



Aufgabe 5.1 (Ü) Strings

Schreiben Sie eine Methode, welche einen Dateinamen (als String) als Argument erhält und die mittlere Länge (als float) der Wörter des in der Datei enthaltenen Textes berechnet und zurückgibt, in dem Sie den Code auf der Übungshomepage erweitern. Mit der vorgegebenen Funktion `readFile` kann eine Textdatei eingelesen werden und der darin enthaltene Text in einem String umgewandelt werden. Sie können einen String aufsplitten in dem Sie die String-Methode `split()` wie folgt verwenden:

```
String [] einzelneWoerter= textAlsString.split("Trennzeichen");
```

wobei das Trennzeichen die Stellen spezifiziert, an dem der Text getrennt wird.

Aufgabe 5.2 (Ü) Fakultät

Die Fakultätsfunktion ist für positive Zahlen wie folgt definiert:

$$n! = n \cdot (n - 1) \cdot \dots \cdot 1 \quad (n \geq 1)$$
$$0! = 1$$

Schreiben Sie ein Programm `Fak.java`, das die Fakultätsfunktion auf zwei Varianten berechnet:

- iterativ* in der Funktion `static int fakIt(int n)` und
- rekursiv* in der Funktion `static int fakRec(int n)`.

Aufgabe 5.3 (Ü) Permutationen eines Strings

Schreiben Sie eine rekursive Methode, welche als Parameter einen String der Länge n akzeptiert und dessen $n!$ Permutationen ausgibt. Überlegen Sie sich dazu wie Sie etwa für die Zeichenfolge `abc` die Permutationen

`abc acb bac bca cab cba`

ausrechnen würden, und verallgemeinern Sie Ihre Überlegung auf n stellige Zeichenketten. Sie können für Ihren Algorithmus die folgenden String-Methoden verwenden:

- `substring(i,j)`: Gibt den Teilstring von Stelle i bis Stelle j zurück.
- `charAt(i)`: Gibt das i -te Zeichen eines Strings zurück.

Aufgabe 5.4 (H) Matrix-Multiplikation

Vektoren und Matrizen lassen sich durch Arrays darstellen. Eine Matrix ist hier durch einen Vektor von Zeilen gegeben. D.h. der Eintrag `m[2][0]` steht in der dritten Zeile an erster Stelle. Im folgenden sollen Methoden zur Multiplikation von Matrizen und Vektoren erstellt werden.

a) Die Multiplikation zweier Vektoren ist folgendermaßen definiert:

$$(a_1 \quad a_2 \quad \dots \quad a_n) \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = \sum_{j=1}^n a_j \cdot b_j$$

Schreiben Sie eine Methode `static int vecvecmul(int[] a, int[] b)`, welche die Multiplikation zweier Vektoren implementiert.

b) Die Multiplikation einer Matrix mit einem Vektor ist folgendermaßen definiert:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_m \end{pmatrix}$$

wobei die Einträge c_i gegeben sind durch

$$c_i = \sum_{j=1}^n a_{ij} \cdot b_j.$$

Schreiben Sie eine Methode `static int[] matvecmul(int[][] a, int[] b)`, welche die Multiplikation einer Matrix und eines Vektors implementiert.

c) Die Transposition einer Matrix ist folgendermaßen definiert:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}^{\top} = \begin{pmatrix} a_{11} & a_{21} & \dots & a_{m1} \\ a_{12} & a_{22} & \dots & a_{m2} \\ \vdots & \vdots & \vdots & \vdots \\ a_{1n} & a_{2n} & \dots & a_{mn} \end{pmatrix}$$

Schreiben Sie eine Methode `static int[][] transpose(int[][] a)`, welche die transponierte zur Matrix a zurückgibt.

d) Die Multiplikation zweier Matrizen ist folgendermaßen definiert:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1k} \\ b_{21} & b_{22} & \dots & b_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nk} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1k} \\ c_{21} & c_{22} & \dots & c_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ c_{m1} & c_{m2} & \dots & c_{mk} \end{pmatrix}$$

wobei die Einträge c_{ij} gegeben sind durch

$$c_{ij} = \sum_{l=1}^n a_{il} \cdot b_{lj}.$$

Schreiben Sie eine Methode `static int[][] matmatmult(int[][] a, int[][] b)`, welche das Produkt der Matrizen a und b zurückgibt.

e) Schreiben Sie eine Methode `main()` um Ihre Methoden zu testen.