



Klausur Einführung in die Informatik I für Elektrotechniker

3. März 2005

Name:

Matr.-Nr.

Bearbeitungszeit: 120 Minuten

Bewertung (bitte offenlassen :-)

Aufgabe	Punkte	Erreichte Punkte
1	8	
2	9	
3	4	
4	6	
5	4	
6	5	
7	8	
Summe	44	

Spielregeln (**Jetzt lesen!**):

- Benutzen Sie für die Lösung der Aufgaben **nur** das mit diesem Deckblatt ausgeteilte Papier. Lösungen, die auf anderem Papier geschrieben werden, können **nicht** bewertet werden. Schreiben Sie ihre Lösung auch auf die Rückseiten der Blätter; benötigen Sie für eine Lösung mehr als ein Blatt, finden Sie am Ende der Klausur Leerblätter. Zusätzliches Papier können Sie von den Tutoren bekommen.
- Tragen Sie jetzt (vor Beginn der eigentlichen Bearbeitungszeit !!!) auf **allen** Blättern ihren Namen und ihre Matrikelnummer ein.
- Schreiben Sie deutlich! Unleserliche oder zweideutige Lösungen können nicht gewertet werden.
- Schreiben Sie **nicht** mit Bleistift und **nicht** mit rotem oder grünem Stift (das sind die Farben für die Korrektur).
- Lesen Sie die Aufgaben jeweils bis zum Ende durch; oft gibt es hilfreiche Hinweise!
- Kommentare kosten Zeit; kommentieren Sie ihr Programm nur da, wo der Code alleine nicht verständlich wäre.
- Wir weisen noch einmal darauf hin, daß die Benutzung von Taschenrechnern und anderen elektronischen Hilfsmitteln nicht gestattet ist.

Viel Erfolg!

4. (1 Punkt) Was ist der Unterschied zwischen einer *lokalen Variablen* und einem *Attribut*?

5. (1 Punkt) Was verstehen Sie unter der *Redundanz* einer Nachricht?

6. (2 Punkte) Nennen Sie die Definition des Begriffs *Algorithmus*.

Aufgabe 2 (9 Punkte) Java.

1. (2 Punkte) Geben Sie die Werte der Variablen x und y an den mit (1) und (2) gekennzeichneten Stellen im folgenden Programmfragment an.

```
int x = 2;
int y = 3;
for (int i = 2; i > 0; i=i-1) {
    x = x + 1;
    y = y - x;
}
// (1)
x = y;
y = 2 * x;
// (2)
```

2. (2 Punkte) Wieviele Objekte werden von dem folgenden Programmfragment erzeugt? (Die Klassen `Point` und `Line` entsprechen den Klassen aus der Vorlesung.)

```
Point p1 = new Point(0, 0);
Point p2 = new Point(1, 1);
for (int i = 0; i < 4; i++) {
    Line l1 = new Line(p1, p2);
    Line l2 = new Line(new Point(0, 0), new Point(1, 1));
} // for
```

3. (2 Punkte) Wandeln Sie das folgende Programmfragment so um, dass statt der `do`-Schleife eine `while`-Schleife verwendet wird. Das Verhalten des Programms soll nicht verändert werden.

```
int i = Terminal.readInt();
int s = 0;
do {
    s = s + 1;
    i = i - 1;
} while (i > 0);
```

4. (3 Punkte) Welche Fehler enthält folgende Java-Klasse? Geben Sie jeweils die Zeilennummer an und beschreiben Sie den Fehler. Folgefehler (also Fehler, die aus anderen Fehlern resultieren) sollen ignoriert werden.

```
1 class Fehlerhaft {
2     int i;
3     boolean[] fehler;
4     Ordnungshaft(int i) {
5         this.i = i;
6         fehler = new boolean[i];
7     }
8     void korrigieren(long i) {
9         this.i = i;
10    }
11    boolean istFehler(int i) {
12        return fehler[fehler.length];
13    }
14 } // Fehlerhaft
```

Aufgabe 3 (4 Punkte) Zahlensysteme.

