

Informatik I WS 06/07
Prof. Dr. C. Damm
Dipl.-Inform. Marc Njoku

Übungsblatt 7

Ausgegeben am:	06.12.2006
Abgabe bis:	15.12.2006

Thema: Iteration und Rekursion, Java

Die Theorieaufgaben auf diesem Blatt sind bis zum Freitag, 15. Dezember um 14.00 Uhr in die Info-I-Briefkästen im Erdgeschoss der NAM einzuwerfen. Die Kästen sind nach Übungsgruppen geordnet. Achten Sie bitte darauf dass Sie Ihren Zettel in den richtigen Kasten werfen, und dass **Name und Gruppe** auf **jedem Blatt** stehen! Falls Ihre Lösung mehrere Blätter umfasst, heften Sie diese bitte zusammen. Die Praxis-Aufgaben bearbeiten Sie bitte am Rechner und führen Sie Ihrem Tutor am Rechner vor. Eine Bearbeitung in Zweier-Teams innerhalb Ihrer Übungsgruppe ist möglich.

Aufgabe 1 (Theorie: 20 Punkte):

Speicherorganisation in Java

1. Welchen Wert hat `a` nach Ausführung folgender Java-Anweisungen? Erläutern Sie den Berechnungsablauf.

```
int a = 1; a ^= ++a > 1 ? a : 0;
```

2. Welchen Wert hat `a` nach Ausführung folgender Java-Anweisungen? Erläutern Sie den Berechnungsablauf.

Was passiert, wenn `a` vom Typ `int` wäre?

```
float a = 1; a = a * a++ == 2 ? 2F : 1;
```

Aufgabe 2 (Theorie: 30 Punkte):

Spezifikation, Iteration und Rekursion

Ein Palindrom ist ein Wort bzw. ein Satz, der vorwärts und rückwärtsgelesen, genau gleich ist. (Beispiel: *Ein Esel lese nie.*)

Bei den folgenden Aufgaben kann davon ausgegangen werden, dass ausschließlich GROSSBUCHSTABEN eingegeben werden und das Leerzeichen (Blanks) bzw. Kommata, Punkte o.ä. schon bei der Eingabe ignoriert werden.

1. Geben Sie eine Spezifikation an, für das Problem:
Ist eine gegebene Eingabe $e=e_1...e_n$ ein Palindrom? ($e=e_R$ reicht als Lösung nicht aus! Die Spezifikation soll auf den eingegebenen Buchstaben $e_1...e_n$ erklären, was es bedeutet, dass eine Eingabe vor- und rückwärts gelesen genau gleich ist.).
2. Geben Sie einen rekursiven Algorithmus an, der das Problem bearbeitet.
3. Schreiben Sie einen iterativen Algorithmus, der das Problem bearbeitet.

Aufgabe 3 (Praktisch: 10 Punkte):

Rekursion in Java

Compilieren Sie die folgende Klasse in Java und führen Sie sie einige Male aus. Beobachten Sie was passiert, wenn Sie die letzten beiden Aufrufparameter von `print_segment()` in der Methode `main`

verändern.

Den Code der Klasse Baum.java können Sie kopieren:

```
import java.util.Random;

public class CB {

    Random rnd = new Random();

    public void print_segment(int size, int max_size, int offset,
        int max_offset) {

        int i, r;

        if (size < 0) size = 0;

        for (i = 0; i < offset; i++) System.out.print(" ");

        for (i = 0; i < max_size - size; i++)
            System.out.print(" ");

        System.out.print("/");

        for (i = 0; i < size; i++) {

            r = rnd.nextInt();

            if (size == max_size) {

                if (size % 2 == 0)
                    System.out.print(".~");
                else System.out.print("~.");

            } else {

                if (r < 0)
                    r = -r;
                r = r % 20;

                switch (r) {

                    case 0 :
                        System.out.print("+ ");
                        break;
                    case 1 :
                        System.out.print(" +");
                        break;
                    case 2 :
                        System.out.print("o ");
                        break;
                    case 3 :
                        System.out.print(" o");
                        break;
                    default :
                        System.out.print(" ");
                        break;

                }

            }

        }

        System.out.println("\\");

        if ((size < max_size) && (offset < max_offset))
```

```

        print_segment(size + 1, max_size, offset,
            max_offset);

        else if (offset > 0) print_segment(max_size,
            max_size + 2, offset - 2, max_offset);

    }

    public static void main(String[] args) {

        CB cb = new CB();
        cb.print_segment(0, 0, 10, 10);

    }

}

```

Aufgabe 4 (Praktisch: 20 Punkte):

Berechnung der Wurzel von double-Zahlen

Verwenden Sie folgende Strategie zum Berechnen der Quadratwurzel von d (Double-Wert, $d > 1$):

1. Starten Sie mit dem Intervall von 0 bis d als aktuellem Intervall (from = 0, to = d).
2. Berechnen Sie die Mitte m des aktuellen Intervalls (das aktuelle Intervall geht von "from" bis "to").
3. Betrachten Sie die so entstandenen Intervalle von "from" bis m und von m bis "to".
4. Überprüfen Sie, ob d kleiner ist als m^2 , ob die Wurzel von d also im rechten oder im linken Intervall liegt.
 - Liegt d im linken Intervall? wiederholen Sie das Verfahren mit dem linken Intervall als aktuellem Intervall.
 - Liegt d im rechten Intervall? wiederholen Sie das Verfahren mit dem rechten Intervall als aktuellem Intervall.
5. Beenden Sie das Verfahren nach einer vorgegebenen Anzahl Schritte (steps oder depth) und geben Sie den letzten Wert von m zurück.

Implementieren Sie den Algorithmus einmal iterativ und einmal rekursiv. Verwenden Sie dazu folgenden Programmstumpf:

```

public class SquareRoot {
    private static double calc_rec(
        double d,
        double from,
        double to,
        int depth) {
        // . . .
    }

    private static double calc_ite(
        double d,
        double from,
        double to,
        int steps) {
        // . . .
    }
}

public static void calc(double d, int steps) {
    System.out.println(calc_rec(d, 0.0, d, steps));
    System.out.println(calc_ite(d, 0.0, d, steps));
}
}

```

Eine Testklasse "SquareRootTest" finden Sie unter

<http://user.informatik.uni-goettingen.de/~info1/Java/SquareRootTest.java> .

Hinweis: Sie brauchen keinerlei Methoden aus dem Math- oder aus anderen vordefinierten Java-Packages. Die komplette Aufgabe ist mit den ihnen bislang aus der Vorlesung bekannten "Bordmitteln" lösbar.

Aufgabe 5 (Praktisch: 20 Punkte):

Java

Schreiben Sie ein Java-Programm, das eine natürliche Zahl z ($1 \leq z \leq 999$) einliest und ihren Wert in natürlicher Sprache auf den Bildschirm schreibt, also z.B. bei Eingabe $z = 598$ das Wort "fünfhundertachtundneunzig".