Universität Mannheim
Fakultät für Mathematik und Informatik
Dr. Heinz Kredel

Teilprüfung
Software- und Internet-Technologie
Oktober 2003: 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 (**11** 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	14	
2	10	
3	10	
4	11	
5	10	
6	11	
Summe	66	

Aufgabe 1: HTML-Programmierung *[14 Punkte]*

a) [3 Punkte] Schreiben Sie die Body-Section eines HTML-Dokumentes, das nur aus dem Bild `bild1.jpg` besteht. Beim Anklicken des Bildes soll die Seite `inhalt.html` aufgerufen werden.

b) [2 Punkte] Schreiben Sie die zugehörige Head-Section, in der Titel ("Startseite") und Content-Typ spezifiziert werden.

c) [4 Punkte] Geben Sie den XHTML-Code für das folgende Webseiten-Fragment an:

> **Tagesordnung**
>
> 1. Begrüßung
> 2. Planung der nächsten Tagung
>   - Themenwahl
>   - Terminfestlegung
> 3. Schlusswort

d) [5 Punkte] Geben Sie CSS-Regeln für folgende Elemente und Darstellungen an:

da) Das gesamte Dokument soll vor weißem Hintergrund dargestellt werden.

db) Die Überschriften H1 bis H3 sollen fett und rot dargestellt werden.

dc) Unbenutzte Links sollen blau, benutzte Links grün dargestellt werden.

(Platz zur Lösung von Aufgabe 1)

Aufgabe 2: Programmieren in Java 1

[10 Punkte]

Betrachten Sie die folgende Funktion in Java:

```
public int wasndas(int n) {
  int k = 1;
  for(int i=1; i<=n; i++)
    k = 2*k;
  return k;
}
```

a) [2 Punkte] Welche Funktion berechnet `wasndas`?

b) [4 Punkte] Schreiben Sie `wasndas` so um, dass die Methode die gleiche Funktion berechnet, dazu aber eine `while`-Schleife verwendet.

c) [4 Punkte] Schreiben Sie ein Hauptprogramm `main`, das genauso funktioniert wie `wasndas`, das aber den Parameter n als Eingabe von der Konsole erhält.

(Zusatzplatz zur Lösung von Aufgabe 2)

Aufgabe 3: Programmieren in Java 2 *[10 Punkte]*

Gegeben sei die folgende Klasse:

```
public class Wuerfel{
  // die Methode ‘‘werfen’’ gibt eine Zufallszahl
  // zwischen 1 und 6 zurueck.
  public static int werfen() {...}
}
```

Schreiben Sie ein vollständiges Java-Programm, das die Qualität eines solchen Würfels testet. Dazu soll es den Würfel 1000 mal werfen und anschließend ausgeben, wie oft jede der Zahlen 1 bis 6 aufgetreten ist.
Eine Beispiel-Ausgabe könnte wie folgt aussehen:

```
1:  162 mal
2:  170 mal
3:  168 mal
4:  161 mal
5:  172 mal
6:  167 mal
```

Aufgabe 4: Objektorientierte Programmierung *[11 Punkte]*

a) [4 Punkte] Beschreiben Sie, welche Bedeutung die Methoden-Modifizierer `public`, `protected` und `private` haben.

b) [7 Punkte] Gegeben sei die folgende Klasse, die das Rechnen mit reellen Zahlen erlaubt:

```
public class RealZahl{
  double x;
  public RealZahl() {x = 0.0;}
  public RealZahl(double a) {x = a;}
  public double plus(double a) {return (x + a);}
  public double minus(double a) {return (x - a);}
}
```

Leiten Sie eine Klasse `SuperZahl` von `RealZahl` ab, die zusätzlich noch die Methoden `mal` und `geteiltdurch` beherrscht. Beim Versuch, durch Null zu dividieren, soll eine `IllegalArgumentException` ausgeworfen werden; dazu kann einer der Konstruktoren `IllegalArgumentException()` oder `IllegalArgumentException(String)` verwendet werden.

