



**RWTH AACHEN**  
**Lehrstuhl für Informatik X**  
**Programmierung für Alle (Java)**

---

**Übung 3**

**Abgabe der Lösungszettel: 14. November 2006 vor der Übung**

**Deadline für die Abgabe der Programme per *E-Mail* beim Tutor:**  
**14. November 2006, 8:00 Uhr**

---

Tragen Sie hier Ihre **Gruppennummer** ein:

Tragen Sie hier den **Namen Ihres Tutors** ein:

Die Abgabe der Übung erfolgt in **Dreiergruppen**. Tragen Sie dazu alle Namen und Matrikelnummern Ihrer Gruppenmitglieder hier ein. Der / Die erste in der Tabelle gibt die Übung beim Tutor ab.

NAMEN	MATRIKELNUMMERN

Punktetabelle für die Korrektur:

	Punkte
1 (4)	
2 (8)	
3 (6)	
Total (18)	

1. [4 Punkte] **Pool-Puzzle** (Verständnisaufgabe)

Ein Magisches Quadrat ist eine quadratische Matrix, in der die Summe jeder Zeile, jeder Spalte und der beiden Diagonalen denselben Wert ergibt. Den unvollständigen Code zu dieser Aufgabe finden Sie am Ende des Aufgabenblattes. Das Programm soll feststellen, ob es sich bei einer  $n \times n$ -Matrix um ein Magisches Quadrat handelt.

Ihre Aufgabe ist es, die richtigen Code-Schnipsel aus dem Pool zu nehmen und sie in die Lücken im Code einzusetzen. Jedes Schnipsel darf *nur einmal* verwendet werden, es werden jedoch nicht alle Schnipsel benötigt. Erstellen Sie eine Klasse, die sich compilieren und ausführen lässt und die gezeigte Ausgabe erzeugt.

```
Eingabe:
12 6 15 1
13 3 10 8
2 16 5 11
7 9 4 14

Die Matrix ist ein magisches Quadrat
```

Figure 1. Ausgabe

Tragen Sie die Lösung zu dieser Aufgabe handschriftlich auf dem Blatt ein und geben Sie die Lösung beim Tutor vor Übungsbeginn ab.

2. **Kommunizierende Studenten** (Programmieraufgabe)

- (a) [4 Punkte] Sie sitzen in der Java-Vorlesung und möchten zwei andere Studenten kennenlernen. Jeder Student soll etwas über sich erzählen (Name, Alter, Matrikelnummer, Studiengang).

Fügen Sie die fehlenden Codezeilen in den drei Java-Dateien ein und starten Sie das Gespräch. Tipp: Sie benötigen ein Vorlesungs-Objekt und drei Studenten-Objekte, die Sie in der Vorlesung erstellen müssen. Das Gespräch zwischen den Studenten starten Sie aus dem Hauptprogramm. (Das Flugzeugbeispiel aus der letzten Vorlesung bringt Sie auf den richtigen Kurs :-)

- (b) [3 Punkte] Schreiben Sie eine Klasse Gasthoerer.java. Ein Gasthörer unterscheidet sich dadurch von einem Studenten, dass er keine Matrikelnummer besitzt.

Überlegen Sie sich vier geeignete Instanzvariablen für den Gasthörer. Erstellen Sie zusätzlich zu den drei Studenten in der Vorlesung ein Gasthörer-Objekt. Lassen Sie den Gasthörer ebenfalls an dem Gespräch teilnehmen.

- (c) [1 Punkt] Die Klasse Vorlesung hat noch keine Instanzvariablen. Überlegen Sie sich 4 passende Instanzvariablen und fügen Sie diese im Code ein.

### 3. Arrays und Datentypen (Programmieraufgabe)

- (a) [3 Punkte] Schreiben Sie eine Klasse Arrays.java. Deklarieren Sie in der main-Methode für alle primitiven Datentypen ein eigenes Array. Initialisieren Sie die Arrays mit den folgenden Werten:

- byte: -127, 4, 128
- short: 35480, 42, -33000
- int: -3, 2364, -7456
- float: 3.245, -57.312, 9
- double: 4.94065645, 523.241, 0.111

Kompilieren Sie das Programm. Falls Fehler auftreten, suchen Sie eine Lösung für diese Fehler und ändern Sie das Programm entsprechend.

