

## 2. Übungsblatt : Grundlagen der Programmierung [8 Punkte]

### Aufgabe 1 (Funktionen) [3 Punkte]:

#### „Mr. Summe und Mr. Produkt“<sup>1</sup>

Das folgende mathematische Rätsel eröffnet einige spannende Einblicke in die elementare Zahlentheorie:

Gott wählt zwei Zahlen  $a$  und  $b \in \{2, 3, \dots, N\}$ , wobei  $a = b$  möglich ist, und gibt Mr. Summe die Summe  $S = a + b$  und Mr. Produkt das Produkt  $P = a * b$ . Nun ergibt sich folgender Dialog.

1. Mr. Produkt sagt:  
„Ich kenne die beiden Zahlen nicht.“
2. Mr. Summe sagt:  
„Ich kenne die Zahlen auch nicht, aber ich wusste, dass Du sie auch nicht kanntest.“
3. Mr. Produkt antwortet:  
„Dann kenne ich die Zahlen.“
4. Darauf Mr. Summe:  
„Dann kenne ich sie auch!“

Welche Zahlen hat Gott gewählt?

- a) Lösen Sie das Problem zunächst für  $N = 100$ , d. h. 2 Zahlen zwischen 2 und 100?
- b) Welche Lösung(en) ergeben sich für  $N = 1000$ ?

Hinweis: Erstellen Sie ein Programm, welches für ein gegebenes  $N$  alle korrekten Lösungen (Zahlenpaare) ausgibt. Erstellen Sie zum Prüfen jeder Aussage eine Funktion.  $N$  wird als Argument an das Programm übergeben.

Abgabedatei im Goya: **MrProduktSumme.java**

Aufruf & Ausgabe:

```
$ java MrProduktSumme 100
```

```
1 2
```

---

<sup>1</sup> „Mr. Summe und Mr. Produkt“ von Martin Aigner und Volker Schulze, <http://link.springer.com/article/10.1007/s00591-007-0026-y>

Ausgabe bei mehreren möglichen Lösungen:

```
$ java MrProduktSumme 1000  
1 2  
3 4
```

## Aufgabe 2 (Arrays,Funktionen) [2 Punkte]:

### „Matrix-Operationen“

Im Folgenden sollen die folgenden Matrix-Operationen in Java implementiert werden. Addition, Subtraktion, Multiplikation sowie die Berechnung der transponierten Matrix:

- a.) `static double[][] add(double[][] M1, double[][] M2)`
- b.) `static double[][] sub(double[][] M1, double[][] M2)`
- c.) `static double[][] mul(double[][] M1, double[][] M2)`
- d.) `static double[][] transpose(double[][] M)`

Wenn die Größen der Matrizen für die Matrix-Operation nicht kompatibel sind, soll eine Fehlermeldung an den Nutzer ausgegeben werden und „null“ zurückgegeben werden. Wir beschränken uns auf Matrizen vom Basistyp `double`. Verwenden Sie zum Testen ihrer Funktionen beliebige Matrizen. Beachten Sie dabei auch die Fehlerbehandlung zu überprüfen.

Abgabedatei im Goya: `Matrix.java`

## Aufgabe 3 (Arrays,Funktionen) [3 Punkte]:

### „N-Body Simulation“

Erstellen Sie ein Programm, welches die Bewegung von Planeten simuliert und darstellt. Die genaue Aufgabenstellung finden Sie unter:

<http://sar.informatik.hu-berlin.de/teaching/2012-w/2012-w%20GdP/uebung02/02.htm>

Abgabedatei im Goya: `NBody.java`

## Abgabehinweise

- Die angegebenen Ausgaben zeigen natürlich nicht zwingend die korrekten Resultate, sondern sind **fiktiv**.
- Referenzrechner (dort sollte Ihre Lösung funktionieren):  
[star.informatik.hu-berlin.de](http://star.informatik.hu-berlin.de)
- Die **Abgabe** der Übungsaufgaben erfolgt **in Gruppen**, ideale Gruppengröße=2.  
**Wichtig: Erst Gruppe bilden, dann Lösungen hochladen!**
- Folgende Dateien sind **rechtzeitig über Goya** als **UTF-8** codierte Lösungsquelltexte abzugeben:  
`MrProduktSumme.java`  
`Matrix.java`  
`NBody.java`