



Informatik I

Exercise session 6

Autumn 2020

Hausaufgaben

■ Fragen?

Hausaufgaben

- Fragen?
- Typische Fehler

Hausaufgaben

- Fragen?
- Typische Fehler
- Formatierung

Aufgabe 1: Code Analysis, Typische Fehler

```
int n = In.readInt();  
int x = 1;  
if (n > 0) {  
    int k = 0;  
    boolean e = true;  
    do {  
        if (++k == n) {  
            e = false;  
        }  
        x *= 2;  
    } while (e);  
}
```

$$f(n) \approx 2^n$$

Aufgabe 2: Loop-Mixup 1, Typische Fehler

```
public static void main(String[] args) {  
    Out.print("n? ");  
    int n = In.readInt();  
    int f = 1;  
    if (n != 0) {  
        do {  
            f = f * n;  
            --n;  
        } while (n > 0);  
    }  
    Out.println(f);  
}
```

for (...) {
}

Formatierung

Formatierung

```
int[] vector = new int[10];  
boolean x = true;  
for (int i=0; i<10; i++){  
    if (x){  
        vector[i] = In.readInt();  
    }  
}
```


Formatierung

```
int[] vector = new int[10];  
boolean x = true;  
for (int i=0; i<10; i++){  
    if (x){  
        vector[i] = In.readInt();  
    }  
}
```

Formatierung

```
boolean again;  
    do {  
        Out.print("again? ");  
        again = In.readBoolean();  
  
    if (again){  
        Out.print("b? ");  
        int b = In.readInt();  
        Out.println(b);  
    }  
  
    while (again);
```

Formatierung

```
boolean again;  
do {  
    Out.print("again? ");  
    again = In.readBoolean();  
  
    if (again){  
        Out.print("b? ");  
        int b = In.readInt();  
        Out.println(b);  
    }  
} while (again);
```

Matrix und Vector in Java

Vektor v der Länge n :

$$v = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \Leftrightarrow v = [\underline{v_1}, \underline{v_2}, \underline{v_3}]$$

Matrix und Vector in Java

Vektor v der Länge n :

```
double[] v = new double[n];
```

Matrix M mit n Zeilen m Spalten:

Matrix und Vector in Java

Vektor v der Länge n :

```
double[] v = new double[n];
```

Matrix M mit n Zeilen m Spalten:

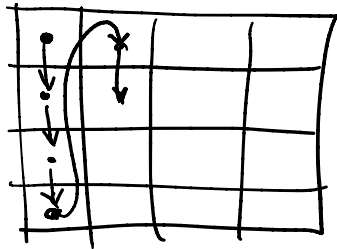
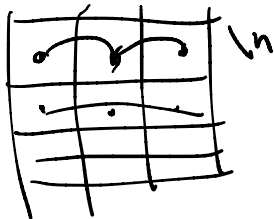
```
double M[][] = new double[n][m];
```

(ein Vektor der Länge n von Vektoren der Länge m)

Aufgabe

Schreibe ein Programm, welches eine $n \times m$ Matrix von integers einliest (von oben nach unten und von links nach rechts). Gib dann die Matrix aus (eine Matrix-Zeile per Output Zeile).

1. Initialisiere eine $n \times m$ Matrix.



Aufgabe

Schreibe ein Programm, welches eine $n \times m$ Matrix von integers einliest (von oben nach unten und von links nach rechts). Gib dann die Matrix aus (eine Matrix-Zeile per Output Zeile).

1. Initialisiere eine $n \times m$ Matrix.
2. Für jede Spalte j (m insgesamt), Zeile i (n in total), lies einen integer vom input und speichere ihn in der Matrix.

Aufgabe

Schreibe ein Programm, welches eine $n \times m$ Matrix von integers einliest (von oben nach unten und von links nach rechts). Gib dann die Matrix aus (eine Matrix-Zeile per Output Zeile).

1. Initialisiere eine $n \times m$ Matrix.
2. Für jede Spalte j (m insgesamt), Zeile i (n in total), lies einen integer vom input und speichere ihn in der Matrix.
3. Für jede Zeile i (n in total), Spalte j (m in total), gib $M[i][j]$ aus (print). Printe eine neue Zeile am ende jeder Matrix-Zeile.

