PGdP Woche #4 – Arrays und Kontrollfluss

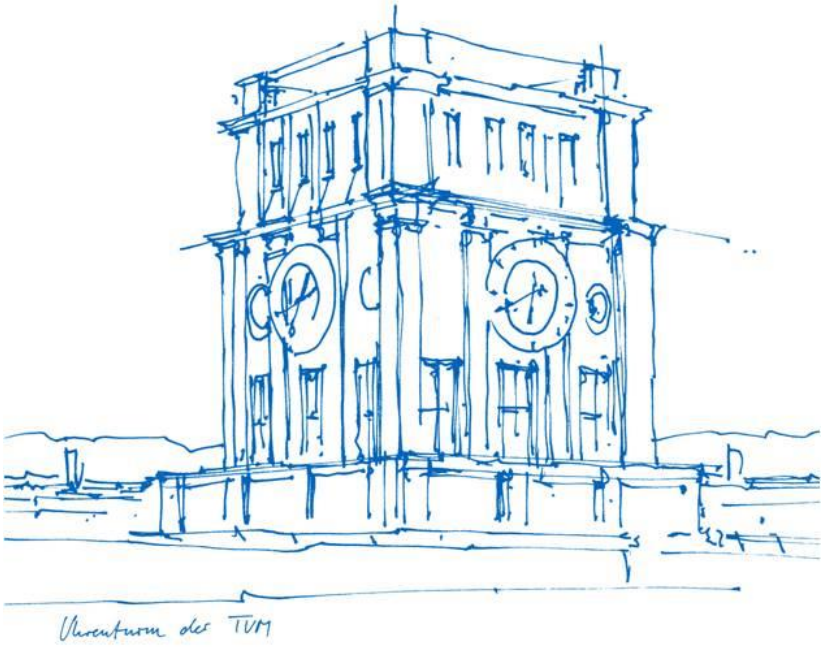
Sebastian Oßner – ossner@in.tum.de

Technische Universität München

Garching, 11.November 2019

Ablauf:

1. P01 – Kontrollflussdiagramm/Zustandstabelle
2. P03 – Zahleneigenschaften
3. P02 – Arrays I
4. P04 – Arrays II

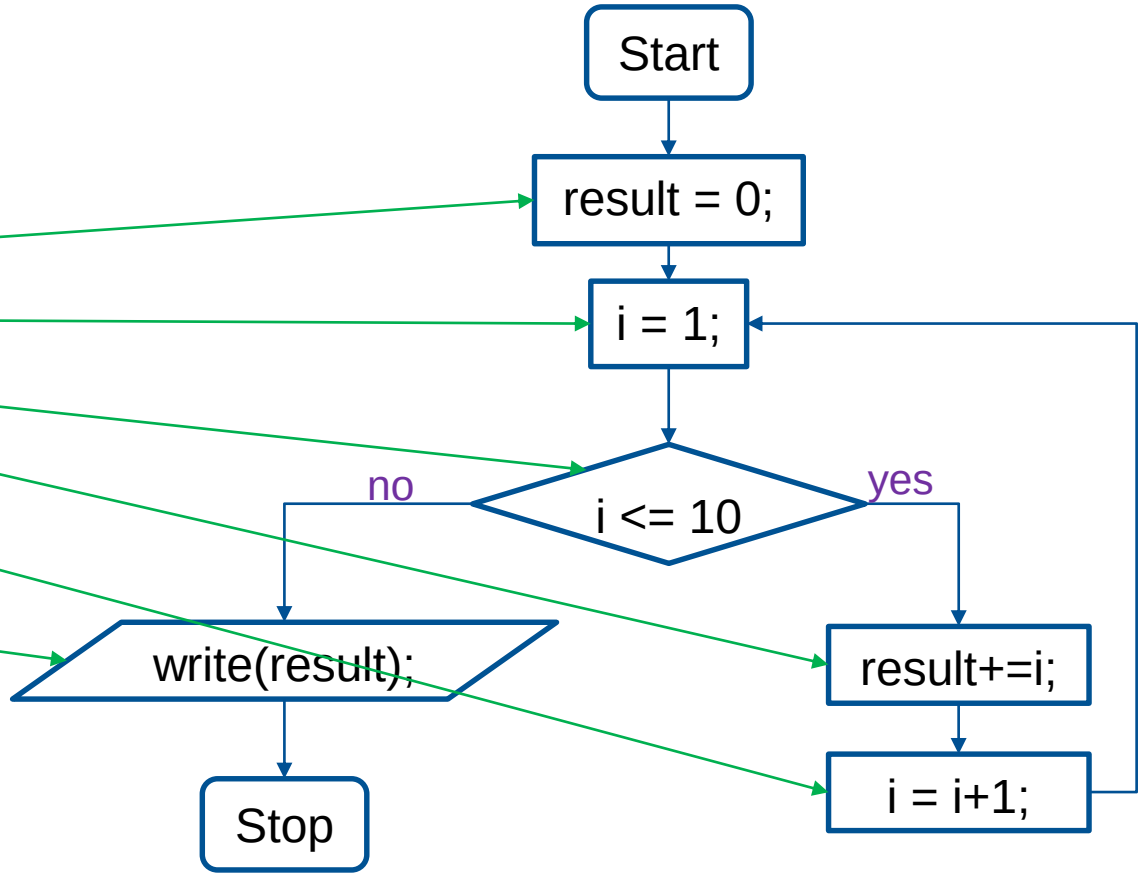


Lehrstuhl für Musterverfahren
Fakultät für Mustertechnik
Technische Universität München

