

Technische Universität Chemnitz
Fakultät für Informatik
Prof. Dr. Andreas Goerdt

AuP WS 2003/2004 Nachholklausur 6 Aufgaben, 5 Seiten, Zeit 2 Stunden

Aufgabe 1

(1 + 2 Punkte)

Wir betrachten die Zweierkomplementdarstellung auf 4 Bits.

- (a) Geben sie das Zweierkomplement von -8 an.
- (b) Geben Sie die Summe von -8 und 6 im Zweierkomplement an.

Aufgabe 2

(3 + 3 + 4 Punkte)

- (a) Geben Sie die periodische Dualzahl $(0, \overline{101})_2$ als Bruch im Dezimalsystem (also "ohne Komma") an.
- (b) Wir betrachten die folgende Bitfolge als Wert des Datentyps **float** gemäß dem IEEE Standard 754.

0 10000011 010 0000000000 0000000000

Bit 31

Bit 22

Bit 0.

Geben Sie Exponent, Signifikant (oder Mantisse) und die bezeichnete Zahl selbst, alles im 10er System an.

- (c) Welche der folgenden Aussagen sind wahr?
 - (i) Jede endliche "Kommazahl" zur Basis 7 , wie zum Beispiel $(0, 563)_7$ lässt sich als endlicher Dezimalbruch (= "Kommazahl") darstellen.
 - (ii) Jede endliche Kommazahl zur Basis 8 lässt sich als endlicher Dezimalbruch darstellen.

Aufgabe 3

(2 + 6 Punkte)

Wir betrachten das folgende Programmstück:

```
//Eingabe ist  $y, x \geq 0$  ganzzahlig,  $x, y$  Variablen vom Typ int  
// $z$  ist eine weitere Variable vom Typ int  
 $z = y$ ;  
while( $x > 0$ ) {  
     $z = z * z$ ;  
     $x = x - 1$ ;  
}  
// Ergebnis steht in  $z$ 
```

(a) Geben Sie das Ergebnis des Programms auf z bei Eingabe von $x = 3, y = 2$ und $x = 3, y = 3$ an. Geben Sie die berechnete Funktion in Abhängigkeit von den ganzen Zahlen $x, y \geq 0$ an.

(b) Weisen Sie die Korrektheit Ihrer Aussage aus (a) nach. Gehen Sie dabei nach dem Muster der Vorlesung vor: Schleifendurchläufe. Nachweis einer geeigneten Invariante. Quintessenz.

Aufgabe 4

(1 + 6 Punkte)

Wir betrachten folgendes Programmstück.

```
//Eingabe ist  $x \geq 0$  ganzzahlig,  $x$  Variable vom Typ int  
  
while( $x \geq 3$ ) {  
     $x = x - 3$ ;  
}  
// Ergebnis steht in  $x$ 
```

(a) Geben Sie das Ergebnis des Programms auf x bei Eingabe von $x = 3$ und $x = 5$ an. Geben Sie die berechnete Funktion in Abhängigkeit von der ganzen Zahl $x \geq 0$ an.

(b) Weisen Sie die Korrektheit Ihrer Aussage aus (a) nach. Gehen Sie dabei nach dem Muster der Vorlesung vor: Schleifendurchläufe. Nachweis einer geeigneten Invariante. Quintessenz. Hinweis: Für ganze Zahlen a, b, c und $c > 0$ gilt:

$$\text{Mod}(a + b, c) = \text{Mod}(\text{Mod}(a, c) + \text{Mod}(b, c), c)$$

Aufgabe 5

(2 + 6 Punkte)

Wir betrachten das Problem der “Türme von Hanoi“ mit drei Stäben: A , B , C und n Scheiben. Die Scheiben sind mit 1 (oberste, kleinste) bis n (unterste, größte Scheibe) nummeriert. Die Scheiben sollen unter der bekannten Einschränkung von A nach B umgesetzt werden.

Zusätzlich muss folgende Einschränkung eingehalten werden: Jeder Schritt muss von der Form sein: “ Von A nach B ” oder “ von B nach C ” oder “ von C nach A ”.

(a) Tragen Sie für die Situation auf dem Lösungsblatt jeden einzelnen Schritt bis zum Ende, das heißt alles auf Stab B , ein.

(b) Schreiben Sie eine rekursive Prozedur, die eine zulässige Schrittfolge erzeugt und tragen sie diese in den Rahmen auf dem Lösungsblatt ein.

Hinweis: Es gibt mindestens zwei Lösungen: Eine mit einer Prozedur, die sich dann viermal rekursiv aufruft und eine weitere mit zwei rekursiven Prozeduren, die sich gegenseitig mehrfach aufrufen.

Aufgabe 6

(3 + 3 Punkte)

(a) Wir betrachten folgendes Java-Programm.

```
public class VerdeckenTest {  
    //  
    // Zwei Klassenvariablen static  
    //  
    static int a = 1, b = 2;  
  
    public static void m(int a) {  
        System.out.println("Der Wert von a ist " + a);           // Aufruf 1  
        System.out.println("Der Wert von b ist " + b);  
    }  
  
    public static void main(String[] args) {                   // Zeile 1  
        int a = 7, b = 8;                                       // Zeile 2  
                                                                // Zeile 3  
        m(a);                                                   // Zeile 4  
    }                                                           // Zeile 5  
}
```

(i) Tragen Sie den Inhalt von Prozedurkeller und Heap direkt nach dem Aufruf von $m(a)$, also vor Ausführung der Zeile Aufruf 1 in den Rahmen auf dem Lösungsblatt ein.

(ii) Geben Sie die Ausgabe des Programms an.

(b) Wir betrachten nachfolgendes Java-Programm:

```
public class BooleanMethods {  
    static boolean test1(int val) {  
        System.out.println("test1(" + val + ") ");  
        System.out.println("result: " + (val < 1));  
        return val < 1;  
    }  
  
    static boolean test2(int val) {  
        System.out.println("test2(" + val + ") ");  
        System.out.println("result: " + (val < 2));  
        return val < 2;  
    }  
  
    static boolean test3(int val) {  
        System.out.println("test3(" + val + ") ");  
        System.out.println("result: " + (val < 3));  
        return val < 3;  
    }  
  
    public static void main(String args[]) {  
        if (test1(0) && test2(2) && test3(2)) // ***  
            System.out.println("expression is true");  
        else  
            System.out.println("expression is false");  
    }  
}
```

(i) Geben Sie die Ausgabe des Programms an.

(ii) Wie verändert sich die Ausgabe des Programms, falls in der mit *** gekennzeichneten Zeile alle Operatoren && durch & ersetzt werden?