müssten, damit dieser stets möglichst balanciert bleibt.
- Zeichnen Sie den resultierenden Baum, falls die genannten Elemente in der gegebenen (alphabetischen) Reihenfolge eingefügt werden.
- Angenommen, ein Binärbaum sei perfekt balanciert, wie tief ist er, wenn er 127 Element enthält?

## 15 Datenbanken 1

Gegeben sind zwei Tabellen in einer MySQL-Datenbank (DB). Zur Veranschaulichung hier zwei Beispiele:

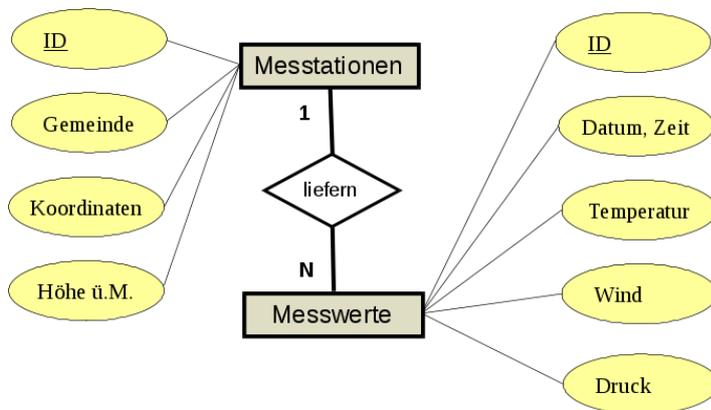
<i>Users</i>		
<u>ID</u>	Name	GameID
1	Anna	1
2	Berta	1
3	Chiara	3
...	...	...

<i>Games</i>	
<u>ID</u>	WinnerID
1	2
2	3
3	3
...	...

- Zeichnen Sie ein ER-Diagramm das vollständig den gegebenen Tabellen entspricht.
- Schreiben Sie das SQL-Statement, um die Namen aller TeilnehmerInnen des Games mit der ID 1 auszugeben.
- Schreiben Sie das SQL-Statement, um die Spielerin Diana einzufügen. Diana spielt in Game 3. *Um die ID von Diana müssen Sie sich nicht kümmern, die wird automatisch von MySQL gesetzt.*
- Chiara hat Game 3 beendet und spielt nun in Game 4. Schreiben Sie das entsprechende SQL-Statement, um den entsprechenden Eintrag in der Tabelle Users anzupassen.

## 16 Datenbanken 2

Betrachten Sie folgendes ER-Diagramm.



- Bilden Sie das Diagramm korrekt auf eine Tabellenstruktur ab. Tatsächliche Daten sind keine im Spiel, sie müssen nur die nötigen Spalten der Tabellen auflisten.
- Schreiben Sie den Ausdruck in **relationaler Algebra**, um Zeitstempel und Temperaturmessungen aller Stationen auszugeben, die höher als 2000müM liegen.
- Beschreiben Sie, was eine **schwache Entität** ausmacht.

*Alles Gute!*

*Viel Glück & Erfolg bei den Vorbereitungen!*

*Felix Friedrich & Lars*