

Technische Universität Braunschweig

Institut für Programmierung und Reaktive Systeme

Programmieren I

Dr. Werner Struckmann

6. März 2013

Name:

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

Vorname:

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

Matrikelnummer:

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

Kennnummer:

Anrede: ☐ Frau ☐ Herr

Studiengang: ☐ Bachelor ☐ Master ☐ Diplom ☐ Frühstudium ☐ Erasmus

Fachrichtung: ☐ Informatik ☐ Wirtschaftsinformatik ☐ IST ☐ Mathematik

☐ Mobilität und Verkehr ☐ Mechatronik ☐ Maschinenbau ☐ Psychologie

☐ Finanz- u. Wirtschaftsmathematik ☐ Physik ☐ Sonstiges: _____

Die Bearbeitungszeit beträgt 120 Minuten. Die Klausur besteht aus 6 Aufgaben. Sie haben die Klausur bestanden, wenn Sie mindestens 35 von 70 möglichen Punkten erreicht haben.

Aufgabe	1	2	3	4	5	6	Σ
max. Punkte	8	8	10	12	10	22	70
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.

Das Ergebnis Ihrer Klausur erfahren Sie ab dem 22. März 2013 auf der WWW-Seite zu dieser Veranstaltung. Ihre Klausur können Sie am

Montag, den 25. März 2013,

von 09:00 bis 11:00 Uhr und von 14:00 bis 16:00 Uhr im Raum 251 des Informatikzentrums einsehen.

Aufgabe 1: (*Zahldarstellung*) Schreiben Sie die Dezimalzahl 47 als Binär-, Oktal- und Hexadezimalzahl. Durch welche Bitfolge wird der byte-Wert -47 in Java gespeichert?

a) 47 als Binärzahl: _____

b) 47 als Oktalzahl: _____

c) 47 als Hexadezimalzahl: _____

d) Darstellung von -47 : _____

8 Punkte

Aufgabe 2: (*Grundlagen, Objektorientierung*) 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 d = 's';</code> erzeugt eine Fehlermeldung.	<input type="checkbox"/>	<input type="checkbox"/>
Jede Klasse kann zu einer Unterklasse abgeleitet werden.	<input type="checkbox"/>	<input type="checkbox"/>
Zwei abgeleitete Klassen können die gleiche Basisklasse besitzen.	<input type="checkbox"/>	<input type="checkbox"/>
Es kann Variablen geben, die während ihrer Existenz nicht immer auf Objekte des gleichen Typs verweisen müssen.	<input type="checkbox"/>	<input type="checkbox"/>
Statische Methoden dürfen den <code>this</code> -Zeiger nicht verwenden.	<input type="checkbox"/>	<input type="checkbox"/>
In der Anweisung <code>return e</code> muss der Typ des Ausdrucks <code>e</code> stets vom Rückgabetyt der zugehörigen Methode sein.	<input type="checkbox"/>	<input type="checkbox"/>
Für jede Klasse existiert ein Konstruktor ohne Parameter.	<input type="checkbox"/>	<input type="checkbox"/>
Interfaces dürfen nur Methoden enthalten.	<input type="checkbox"/>	<input type="checkbox"/>

8 Punkte

Aufgabe 3: (*Programmverständnis*) Gegeben seien die Methode

```
static void wasPassiert(int[] a) {
    int n = a.length-1;
    int i = 0,
        j = n;
    while (i < j) {
        while (j >= 0 && a[j] <= 0) j--;
        while (i <= n && a[i] > 0) i++;
        if (i < j) {
            int t = a[i];
            a[i] = a[j];
            a[j] = t;
        }
    }
    for (int k : a) {
        System.out.print(k+"");
    }
}
```

sowie das Programmfragment

```
int[] a = {0,1,-2,3,4,-5,6,7,-8};
wasPassiert(a);
```

- a) Wie lautet die Ausgabe des Programmfragments?
- b) Beschreiben Sie die Funktionsweise der Methode `wasPassiert(int[] a)` in Abhängigkeit vom Parameter `a`.

