Universität Mannheim
Fakultät für Mathematik und Informatik
Dr. Heinz Kredel

Teilprüfung Software- und Internet-Technologie April 2004: Programmierkurs 1

Name: .... Vorname: ....

Matrikel-Nr.: .... Semester: .... Fach: ....

Hinweise:

1. Bitte füllen Sie sofort den Kopf des Deckblattes aus.
2. Unterschreiben Sie die Klausur auf der letzten Seite.
3. Überprüfen Sie bitte Ihr Klausurexemplar auf Vollständigkeit (**9** Seiten).
4. Tragen Sie die Lösungen – soweit möglich – direkt in die Klausur ein.
5. Eigenes Papier und sonstige Hilfsmittel (z.B. Taschenrechner, Palm, Handy) sind nicht zugelassen.
6. Bearbeitungszeit: 66 Minuten.

Aufgabe	max. Punktzahl	Punkte
1	21	
2	11	
3	11	
4	12	
5	11	
Summe	66	

Aufgabe 1: Programmieren in HTML *[21 Punkte]*

a) (10 Punkte) Schreiben Sie eine vollständige HTML-Seite, die standardkonform ist nach HTML 4.01 und die folgenden Inhalt hat:

Starsky and Hutch

Cinema-Bewertung

Kurzkritik: Die Verfilmung der Kultserie hat mit Ben Stiller und Owen Wilson zwei Stars, die den Original-Helden cool das Wasser reichen.

Humor	Anspruch	Action	Spannung	Erotik
**	-	**	*	*

Beachten Sie dabei, dass die Einträge in der Tabelle zentriert erfolgen.

b) (4 Punkte) Um welche CSS-Anweisungen muss der Code ergänzt werden, wenn der Seitenhintergrund grün, die Schriftfarbe weiß und Überschriften und Tabelle zentriert angeordnet sein sollen?

c) (3 Punkte) Definieren Sie die Begriffe *wohlgeformt* und *gültig* für XHTML-Dokumente.

d) (4 Punkte) Wandeln Sie das folgende HTML-Fragment in ein wohlgeformtes und gültiges XHTML-Fragment um.

```
Was ich diese Woche esse?
<p>
Mal sehen:
<ul title=Essensplan>
  <li>Montag: Huhnfleischw&uuml;rfel
  <li>Dienstag: Zwiebelfleischk&auml;se
</ul>
```

Aufgabe 2: Programmieren in Java *[11 Punkte]*

Schreiben Sie ein vollständiges Java-Programm zum Üben des kleinen Einmaleins, das Folgendes leistet:

- Dem Benutzer werden 20 Fragen der Art "`Wieviel ist 5 mal 8?`" gestellt. Die Zahlen (hier 5 und 8) werden jeweils per Zufallsgenerator aus der Menge $\{1, \ldots, 10\}$ gezogen.
- Beantwortet der Benutzer eine Frage richtig, so lobt ihn das Programm, andernfalls tadelt es ihn. Dann stellt es die nächste Frage.
- Am Ende gibt das Programm noch an, wieviel Prozent der Fragen der Benutzer richtig beantwortet hat.

Für die Eingaben können Sie eine Methode `int readDigit()`, verwenden, die einen Integer-Wert von der Konsole einliest und evtl. auftretende Fehler selbst behandelt.

Aufgabe 3: Objekte in Java *[11 Punkte]*

Gegeben sei eine Klasse `Test` wie folgt:

```
public class Zahl {
  public int x;
  public Zahl() {x=0;}
  public Zahl(int a) {x=a;}
}
```

a) (5 Punkte) Legen Sie ein Feld (Array) aus fünf `Zahl`-Objekten an und initialisieren Sie diese so, dass sie die Werte 1 bis 5 enthalten.

b) (3 Punkte) Welchen Wert haben $a.x$ und $b.x$ nach Ausführung des folgenden Codefragments:

```
Zahl a,b;
a = new Zahl(5);
b = a;
a.x += 1;
b.x += 2;
```

c) (3 Punkte) Was ist eine Klassen-Variable, und wie wird sie deklariert?

Aufgabe 4: Objektorientierte Programmierung in Java *[12 Punkte]*

Gegeben sei die Basisklasse

```
public class Laeufer {
  int position;
  public Laeufer() {position = 0;}
    // initialisiert die Position.
  public int givePosition() {...}
    // gibt aktuelle Position aus.
  public void move() {...}
    // aktualisiert die Position.
}
```

Die abgeleiteten Klassen

```
public class Hase extends Laeufer {...}
public class Igel extends Laeufer {...}
```

überschreiben die `move`-Methode der Basisklasse, d.h. Hasen und Igel bewegen sich auf unterschiedliche Weise. Schreiben Sie nun unter Verwendung dieser Klassen ein vollständiges Java-Programm, das einen Wettlauf implementiert:

- Am Lauf nehmen 50 Läufer teil. Für jeden von ihnen wird per Zufallsgenerator bestimmt, ob es sich um einen Hasen oder einen Igel handelt (Wahrscheinlichkeit jeweils 50%).
- In jeder Runde bewegen sich die Läufer der Reihe nach, indem ihre `move`-Methode ausgeführt wird.
- Wenn der erste Läufer den Zielstrich bei Position 200 überquert, dann hat er gewonnen, und das Rennen wird beendet.

(Platz zur Lösung von Aufgabe 4)

Aufgabe 5: Verwendung von Containerklassen *[11 Punkte]*

Scheiben Sie eine Klasse `Color` zur Speicherung von RGB-Farben, die zusätzlich Namen von Farben speichern kann. D.h. wenn der Konstruktor mit einem Farb-Namen (als Zeichenkette) verwendet wird, soll dieser Name fest mit dieser Zahlenkombination verbunden werden.

```
Color rot = new Color(255,0,0,"rot");
Color r   = new Color(255,0,0);

System.out.println("rot   = " + rot);
System.out.println("r     = " + r);
```

ergibt

```
rot   = rot
r     = rot
```

```
public class Color {



    public final int rot, gruen, blau;

    public int hashCode() {
        return ((rot*256)+blau)*256+gruen;
    }

    public Color(int rot, int gruen, int blau) {
    // Legt neues Objekt an







    }

    public Color(int rot, int gruen, int blau, String name) {
    // Legt neues Objekt an und weist dem Farbcode (rot,gruen,blau)
    // den String name zu. Besteht bereits ein solcher Name fuer
    // den Farbcode, so wird er ueberschrieben.








    }
```

```
    public boolean equals(Object o) {











    }

    public String toString() {
    // Falls ein Name fuer den Farbcode existiert, wird dieser
    // zurueckgegeben. Andernfalls wird der Farbcode selbst
    // (z.B. (255,0,0)) zurueckgegeben.









    }
}
```