(Platz zur Lösung von Aufgabe 4)

Aufgabe 5: Ein-/Ausgabeströme
[10 Punkte]

Die Klasse `PrintWriter` ist aus der Vorlesung bekannt:

```
public class PrintWriter extends Writer {
  PrintWriter(Writer out) { ... }
}
```

Es soll eine Klasse `Ausgabe` von `PrintWriter` abgeleitet werden, die Zeichenströme sowohl auf den Bildschirm als auch in eine Datei ausgeben kann:

```
public class Ausgabe extends PrintWriter { ... }
```

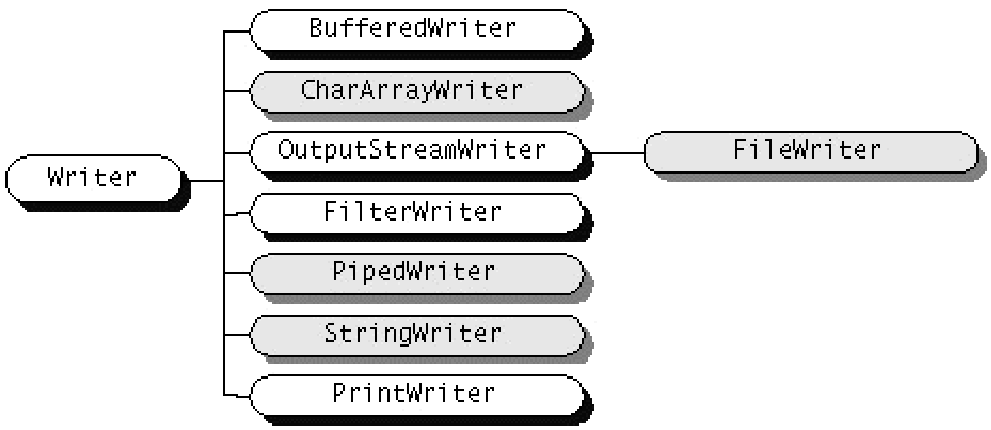
a) [6 Punkte] Implementieren Sie einen Konstruktor eines `Ausgabe`-Objektes, der einen String `fname` als Parameter erhält und damit den Printstrom in die Datei `fname` schreibt. Als `Writer` eignet sich dazu die Klasse `FileWriter` mit dem folgenden Konstruktor:

```
public FileWriter(String fileName)
       throws IOException { ... }
```

b) [4 Punkte] Implementieren Sie einen Konstruktor eines `Ausgabe`-Objektes, der keinen Parameter erhält und daher den Printstrom auf den Bildschirm schreibt. Die Klasse `OutputStreamWriter` eignet sich dazu als `Writer` mit dem folgenden Konstruktor:

```
public OutputStreamWriter(OutputStream out) { ... }
```

Nachfolgende Klassenhierarchie stellt noch einmal das Verhältnis der Klassen `PrintWriter`, `FileWriter` und `OutputStreamWriter` dar.



(Platz zur Lösung von Aufgabe 5)

Aufgabe 6: XML-Dokument *[11 Punkte]*

Gegeben sei folgendes XML-Dokument:

```
<?xml version="1.0" encoding="ASCII" standalone="no" ?>
<!DOCTYPE gaesteliste SYSTEM "gaesteliste.dtd">

<gaesteliste>
<veranstaltung id="abi03-001">
  Abschlussfeier Walter-Moers-Gymnasium
</veranstaltung>
<gaeste>
  <gast><name spitzname="knollennase">Ralf Koenig</name>
        <abschlussarbeit>Der bewegte Mann</abschlussarbeit></gast>
  <gast><name>Thomas Siemensen</name>
        <bild source="haiopei.jpg"/>
        <abschlussarbeit>Ingo Pien der Pinguin</abschlussarbeit></gast>
</gaeste>
</gaesteliste>
```

Geben Sie eine geeignete XML-DTD `gaesteliste.dtd` für dieses Dokument an, so dass es gültig bezüglich dieser DTD ist. Stellen Sie dabei sicher, dass die Veranstaltungs-ID eindeutig ist.