Kontrollflussdiagramm

Darstellen der verschiedenen Rechenschritte/Statements eines Programms

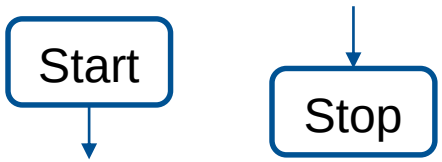
```
int result = 0;
int i = 1;
while(i<=10) {
        result += i;
        i = i+1;
}
write(result);
```



Lehrstuhl für Musterverfahren
Fakultät für Mustertechnik
Technische Universität München

Kontrollfluss - Komponenten

Immer erster und letzter Knoten:

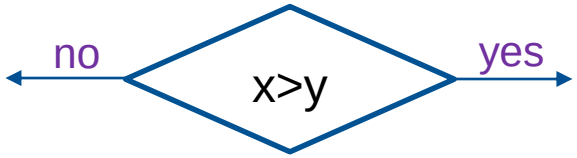


Verbindungen:

Declarations, Calculations:

Function Calls:

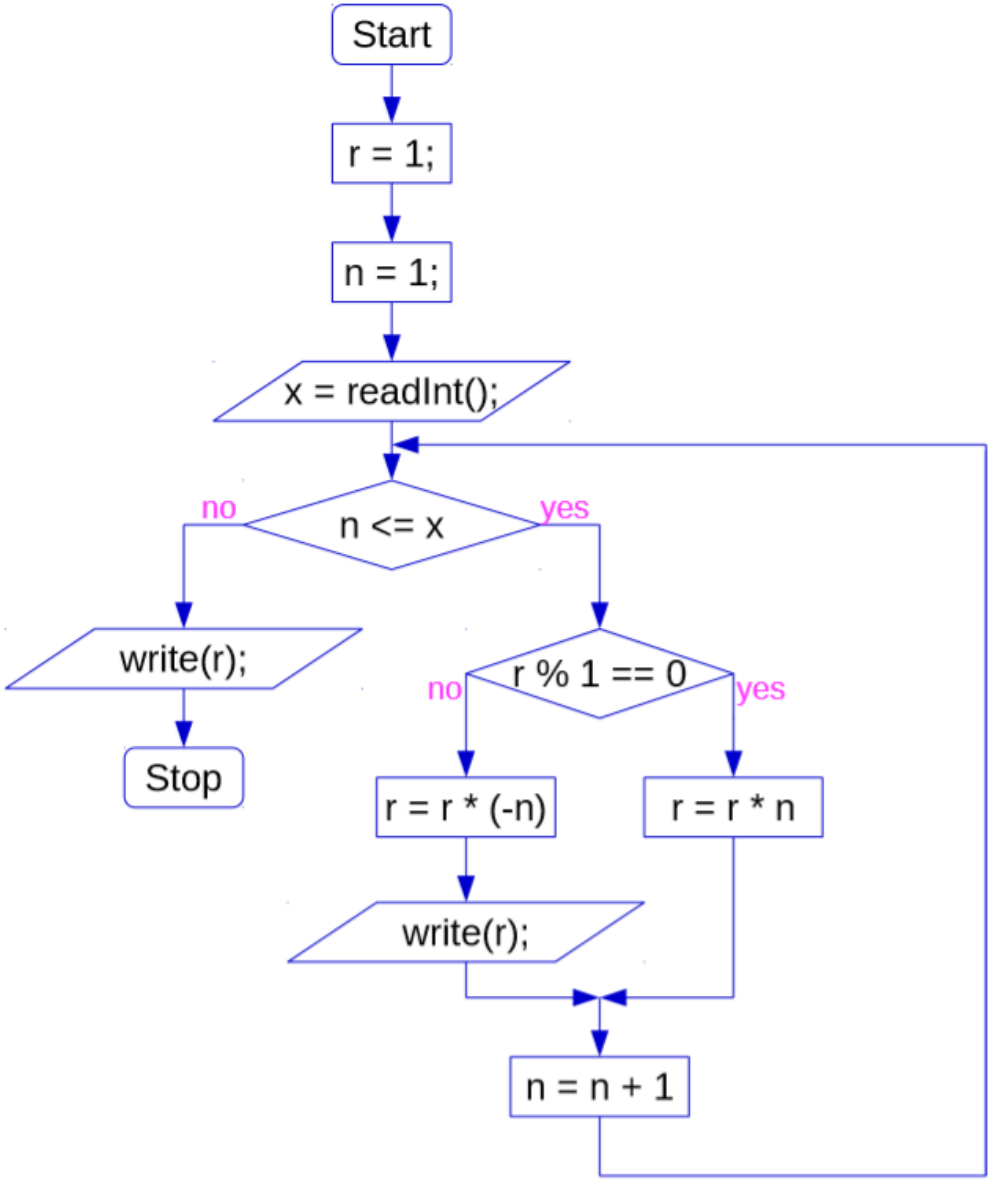
Kontrollstrukturen/Konditionen/Loops



```
1   int x, r, n;
2   r = 1;
3   n = 1;
4   x = readInt();
5   while (n <= x) {
6       if (r % 1 == 0) {
7           r = r * n;
8       } else {
9           r = r * (-n);
10          write(r);
11      }
12      n = n + 1;
13  }
14  write(r);
```