Matrix-Matrix-Multiplikation

Gegeben: Eine $n \times m$ matrix A und eine $m \times p$ Matrix B

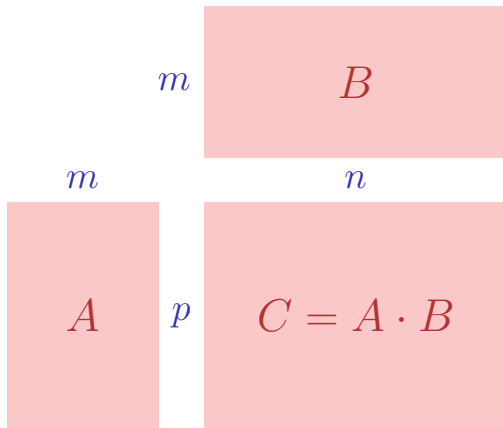
$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ A_{21} & A_{22} & \cdots & A_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nm} \end{bmatrix} \quad B = \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1p} \\ B_{21} & B_{22} & \cdots & B_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ B_{m1} & B_{m2} & \cdots & B_{mp} \end{bmatrix}$$

Matrix-Matrix-Multiplikation

Das Matrixprodukt der $n \times m$ Matrix A mit der $m \times p$ Matrix B ist eine neue $n \times p$ Matrix C

$$A \cdot B = C =: \begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1p} \\ C_{21} & C_{22} & \cdots & C_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n1} & C_{n2} & \cdots & C_{np} \end{bmatrix}$$

Matrix-Matrix-Multiplikation



Matrix-Matrix-Multiplikation

$$C_{ij} = \sum_{k=1}^m A_{ik} \cdot B_{kj} \quad 1 \leq i \leq p, 1 \leq j \leq n$$

Task

Schreibe ein programm, welches 2 Matrizen miteinander Multipliziert.

1. Überlege dir zuerst, welches Format du für den Input brauchen würdest.

Task

Schreibe ein programm, welches 2 Matrizen miteinander Multipliziert.

1. Überlege dir zuerst, welches Format du für den Input brauchen würdest.
2. Überlege dir jetzt, welches Format du für den Output brauchen würdest.

Task

Schreibe ein programm, welches 2 Matrizen miteinander Multipliziert.

1. Überlege dir zuerst, welches Format du für den Input brauchen würdest.
2. Überlege dir jetzt, welches Format du für den Output brauchen würdest.
3. Schreibe Funktionen von In- und Output und teste sie. Teste immer kleine Teile. Was schreibst du zuerst: Input oder Output?

Task

Schreibe ein programm, welches 2 Matrizen miteinander Multipliziert.

1. Überlege dir zuerst, welches Format du für den Input brauchen würdest.
2. Überlege dir jetzt, welches Format du für den Output brauchen würdest.
3. Schreibe Funktionen von In- und Output und teste sie. Teste immer kleine Teile. Was schreibst du zuerst: Input oder Output?
4. Schreibe die Hauptfunktion für die Matrix-Multiplikation.

Task

Schreibe ein programm, welches 2 Matrizen miteinander Multipliziert.

1. Überlege dir zuerst, welches Format du für den Input brauchen würdest.
2. Überlege dir jetzt, welches Format du für den Output brauchen würdest.
3. Schreibe Funktionen von In- und Output und teste sie. Teste immer kleine Teile. Was schreibst du zuerst: Input oder Output?
4. Schreibe die Hauptfunktion für die Matrix-Multiplikation.
5. Teste deine Funktion für ein paar Inputs. Welche Fälle testest du?

Challenge

Benutze jetzt deine funktionierende Funktion für eine Bildtransformation: Behalte die Hauptmethode und ergänze die Matrix-Matrix-Multiplikation.

Nächste Aufgaben

Diese Woche: ☺

- Variable Scope
- Ewiger Kalender (Perpetual Calendar)

Bonus:

- Lauflängenkodierung (Run-length-encoding RLE)
- Korrekte Klammerfolgen (Parenthesis sequences)