

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

Übungsblatt 5

Ausgegeben am:	22.11.2006
Abgabe bis:	01.12.2006

Thema: Objektorientierung in Java, Applets und Algorithmen

Die Theorieaufgaben auf diesem Blatt sind bis zum Freitag, 01. 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):

Java-Applikationen und Applets

Zeigen Sie den Unterschied zwischen einer *Applikation* und einem *Applet* auf.

Kann ein Java-Programm gleichzeitig beides sein?

Wieso ist es aus Sicherheitsgründen problematisch, wenn Applets lesend oder schreibend auf Dateien des lokalen Rechners zugreifen können?

Aufgabe 2 (Theorie: 10 Punkte):

Kühe

Ordnen Sie den folgenden Begriffen die Ausdrücke *Klasse*, *Objekt*, *Attribut* und *Methode* zu:

Vera Vollmilch	Kuh	Milchleistung
Elsa Euter	Gras fressen	Farbe
kalben	Anja von der Alm	Rasse
wiederkäuen		

Aufgabe 3 (Theorie: 20 - Praktisch: 5 Punkte):

Rechnen mit Byte

Gegeben sei folgendes Java-Programm:

```
public class RechnenMitByte {
    public static void main(String[] args) {
        short a=83;
        short b=119;
        short c=a+b;
        byte d=(byte) c;
        System.out.println(a);
        System.out.println(b);
    }
}
```

```

        System.out.println(d);
    }
}

```

1. Tippen Sie das Programm ab und bringen Sie es zum Laufen. Liefert das Programm das Ergebnis das man naheliegenderweise erwartet wenn man 83 und 119 addiert? (*Muss vom Tutor testiert werden!*)
2. Vollziehen Sie die Berechnungen Schritt für Schritt nach, die der Computer durchführt (inklusive bitweiser Integer-Addition). Beachten Sie dazu folgendes:
 - In Java werden Zahlen vom Typ **byte** binär codiert, und zwar mit 8-Bit-Zahlen in Zweierkomplementdarstellung, d.h. man kann Zahlen von -128 bis +127 damit darstellen.
 - In Java werden Zahlen vom Typ **short** binär codiert, und zwar mit 16-Bit-Zahlen in Zweierkomplementdarstellung. Man kann damit also Zahlen von -2^{15} bis $+2^{15} - 1$ darstellen.
 - Eine sogenannter "Typecast" von short auf byte, wie in der Zeile `byte d=(byte) c;`, ist eine *einschränkende Konvertierung*.

Führen Sie die einzelnen Schritte des Programmes durch und versuchen Sie nachzuvollziehen, warum der Rechner nicht das Ergebnis liefert, das ein Laie vermutlich erwarten würde. Was genau geht schief?

Aufgabe 4 (Praktisch: 15 Punkte):

Fahrenheit in Celsius umrechnen

Schreiben Sie eine Java-Klasse "Temperature" mit einer Methode "static int fahrenheit2celsius(int fahrenheit)", die den angegebenen Fahrenheit-Wert in Celsius umrechnet und als int zurückliefert. Verwenden Sie dazu die Formel:

$$\text{Grad Celsius} = (5/9) * (\text{Grad Fahrenheit} - 32)$$

Testen Sie ihre Methode mit der folgenden main-Methode:

```

public static void main(String[] args) {
    System.out.println("fahrenheit=" + -50 + " celsius=" + fahrenheit2celsius(-50));
    System.out.println("fahrenheit=" + 0 + " celsius=" + fahrenheit2celsius(0));
    System.out.println("fahrenheit=" + 32 + " celsius=" + fahrenheit2celsius(32));
    System.out.println("fahrenheit=" + 213 + " celsius=" + fahrenheit2celsius(213));
    System.out.println("fahrenheit=" + 451 + " celsius=" + fahrenheit2celsius(451));
}

```

Aufgabe 5 (Praktisch: 30 Punkte):

Java

Schreiben Sie ein Java-Programm mit folgenden Eigenschaften:

Das Programm ermittelt zufällig eine Zahl zwischen 1 und 100. Diese soll vom Benutzer des Programms erraten werden, indem nach jedem Rateversuch des Bedieners eine Ausgabe erscheint mit dem Hinweis: die Zahl ist zu klein, die Zahl ist zu groß oder die Zahl ist richtig.

Beispiel:

```

> Geben Sie eine Zahl ein: 50
> Die Zahl ist zu groß.
> Geben Sie eine Zahl ein: 20
> Die Zahl ist zu klein.
> Geben Sie eine Zahl ein: 30
> Die Zahl ist zu groß.
> Geben Sie eine Zahl ein: 25
> Die Zahl ist richtig.

```

*Hinweis: verwenden Sie dazu die folgende Klasse **Zufallszahlen** (welche zuvor von Ihnen als solche kompiliert werden muss!)*

```
import java.util.Random; // Benutzen einer Java-Package

public class Zufallszahlen {

    public static int[] ziehen(int k) {

        int[] die_zahlen = new int[k];
        Random rnd = new Random(); // Zufallszahlenzieher
        initialisieren
        for (int i = 0; i < k; i++)

            die_zahlen[i] = Math.abs(rnd.nextInt()); // eine
            Zahl ziehen

        return die_zahlen;

    }

    public static int[] ziehen(int k, int max) {

        int[] die_zahlen = ziehen(k);
        for (int i = 0; i < k; i++)

            die_zahlen[i] = die_zahlen[i] % max + 1;
            // modulo-div

        return die_zahlen; // nun alle Werte zwischen 1 und
        max (einschl.)

    }

}
```