1. (2 Punkte) Rechnen Sie die Zahl $123123123_{(4)}$ in eine Zahl zur Basis 16 um. Lassen Sie den Lösungsweg erkennen.
Hinweis: Die Umrechnung muss nicht über das Dezimalsystem erfolgen.

2. (2 Punkte) Berechnen Sie die folgenden Aufgaben unter Verwendung der Zweierkomplementdarstellung mit 4 Bit. Lassen Sie den Lösungsweg erkennen. Bei welchen Aufgaben findet ein Über- bzw. Unterlauf statt? Woran erkennen Sie jeweils den Über- bzw. Unterlauf?

- (a) $3 + (-6)$
- (b) $-2 + 2$
- (c) $-8 + (-1)$

Aufgabe 4 (6 Punkte) Schaltungen.

1. (3 Punkte) Die Beleuchtung für den Innenraum eines Automobils soll elektronisch gesteuert werden. Dabei gelten folgende Rahmenbedingungen:

- Der Wagen verfügt über einen Sensor für die Türen. Das Signal steht Ihnen am Eingang t zur Verfügung (0 = Türen geschlossen, 1 = mindestens eine Tür offen).
- Ein zweiter Sensor meldet das Starten des Motors am Eingang m (1 = Startvorgang läuft, sonst 0).
- Das Licht soll über Ausgang l gesteuert werden (0 = Licht aus, 1 = Licht an).

Entwerfen Sie eine Schaltung mit folgenden Eigenschaften:

- Das Licht soll eingeschaltet werden, sobald eine Tür geöffnet wird.
- Wird die Tür wieder geschlossen, soll das Licht weiterhin im eingeschalteten Zustand *bleiben*.
- Beim Starten des Motors soll das Licht ausgeschaltet werden, falls die Türen geschlossen sind.

Hinweis: Überlegen Sie zunächst, welche Bauteile für eine solche Schaltung notwendig sind.

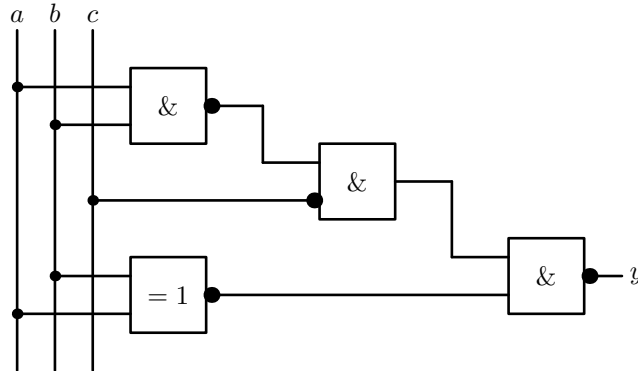
Zeichnen Sie die Schaltung:

t

m

l

2. (3 Punkte) Stellen Sie eine vollständige Wertetabelle für die abgebildete Schaltung auf. Zeichnen Sie dann eine einfachere Schaltung, die dasselbe leistet.



Aufgabe 5 (4 Punkte) Numerik.

Auf einer Geburtstagsparty erklärt ein anwesender Mathematikstudent, dass er eine Formel zur Beantwortung der folgenden Frage kennt: In wie viele Teile kann eine Torte mit n Schnitten maximal geteilt werden?

Um den Gästen das Rechnen zu ersparen, bieten Sie an, ein Programm zur Lösung dieses Problems zu schreiben.

1. (2 Punkte) Schreiben Sie eine Methode `int hoch(int x, int m)`, welche die Funktion x^m berechnet. Verwenden sie **keine** Methoden aus der Klasse `Math`.

2. (2 Punkte) Schreiben Sie eine Methode `int torte(int n)`, welche die folgende Formel berechnet. Verwenden sie **keine** Methoden aus der Klasse `Math`.

$$1 + (-1)^n \cdot \sum_{k=1}^n (-1)^k \cdot k^2$$

Aufgabe 6 (5 Punkte) Array-Operationen.