- (b) [3 Punkte] Schreiben Sie für jedes Array eine eigene Ausgabemethode, die als Parameter ein Array vom passenden Typ erwartet und den Inhalt des übergebenen Arrays auf den Bildschirm ausgibt. Tipp: Die Ausgabemethoden beginnen mit den gleichen Schlüsselwörtern wie die main-Methode, z.B.

```
public static void printByteArray ( ... ) { // Ausgabecode }
```

Benutzen Sie in jeder dieser Methode eine Schleife, um den Inhalt des Arrays auszugeben. Die Methoden müssen auch dann noch funktionieren, wenn sich die Länge der übergebenen Arrays verändert!

Die Ausgabe der Werte auf den Bildschirm soll wie folgt aussehen:

```
byteArray[0] = ...  
byteArray[1] = ...  
...
```

---

### Hinweise zur Übungsabgabe

Bitte geben Sie zu jeder Übung das *Deckblatt*, angeheftet vor Ihren Lösungen, mit ab. *Programmieraufgaben* werden per E-Mail **und** der ausgedruckte Code vor Übungsbeginn beim Tutor abgegeben. *Verständnisaufgaben* werden **handschriftlich** beim Tutor abgegeben.

Sollten Sie Probleme oder Fragen haben, wenden Sie sich bitte an Ihren Tutor oder besuchen Sie die *Betreuungsstunden* im Lila Raum (4U15) im ZIP-Pool der Informatik: Donnerstag von 16:00 bis 18:00 und Freitag von 10:00 bis 12:00.

Häufige Fragen finden Sie auf der *FAQ Seite*:

[http://media.informatik.rwth-aachen.de/programmierung\\_faq.html](http://media.informatik.rwth-aachen.de/programmierung_faq.html)

## Code-Schnipsel und unvollständiger Code zu Aufgabe 1

```

        System.out.println
        public          private          return
            main        class            void
boolean      int      double      float      byte
        printMatrix(a);      sum != magicSum
            Eingabe:      Ausgabe:
            Die Matrix ist ein magisches Quadrat
            Die Matrix ist kein magisches Quadrat
                false      true
        i < n      j < n      i < j      n < i      n < j      i = 0;
            magicSum = magicSum + a[i][i];
            sum = sum + a[i][j];
        n = a.length;      i < a.length      j < a[0].length

```

```

// Magisches Quadrat
public _____ MagischesQuadrat {
    // Print Matrix a
    public static _____ printMatrix(int[][] a) {
        int i = 0;
        while(_____) {
            int j = 0;
            while(_____) {
                _____(a[i][j] + " ");
                j++;
            }
            System.out.println();
            i++;
        }
        _____();
    }
    // Check if a is a magic square
    public static _____ isMagicSquare(int[][] a) {
        ___ i = 0;
        int j = 0;
        int sum = 0;
        int _____
        int magicSum = 0;
        //--- check diagonals
        while(i < n) {
            _____
            i++;
        }
        _____
        while(i < n) {
            sum = sum + a[i][n-i-1];

```

```

        i++;
    }
    if (sum != magicSum) {
        return -----;
    }
    //--- check rows
    i = 0;
    while(-----) {
        sum = 0;
        j = 0;
        while(-----) {
            sum = sum + a[i][j];
            j++;
        }
        if (-----) {
            return false;
        }
        i++;
    }
    //--- check columns
    j = 0;
    while(j < n) {
        sum = 0;
        i = 0;
        while(i < n){
            -----
            i++;
        }
        j++;
        if (sum != magicSum) {
            ----- false;
        }
    }
    return -----;
}

----- static void -----(String[] arg) {

    int[][] a = {{12,6,15,1},{13,3,10,8},{2,16,5,11},{7,9,4,14}};
    System.out.println("-----");
    -----
    if (isMagicSquare(a))
        System.out.println("-----");
    else
        System.out.println("");
}
}

```