4. Übungsblatt:
Programmierpraktikum I (WS 2002/03)

Abgabe: 9. Dezember 2002

Bem. 1: Für jede Aufgabe ist genau eine Datei abzugeben. Der Name der Datei ist bei der jeweiligen Aufgabe angegeben. So muss die Lösungsdatei zur Aufgabe "Integer-Division" den Namen `aufgabe1.java` haben.
Es sei außerdem auf einige Grundregeln des ordentlichen Programmierens hingewiesen:

- Programmstruktur durch Einrückungen herausstellen!
- Quelltext großzügig kommentieren!

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`, 4 Punkte)

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

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

Java: Arithmetisches Mittel

(`aufgabe2.java`, 5 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`, 5 Punkte)

Schreibe eine Java-Applikation, die als Eingabe einen `int`-Wert erhält und als Ausgabe dessen Binärdarstellung (ohne führende Nullen) zurückgibt.
Eine Musterausgabe des Programmes könnte wie folgt aussehen:

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

Java: Samstagslotto

(`aufgabe4.java`, 6 Punkte)

Schreibe eine Java-Applikation, das Lottozahlen für das Samstagslotto 6 aus 49 ermittelt. Dabei soll gelten:

- Es werden Zahlen zwischen 1 und 49 verwendet.
- Es werden 6 Lottozahlen und 1 Zusatzzahl gezogen.
- Keine Zahl darf doppelt vorkommen.

- Die 6 “normalen” Lottozahlen sollen aufsteigend geordnet ausgegeben werden, die Zusatzzahl wird separat ausgegeben.

Das Programm soll zur Speicherung der Lottozahlen ein Feld (Array) verwenden.
Eine Musterausgabe des Programmes könnte wie folgt aussehen:

```
Lottozahlen (6 aus 49):
1 - 3 - 12 - 13 - 27 - 46 - Zusatzzahl 30
```

Bem.: Um (Pseudo-)Zufallszahlen zu erhalten, muss zunächst ein Pseudozufallsgenerator angelegt werden:

```
java.util.Random rand = new java.util.Random();
```

Der Befehl `rand.nextInt(49)` gibt dann bei jedem Aufruf eine Pseudozufallszahl zwischen 0 und 48 zurück.