10 Punkte

Lösung:

Aufgabe 4: (*Kontrollstrukturen, Operatoren*) Welche der folgenden Schleifen terminieren ohne Fehlermeldung? Geben Sie in diesem Fall an, welche Werte die Variablen **a** und **b** besitzen, nachdem die jeweiligen Anweisungen ausgeführt wurden.

a)

```
int a = 5;
int b = a - 2;
for (int i = a; i <= a + 3; i++) {
    switch (i % 4) {
        case 0: a = a - 1; b = b - 4;
        case 1: a = a + 1; b = b + 2;
        case 2: a = a - 1; b = 2 + b;
    }
}
```

Die Schleife terminiert: ja ☐ nein ☐ a = _____ b = _____

b)

```
int a = 3;
int b = a + 1;
do {
    a++;
    if ((b % a) == 3) break;
    a = b + 1;
    if ((a % b) == 2) continue;
    b = b + 3;
} while (a != 11);
```

Die Schleife terminiert: ja ☐ nein ☐ a = _____ b = _____

c)

```
int a = 1;
int b = 2;
while (b < 4*a) {
    if (a < 4) {
        a++;
    } else {
        a--;
    }
    b = a + b + 1;
}
```

Die Schleife terminiert: ja ☐ nein ☐ a = _____ b = _____

Geben Sie für jeden der folgenden Ausdrücke den Typ und den Wert an. Setzen Sie vor jedem Ausdruck die Deklaration `int i = 17;` voraus.

d) `(i--)(i/2)` Typ: _____ Wert: _____

e) `(i>>2)>(i/2)` Typ: _____ Wert: _____

f) `i += i -= i+3` Typ: _____ Wert: _____

12 Punkte

Aufgabe 5: (*Rekursion*) Gegeben sei die folgende rekursive Methode:

```
static int f(int x, int y) {  
    if (x <= 1) {  
        return 2;  
    } else if (y <= 0) {  
        return 1;  
    } else  
        return 2*(f(x+1,y-3)-f(x-3,y))+1;  
}
```

Welchen Wert liefert der Aufruf $f(5,4)$? In welcher Reihenfolge und mit welchen Parametern wird f dabei aufgerufen? Geben Sie die Reihenfolge der Aufrufe explizit an. Wie groß ist die maximale Rekursionstiefe, d. h. die maximale Anzahl gleichzeitig aktiver Aufrufe?

Terminiert für $x \geq 0$ und $y \geq 0$ jeder Aufruf $f(x,y)$? Begründen Sie Ihre Antwort.

10 Punkte

Lösung:

Aufgabe 6: (*Felder, Programmerstellung*) Schreiben Sie eine Methode

```
static int[] intersection(int[] a, int[] b) { ... }
```

die als Parameter zwei `int`-arrays `a` und `b` erhält. Diese Methode soll ein `int`-array zurückgeben, das die Elemente genau einmal enthält, die sowohl in `a` als auch in `b` vorkommen. Auf die Reihenfolge der Elemente in der Rückgabe kommt es nicht an. Wenn einer oder beide der Parameter das Null-Objekt sind oder keine `Int`-Werte enthält, soll das Rückgabefeld ein `int`-array sein, das keinen `Int`-Wert enthält. Sie dürfen keine Klassen importieren. Eigene Hilfsmethoden dürfen Sie erstellen. Erläutern Sie Ihren Algorithmus.

Beispiel: Es seien `x` und `y` die folgenden Felder: `x={2,4,3,2,7,0,2,7}`, `y={7,7,8,4,2,4,3,5}`. Zulässige Rückgabewerte sind beispielsweise die Felder `intersection(x,x)={4,3,0,2,7}`, `intersection(x,y)={4,3,2,7}` und `intersection(y,x)={7,2,4,3}`.

22 Punkte

Lösung: