

Programmieren I

Klausur

Name:

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Vorname:

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Matrikelnummer:

--	--	--	--	--	--	--

Kennnummer:

- angestrebter Abschluss:** Bachelor Master Diplom Magister
- Fachrichtung:** Informatik Wirtschaftsinformatik IST Mathematik
- Mobilität u. Verkehr Physik Maschinenbau Sonstiges: _____

Die Bearbeitungszeit beträgt 120 Minuten. Die Klausur besteht aus 8 Aufgaben. Sie haben die Klausur bestanden, wenn Sie mindestens 36 von 72 möglichen Punkten erreicht haben.

Aufgabe	1	2	3	4	5	6	7	8	Σ
max. Punkte	9	5	8	8	18	10	6	8	72
Punkte									

Note: _____

Bitte prägen Sie sich Ihre Kennnummer gut ein. Aus Datenschutzgründen wird das Klausurergebnis nur unter dieser Kennnummer bekannt gegeben. Aus den gleichen Gründen können Ergebnisse weder telefonisch noch per E-Mail mitgeteilt werden.

Die Ergebnisse der Klausur können Sie ab dem 2. April 2007 auf der WWW-Seite zu dieser Veranstaltung erfahren. Ihre Klausur können Sie am

Mittwoch, den 4. April 2007,

von 9:00–11:00 Uhr und von 13:00–15:00 Uhr im Raum 251 des Informatikzentrums einsehen.

Aufgabe 1: Welche Werte besitzen die Variablen `a` und `b` nach Ausführung der folgenden Anweisungen?

```
a)   a = -7;
      b = 1;
      while (a <=2) {
        a = a + 3;
        b = 1 - 2 * b;
      }
```

`a = 5` `b = 11`

```
b)   a = -3;
      b = 1;
      for (int i = -a; i >= 0; i--) {
        a = a - 3;
        b = 1 + 2 * b;
      }
```

`a = -15` `b = 31`

```
c)   a = 5;
      b = 1;
      do {
        a = (1 + a)/2;
        b = (b + 1) * b;
      } while (a != 1);
```

`a = 1` `b = 42`

9 Punkte

Aufgabe 2: Es sei `a` als `int[] a = new int[5]` deklariert. Geben Sie den Inhalt des Felds nach Ausführung der beiden folgenden Anweisungen an:

```
for (int i = 0; i < a.length; i++)    a[i] = (i*i) % 5;
for (int i = a.length-1; i >= 0; i--) a[i] = a[4-a[i]];
```

`a[0] = 4` `a[1] = 0` `a[2] = 0` `a[3] = 0` `a[4] = 4`

5 Punkte

Aufgabe 3: Gegeben sei die folgende rekursiv definierte Methode `f`:

```
static int f(int x, int y) {
    if (x < 2)
        return 1;
    else if (y <= 2)
        return 2;
    else
        return 2 * f(x - 2, y / 2) + f(x - 1, y + 1);
}
```

Welchen Wert liefert ein Aufruf von `f(6,4)`? In welcher Reihenfolge und mit welchen Parametern wird `f` dabei rekursiv aufgerufen?

8 Punkte

Lösung:

`f(6,4) = 19`

Anzahl der Aufrufe: 13

01. Aufruf: `f(6,4)`
02. Aufruf: `f(4,2)`
03. Aufruf: `f(5,5)`
04. Aufruf: `f(3,2)`
05. Aufruf: `f(4,6)`
06. Aufruf: `f(2,3)`
07. Aufruf: `f(0,1)`
08. Aufruf: `f(1,4)`
09. Aufruf: `f(3,7)`
10. Aufruf: `f(1,3)`
11. Aufruf: `f(2,8)`
12. Aufruf: `f(0,4)`
13. Aufruf: `f(1,9)`

Aufgabe 4: Die Methode `tagNummer` erhält als Parameter einen Tag und einen Monat sowie einen booleschen Wert, der anzeigt, ob es sich um ein Schaltjahr handelt. Die Methode soll die Nummer des Tags zurückgeben. Beispielsweise ist in einem Schaltjahr der 31. 12. der 366. Tag des Jahres. Wenn ein ungültiges Datum, beispielsweise der 31. 6., übergeben wird, soll die Methode den Wert 0 liefern.

```
01 static int tagNummer(int tag, int monat, boolean schaltjahr) {
02     int tn = 1;
03     int[] ml = {31,28,31,30,31,30,31,31,30,31,30,31};
04     if ((schaltjahr))
05         ml[1] = 29;;
06     if (tag <= 0 || monat <= 0 || monat > 12 || tag > ml[monat-1])
07         return 0;
08     for (int i = 0; i <= monat - 1; i++)
09         tn += ml[i];
10     tn += tag;
11     return tn
12 }
```

Das Programmstück

```
13 int t = 26,
14     m = 3;
15 boolean s = falsch;
16 System.out.println(tagNummer(t,m,s));
```

soll mithilfe der Methode `tagNummer` die Tageszahl des 26. März in einem Nichtschaltjahr berechnen. Die Programmzeilen 01 bis 16 enthalten Fehler. Markieren Sie die Fehler im Programmtext. Geben Sie im Lösungsteil an, warum die betreffenden Stellen falsch sind, und notieren Sie die korrigierte Zeile. *Achtung:* Für Programmstellen, die nicht zu einem Fehler führen, aber als fehlerhaft gekennzeichnet sind, werden Punkte abgezogen.

