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

Übungsblatt 8

Ausgegeben am:	13.12.2006
Abgabe bis:	22.12.2006

Thema: Iteration und Rekursion, Java, Korrektheit von Algorithmen

Die Theorieaufgaben auf diesem Blatt sind bis zum Freitag, 22. 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

Gegeben Seien die folgenden Klassendefinitionen:

```
class A {
        private int a;
        protected int b;


        public A() {
                a = 0;
                b = 0;
        }

        public void out() {
                System.out.print("\na=" + a + ", b=" + b);
        }

        public void setA(int x, int y) {
                a = x;
                b = y;
        }
}

class B extends A {
        private int c;

        public B() {
                super();
                c = 0;
        }

        public void out() {
                super.out();
                System.out.print(", c=" + c);
        }

        public void setB(int x, int y, int z) {
                setA(x, y);
                c = z;
        }
}
```

Welche der folgenden Codefragmente, die außerhalb der Klassendefinitionen von A und B (in einem

anderen Package) stehen, produzieren Laufzeitfehler? Erläutern Sie, was in einer korrekten Zeile passiert (und welche Ausgabe ggf. produziert wird) und wo der Fehler in einer inkorrekten Zeile liegt.

1.
   (1) `A obj = new B();`
   (2) `obj.setA(1, 2);`
   (3) `obj.out();`
   (4) `((B) obj).setB(2, 3, 4);`
   (5) `obj.out();`
2.
   (1) `A obj = new A();`
   (2) `obj.setA(1, 2);`
   (3) obj.out();
   (4) `((B)obj).setB(1, 2, 3);`
   (5) `((B)obj).out();`
3.
   (1) `B obj = new B();`
   (2) `obj.setB(4, 5, 6);`
   (3) `obj.a = 3;`
   (4) `obj.b = 4;`
   (5) `obj.c = 5;`
4.
   (1) `A[] obj = f new A(), new B() g;`
   (2) `obj[0].setA(1, 2);`
   (3) `obj[1].setA(1, 2);`
   (4) `obj[2].setA(2, 3);`
   (5) `for (int i = 0; i<obj.length; i++) obj[i].out();`

Aufgabe 2 (Praktisch: 20 Punkte):

Iteration und Rekursion in Java

Geben Sie eine iterative und eine rekursive Java-Funktion an, die eine natürliche Zahl *n* als Parameter nimmt und

$$\sum_{i=1}^{n} i$$

(also 1 + 2 + ... + *n*) berechnet.

Zusatzaufgabe (5 Punkte):

Geben Sie eine Funktion an, die die Summe direkt berechnet, d.h. mit einer festen Anzahl von Rechenoperationen für beliebige *n*. Vernachlässigen Sie dabei Integerüberläufe bei großen *n*.

Aufgabe 3 (Praktisch: 10 Punkte):

Rekursion in Java

Implementieren Sie den Euklidischen Algorithmus wie er in den Vorlesungsfolien angegeben ist und vervollständigen Sie damit die "Euklid"-Klasse:

```
public class Euklid {
        public Euklid() {
```

```
                System.out.print("Sie haben soeben ein Objekt vom Typ Euklid erzeugt!";
        }

        .... Hier kommt die Implementierung ...

}
```

Lassen Sie Ihre Lösung in Ihrer Tutorengruppe testieren.

Aufgabe 4 (Theorie: 20 Punkte):

Programmierung in Java

Betrachten Sie folgende Java-Klassendefinition:

```
class A {
        public static void foo (int x, long y) {
                System.out.println(x+y);
        }


        public static void foo (long x, int y) {
                System.out.println(x*y);
        }
}


class B {
        private String x = "foo";

        public void foo (int x, int y) {
                A.foo(x,y);
        }

        public static void main (String[] args) {
                B objB = new B();
                B.foo(1,1);
        }
}
```

1. Geben Sie im obigen Beispiel alle Situationen an, wo Methoden überladen oder überschrieben werden und Attribute überdeckt werden.
2. Was passiert im obigen Beispiel? Sind die Klassen übersetzbar? Geben Sie eine Fehlerbeschreibung des Compilers an oder das Ergebnis, das die Ausführung von B liefert.

Aufgabe 5 (Theorie: 30 Punkte):

Korrektheit von iterativen Algorithmen

Gegeben sei folgendes Programmstück:

```
z=0;
b=true;


do
{
        z=z+1;
        b=!b;
}
while(z != x)
```

wobei x und z int-Variablen sind, x>0.

Beweisen Sie folgende Behauptung:

Nach Beendigung des Programms ist b==true genau dann wenn x gerade ist.

Erklären Sie außerdem, warum die Schleife terminiert.