Lehrstuhl für Musterverfahren
Fakultät für Mustertechnik
Technische Universität München

```
1    int x, r, n;
2    r = 1;
3    n = 1;
4    x = readInt();
5    while (n <= x) {
6        if (r % 1 == 0) {
7            r = r * n;
8        } else {
9            r = r * (-n);
10           write(r);
11       }
12       n = n + 1;
13   }
14   write(r);
```



Lehrstuhl für Musterverfahren
Fakultät für Mustertechnik
Technische Universität München

Zustandstabelle

Tabelle, die den Wert von Variablen für relevante Zeilen des Programms ausgibt.

Tabelle hat die Form:

```
int var1 = 2;
int var2 = 5;
var1 = var1+var2;
```

Zeile	var1	var2
1	2	-
2	2	5
3	7	5

- Wert entspricht Wert nach den Berechnungen in der Zeile.
- Als Nutzereingabe in Zeile 4 wird 3 übergeben
- Deklarationen weisen einer Variable keinen Wert zu
- Irrelevant sind Zeilen, in der sich keine Variable ändert

Lehrstuhl für Musterverfahren
Fakultät für Mustertechnik
Technische Universität München

Zahleneigenschaften

Wann ist n eine…

- Primzahl:
  Falls n nur durch 1 oder sich selbst teilbar ist

- Quadratzahl:
  Falls es eine Zahl x gibt, sodass x*x=n

Der Betrag von n ist:

$$|n| := \begin{cases} -n & \text{falls } n < 0 \\ n & \text{sonst} \end{cases}$$

Die Quersumme von n ist die Summe jeder Ziffer von n

Lehrstuhl für Musterverfahren
Fakultät für Mustertechnik
Technische Universität München

Array-Methoden

Alle ausgaben in einer Zeile, kein verwenden von java.util.Arrays

```
int[] tutorial = new int[] {-100, 1, 2, 3, 4, 5, 6, 7, 15, 22};
int[][] tutorial2d = new int[][] {{1, 2, 3}, {4, 5}, {6}};

1. print(tutorial)
        {-100, 1, 2, 3, 4, 5, 6, 7, 15, 22}
2. minUndMax(tutorial)
        Minimum = -100, Maximum = 22
3. invertieren(tutorial)
        {22, 15, 7, 6, 5, 4, 3, 2, 1, -100}
4. schneiden(tutorial, 5) // falls lange>arraylänge, Rest mit 0 füllen
         {-100, 1, 2, 3, 4}
5. linearisieren(tutorial2d)
        {1, 2, 3, 4, 5, 6}
```

Lehrstuhl für Musterverfahren
Fakultät für Mustertechnik
Technische Universität München

Arrayoperationen

Letzte Tutorenbesprechung:
Die Studenten müssen Matrizen nicht verstehen, sie sollen sie nur programmieren

Matrizen in PGdP sind 2d-int Arrays, die an jeder Stelle einen Wert haben

Mathematik: Erster Wert an Index 1
Informatik: Erster Wert an Index 0

Skalarprodukt eines Vekors: [1, 2, 3, 4] = 1*1+2*2+3*3+4*4

Skalar-Matrix-Multiplikation: Jeder Wert der Matrix wird mit einer Zahl multipliziert

Matrix Transposition: Zeilen der Matrix werden zu Spalten der Matrix (Tipp: Visualisierung)

Lehrstuhl für Musterverfahren
Fakultät für Mustertechnik
Technische Universität München

Arrayoperationen

```
int[] tutorial = new int[]{5, 4, 3, 2, 1};
int[][] tutorial2d = new int[][]{{1,2,3},{4,5,6},{7,8,9}};
```

1. scalarProduct(tutorial)

```
        5*5 + 4*4 + 3*3 + 2*2 + 1*1 = 55
```

2. scalarMultiplication(tutorial2d, 3)

```
        {{3, 6, 9},{12, 15, 18},{21, 24, 27}}
```

3. transpose(tutorial2d)

```
        {{1, 4, 7},{2, 5, 8},{3, 6, 9}}
```