8 Punkte

Lösung:

```
02 int tn = 0;
```

Im Falle `int tn = 1` wäre der 1. Januar schon der zweite Tag des Jahres.

```
08 for (int i = 0; i < monat - 1; i++)
```

Die Tage des aktuellen Monats dürfen nicht vollständig addiert werden.

```
11 return tn;
```

Es fehlt ein Semikolon.

```
15 boolean s = false;
```

Das Literal für den Wahrheitswert `falsch` ist `false`.

Aufgabe 5: Schreiben Sie eine Methode

```
static int maxCount(int[] a),
```

die als Parameter ein Feld ganzer Zahlen erhält. Die Methode soll die Anzahl der Vorkommen des maximalen Werts zurückgeben. Beispielsweise ist 3 der Rückgabewert, wenn **a** das Feld (12, 5, 5, 3, 12, 5, 0, -3, 5, -3, 12) ist, weil das Maximum 12 dreimal in der Folge enthalten ist.

18 Punkte

Lösung:

```
static int maxCount(int[] a) {  
    if (a == null || a.length == 0) return -1;  
    int max = a[0],  
        c = 0;  
    for (int i : a) {  
        if (i > max) {  
            max = i;  
            c = 1;  
        }  
        else if (i == max)  
            c++;  
    }  
    return c;  
}
```

Aufgabe 6: Bitte kreuzen Sie an. Für jede richtige Antwort erhalten Sie einen Punkt, für jede falsche Antwort wird ein Punkt abgezogen. Kein Kreuz bzw. zwei Kreuze bedeuten 0 Punkte. Die minimale Gesamtpunktzahl für diese Aufgabe beträgt 0 Punkte.

Alle Fragen dieser Aufgabe beziehen sich auf Java.

	wahr	falsch
Die Deklaration <code>int i = 5.0</code> erzeugt eine Fehlermeldung.	✓	
Ein Interface kann von höchstens einer Klasse implementiert werden.		✓
Eine Klassenmethode wird durch den Modifikator <code>static</code> definiert.	✓	
Jede Klasse muss eine Methode mit dem Namen <code>main</code> enthalten.		✓
Abstrakte Klassen können nicht instanziiert werden.	✓	
Ein Konstruktor ist eine Methode zum Erzeugen von Objekten.	✓	
Konstruktoren werden vererbt.		✓
Der Typ einer Variablen kann ein Interface sein.	✓	
Alle Methoden einer Klasse müssen unterschiedliche Namen haben.		✓
Für Exceptions der Klasse <code>RuntimeException</code> gilt die <code>catch-or-throw</code> -Regel nicht.	✓	

10 Punkte

Aufgabe 7: Schreiben Sie die Dezimalzahl 125 als Oktal- und als Hexadezimalzahl. Geben Sie die Bitfolge an, durch die der Byte-Wert `-11` in Java gespeichert wird.

6 Punkte

Lösung:

Oktalzahl 175: $125 = 1 \cdot 8^2 + 7 \cdot 8 + 5$
Hexadezimalzahl 7D: $125 = 7 \cdot 16 + 13$
Bitfolge -11: 11110101

Aufgabe 8: Gegeben sei die folgende Java-Methode:

```
static int f(int n) {
    assert n >= 0;          // Vorbedingung P
    int s = 0,
        r = 1;
    for (int i = 0; i <= n; i++) {
        s = s + r;
        r = r + 2;
    }
    assert ...;           // Nachbedingung Q (*)
    return s;
}
```

Welche Werte besitzen die Variablen s und r zum Zeitpunkt der Ausführung der assert-Anweisung (*)? Formulieren Sie eine entsprechende Nachbedingung Q . Schreiben Sie Q als Java-Ausdruck, sodass dieser in der assert-Anweisung (*) verwendet werden kann.

8 Punkte

Lösung: Für die Variablen s und r gelten nach Ausführung der For-Schleife:

$$s = 1 + 3 + 5 + \dots + (r - 2),$$
$$r = 1 + 2(n + 1) = 2n + 3.$$

Daher ist $s = (n + 1)^2 \wedge r = 2n + 3$ eine geeignete Nachbedingung Q . Als Java-Ausdruck geschrieben lautet Q beispielsweise `s == n*n+2*n+1 && r == 2*n+3`.