Fakultät IV
Elektrotechnik/Informatik

# Probeklausur

# Einführung in die Informatik I

# Musterlösung
# Stand: 5. Februar 2009

**Hinweis: Diese Probeklausur ist eine kleine Aufgabensammlung, die etwa dem Schwierigkeitsgrad der Teilleistung TL 2 (Programmiertest) des Moduls Inftech I entspricht. Die Probeklausur ist umfangreicher als die Teilleistung TL 2. Die Aufgaben decken jedoch nicht alle behandelten Themenbereiche ab.**

**Aufgabe 1 (Bedingte Anweisungen).**

1. Betrachten Sie die folgende Java-Methode

```
boolean f(boolean x, boolean y) {
    if(x) {
        return y;
    } else if(y) {
        return x;
    } else {
        return x;
    }
}
```

(a) Stellen Sie eine Tabelle auf, die für alle möglichen Wertepaare $(x, y)$ der Parameter den zugehörigen Rückgabewert der Methode `f()` angibt.

**Lösung:**

| x | y | f(x, y) |
|---|---|---|
| true | true | true |
| true | false | false |
| false | true | false |
| false | false | false |

(b) Implementieren Sie eine möglichst einfache Methode `f2(boolean x, boolean y)`, die ohne `if`-Anweisung auskommt und für alle möglichen Wertekombinationen von `x` und `y` die gleichen Werte wie die Methode `f()` zurückliefert.

**Lösung:**

```
boolean f(boolean x, boolean y) {
    return x && y;
}
```

2. Betrachten Sie das folgende Punktesystem zur Notenvergabe:

| Punkte | Note |
|---|---|
| 36-40 | 1 |
| 31-35 | 2 |
| 26-30 | 3 |
| 21-25 | 4 |
| 0-20 | 5 |

Implementieren Sie eine Methode `int getNote(int punkte)`, die als Wert die Note der übergebenen Punkte zurückliefert. Liefern Sie den Wert `-1` zurück, falls der Parameter `punkte` keine gültige Punktezahl darstellt.

**Lösung:**
siehe nächste Seite.

```
int getNote(int punkte) {
    if(punkte > 35 && punkte < 41) {
         return 1;
    } else if(punkte > 30 && punkte < 36) {
         return 2;
    } else if(punkte > 25 && punkte < 31) {
         return 3;
    } else if(punkte > 20 && punkte < 26) {
         return 4;
    } else if(punkte >= 0 && punkte < 21) {
         return 5;
    } else {
         return -1;
    }
}
```

**Aufgabe 2 (Schleifen).**

1. Was ist der Wert von `k` nach Ausführung der folgenden Anweisungen?

```
int k = 2;
for(int i = 2; i < 7; i += 2) {
    k *= 2;
}
```

**Lösung:**

`k = 16`

2. Wandeln Sie die folgende `for`-Anweisung in eine `while`-Anweisung um.

```
for(int i = 533; i > -12; i -= 7) {
    System.out.println(i);
}
```

**Lösung:**

```
int i = 533;
while(i > -12) {
    System.out.println(i);
    i -= 7;
}
```

3. Implementieren Sie eine Methode `int[] countZeroes(int[][] tabelle)`, die als Wert die Anzahl der Nullen in jeder Spalte der übergebenen Tabelle zurückliefert. Dabei sei `tabelle[i][j]` der Wert in der $i$-ten Zeile und $j$-ten Spalte der Tabelle.

**Lösung:**

```
// Annahme: tabelle besteht aus n Zeilen und m Spalten, wobei n,m > 0.
int[] countZeroes(int[][] tabelle) {
    int n = tabelle.length;
    int m = tabelle[0].length;
    int[] anzahl0 = new int[m];
    for(int j = 0; j < m; j++) {
        anzahl0[j] = 0;
        for(int i = 0; i < n; i++) {
            if(tabelle[i][j] == 0) {
                anzahl0[j]++;
            }
        }
    }
    return anzahl0;
}
```

**Aufgabe 3 (Klassen und Objekte).**

1. Ergänzen Sie die folgende Java-Klasse wie in den Kommentaren beschrieben.

   **Lösung:**

```
class C {

    int a;

    C(int a) {
        // initialisiere Attribut mit uebergebenen Parameter
        this.a = a;
    }

    void f() {
        a *= 2;
        System.out.println("a = " + a);
    }

    public static void main(String[] args) {
        // Erzeuge ein Objekt der Klasse C. Dabei soll der Wert des
        // Attributs a mit 5 initialisiert werden.
        C c = new C(5);

        // Rufe die Methode f() auf.
        c.f();
    }
}
```

2. Implementieren Sie eine Klasse `Fahrrad`. Die Klasse soll folgenden Anforderungen genügen:

   (a) Der Zustand eines Objekts der Klasse `Fahrrad` wird durch seine Geschwindigkeit und die Anzahl der Gänge beschrieben. Beide Attribute sind vom Typ `int`. Ein Zugriff auf die Attribute außerhalb der Klasse ist nicht möglich.