1. (2 Punkte) Schreiben Sie eine Methode

```
double[] replizieren(double[] d, int n)
```

welche das Eingabe-Array n-mal hintereinanderhängt.

Beispiel: Der Aufruf

```
replizieren({1.2, 4.2, -2.0}, 3)
```

ergibt das Array

```
{1.2, 4.2, -2.0, 1.2, 4.2, -2.0, 1.2, 4.2, -2.0}
```

2. (3 Punkte) Schreiben Sie eine Methode

```
int[] maskieren(int[] a, boolean[] m)
```

welche zwei gleichlange Arrays als Eingabe erhält, ein `int`-Array `a` und ein Array von Wahrheitswerten `m`. Das Resultat der Methode soll ein `int`-Array sein, das alle Elemente aus dem Array `a` enthält, deren entsprechender Eintrag (also der Eintrag mit dem selben Index) im Array `m` den Wert `true` hat.

Beispiel: Der Aufruf

```
maskieren({1, -2, 3, 8}, {true, true, false, true})
```

ergibt das Array

```
{1, -2, 8}
```



Diese Seite wurde absichtlich freigelassen.

Aufgabe 7 (8 Punkte) Klausurverwaltung.

Das Prüfungsamt einer Universität ist überlastet und will deshalb die Verwaltung von Klausuranmeldungen und Ergebnissen auf ein Computersystem umstellen. Zur Modellierung des Problems existiert bereits folgende Klasse zur Repräsentation von Klausurteilnehmern:

```
class Student {  
    String name ;  
    String vorname ;  
    int matrikel ;  
  
    boolean hatSchein ;  
    float note ;  
}
```

Eine Klausur wird beschrieben durch die Liste der Teilnehmer, dargestellt als Java-Array. Erweitern Sie diese Klasse um Methoden zur Lösung von Verwaltungsaufgaben. Sie können dabei davon ausgehen, dass das Attribut `teilnehmer` korrekt initialisiert ist. Eine einfache Methode ist bereits gegeben und darf verwendet werden.

```
class Klausur {  
    Student[] teilnehmer ;  
  
    // hat der i-te Teilnehmer bestanden?  
    boolean bestanden(int i) {  
        return teilnehmer[i].note >= 1.0f & teilnehmer[i].note <= 4.0f ;  
    }  
  
    // weiteres siehe unten  
}
```

1. (2 Punkte) Schreiben Sie eine Methode `boolean doppelt()`, welche prüft, ob es in dem Array der Teilnehmer doppelte Einträge gibt. Dabei sollen die Studenten anhand ihrer Matrikelnummer verglichen werden.
Sie können davon ausgehen, dass die Teilnehmer sortiert sind. Falls doppelte Einträge vorhanden sind, folgen diese also unmittelbar aufeinander.

2. (1 Punkt) Schreiben Sie eine Methode `void disqualifizieren()`, welche allen Teilnehmern, die keinen Schein haben, die Note 5.0 zuweist.

3. (2 Punkte) Schreiben Sie eine Methode `float durchschnittsnote()`, welche den arithmetischen Mittelwert der Noten aller Teilnehmer berechnet.

4. (2 Punkte) Schreiben Sie eine Methode `float durchfallquotient()`, welche den Quotienten aus der Anzahl der durchgefallenen (nicht bestanden) und der Anzahl aller Teilnehmer bestimmt.

5. (1 Punkt) Schreiben Sie eine Methode `void ausgabeBestanden()`, welche alle erfolgreichen Teilnehmer in der Form *Name, Vorname* auf den Bildschirm ausgibt.



Diese Seite wurde absichtlich freigelassen.



Fak. ET/Inform.
Klausur InfET I
3. März 2005

Name:

Matr.-Nr.

Diese Seite wurde absichtlich freigelassen.



Fak. ET/Inform.
Klausur InfET I
3. März 2005

Name:

Matr.-Nr.

Diese Seite wurde absichtlich freigelassen.



Fak. ET/Inform.
Klausur InfET I
3. März 2005

Name:

Matr.-Nr.

Diese Seite wurde absichtlich freigelassen.