7. Übungsblatt: Programmierpraktikum I (WS 2003/04)

Abgabe: 8. Dezember 2003

Bem. 1: Bitte beachte (spätestens) ab diesem Übungsblatt die folgenden Grundregeln des ordentlichen Programmierens:

- Programmstruktur durch Einrückungen herausstellen!
- Quelltext großzügig kommentieren!
- Quelltext, der nicht kompiliert, wird auch nicht korrigiert.

Bem. 2: Bei einigen Aufgaben sollen `int`-Zahlen vom Keyboard eingelesen werden. Dazu kann die folgende Funktion verwendet werden:

```
static int LiesInt() {
  BufferedReader in =
    new BufferedReader(new InputStreamReader(System.in));
  int wert = 0;
  try { wert = Integer.parseInt(in.readLine()); }
  catch (IOException io) {}
  return wert;
}
```

Damit diese Funktion auch läuft, muss außerdem die Bibliothek `java.io.*` importiert werden!

Bem. 3: Im Gegensatz zu späteren Übungsblättern müssen diesmal noch keine Eingabefehler abgefangen werden. Du kannst also davon ausgehen, dass die Eingabe jeweils korrekt ist (bei Aufgabe 1 also beispielsweise $b \neq 0$).

Java: Integer-Division

(`Aufgabe1.java`, 5 Punkte)

Schreibe eine Java-Applikation, die als Eingabe zwei `int`-Variablen a und b erhält und die den Quotienten a/b und den Divisionsrest a mod b zurückgibt. Eine Musterausgabe des Programmes könnte wie folgt aussehen:

```
Zahl a:    77
Zahl b:    15
Quotient:  5
Rest:      2
```

Java: Arithmetisches Mittel

(`Aufgabe2.java`, 6 Punkte)

Schreibe eine Java-Applikation, die das arithmetische Mittel

$$m = \frac{1}{n} \sum_{i=1}^{n} a_i$$

berechnet. Dazu soll das Programm zunächst den Wert n erfragen und dann die Inputs $a_1, \ldots, a_n$ einlesen (`int`-Werte). Als Ausgabe soll der Wert m (als `double`-Wert) zurückgeliefert werden.
Eine Musterausgabe des Programmes könnte wie folgt aussehen:

```
Anzahl der Werte: 4
Wert 1: 15
Wert 2: 7
Wert 3: 1
Wert 4: 4
Arithmetisches Mittel: 6.75
```

Java: Binärdarstellung

(`Aufgabe3.java`, 6 Punkte)

Schreibe eine Java-Applikation, die als Eingabe eine positive `int`-Zahl erhält und als Ausgabe deren Binärdarstellung (ohne führende Nullen) zurückgibt, und zwar ohne die Methode `toBinaryString` zu verwenden.
Eine Musterausgabe des Programmes könnte wie folgt aussehen:

```
Dezimalzahl:       1217
Binaerdarstellung: 10011000001
```