   (b) Die Klasse besitzt einen Standardkonstruktor, der die Geschwindigkeit mit 0 und die Anzahl der Gänge mit 1 initialisiert.

   (c) die Klasse besitzt einen erweiterten Konstruktor mit zwei Parametern zur Initialisierung der Attribute.

   (d) Die Klasse besitzt eine Methode zum Ändern der Geschwindigkeit. Der Betrag zur Geschwindigkeitsänderung wird als Parameter der Methode übergeben.

   (e) Die Klasse besitzt eine Methode zum Setzen der Anzahl der Gänge. Die Anzahl wird dabei als Parameter der Methode übergeben.

   (f) Auf die Klasse, die Konstruktoren und auf die Methoden kann von jeder anderen Klasse zugegriffen werden.

   **Lösung:**
   siehe nächste Seite

```
0   public class Fahrrad {
1
2       private int speed;
3       private int gears;
4
5       public Fahrrad() {
6           this(0, 1);
7       }
8
9       public Fahrrad(int speed, int gears) {
10          this.speed = speed;
11          this.gears = gears;
12      }
13
14      public void changeSpeed(int value) {
15          this.speed += value;
16          // falls negative Geschwindigkeit nicht zugelassen:
17          this.speed = Math.max(0, speed);
18      }
19      public void changeGears(int value) {
20          this.gears += value;
21          this.gears = Math.max(0, gears);
22      }
23  }
```

**Aufgabe 4 (Vererbung).**

Betrachten sie die Klassenhierarchie in Abbildung 1.

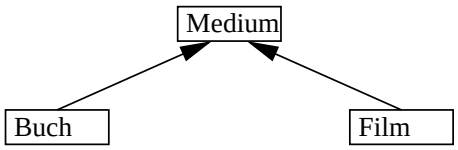


Abbildung 1: Klassenhierarchie

Bücher besitzen einen Titel und einen Autor. Filme werden durch ihren Titel und ihre Spieldauer beschrieben. Implementieren Sie die Klassenhierarchie von Abbildung 1. Gehen Sie dabei folgendermaßen vor:

1. Implementieren Sie eine Klasse `Medium`. Die Klasse `Medium` besitzt als Attribut einen Titel vom Typ `String`. Implementieren Sie einen erweiterten Konstruktur zur Initialisierung des Titels. Stellen Sie sicher, dass Unterklassen der Klasse `Medium` eine abstrakte Methode `void ausgeben()` implementieren. Die Methode `ausgeben()` gibt die Werte aller Attribute der jeweiligen Unterklasse auf der Konsole aus.

2. Implementieren Sie eine Unterklasse `Buch` der Klasse `Medium`. Die Klasse `Buch` besitzt als zusätzliches Attribut einen Autor. Implementieren Sie einen erweiterten Konstruktur zur Initialisierung der Attribute der Klasse `Buch`. Stellen Sie sicher, dass von der Klasse `Buch` Objekte erzeugt werden können.

3. Implementieren Sie eine Unterklasse `Film` der Klasse `Medium`. Die Klasse `Film` besitzt als zusätzliches Attribut die Spieldauer. Implementieren Sie einen erweiterten Konstruktur zur Initialisierung der Attribute der Klasse `Film`. Stellen Sie sicher, dass von der Klasse `Film` Objekte erzeugt werden können.

**Lösung:**

```
0   abstract class Medium {
1
2       String titel;
3
4       Medium(String titel) {
5           this.titel = titel;
6       }
7
8       abstract void ausgeben();
9   }
```

```
0   class Buch extends Medium {
1
2       String autor;
3
4       Buch(String titel, String autor) {
5           super(titel);
6           this.autor = autor;
7       }
8
9       void ausgeben() {
10          System.out.println("Buchtitel : " + titel);
11          System.out.println("Autor     : " + autor);
12      }
13  }
```

```
0   class Film extends Medium {
1
2       int dauer;
3
4       Film(String titel, int dauer) {
5           super(titel);
6           this.dauer = dauer;
7       }
8
9       void ausgeben() {
10          System.out.println("Filmtitel : " + titel);
11          System.out.println("Spieldauer: " + dauer);
12      }
13  }
```

**Aufgabe 5 (Rekursion).**

Betrachten Sie die rekursiv definierte Funktion für alle $n \geq 3$

$$\begin{aligned} f(3) &= 1 \\ f(4) &= 2 \\ f(n) &= 2 \cdot f(n-1) + 3 \cdot f(n-2) \quad \text{für } n > 4 \end{aligned}$$

1. Implementieren Sie eine rekursive Java-Methode, die den Wert $f(n)$ zurückliefert. Die Variable $n$ wird hierbei als Parameter der Methode übergeben. Die Deklaration von lokalen Variablen und Verwendung von Schleifen ist nicht erlaubt.

   **Lösung:**

```
int f(int n) {
    if(n <= 3) {
        return 1;
    } else if(n == 4) {
        return 2;
    } else {
        return 2*f(n-1) + 3*f(n-2);
    }
}
```

2. Zeichnen Sie den Aufrufbaum der rekursiven Methode für $n = 7$.

   **Lösung:**