PROPÄDEUTIKUM INFORMATIK
SOSE 2018

Martin Mehlhose

KONTROLLSTRUKTUREN UND SCHLEIFEN

FORTSETZUNG

WHILE-SCHLEIFE

- Syntax: **while(condition){statements}**

- condition : beliebiger boolescher Ausdruck. Die schleife wird so lange ausgeführt, wie dieser zu true ausgewertet wird.

- Wichtig: Bei der Abbruchbedingung aufpassen, dass keine Endlosschleifen entstehen!

GIB ALLE ZAHLEN BIS 100 AUS

```
int i =0;

while(i<=100){
    System.out.println(i);
    i++;
}
```

BERECHNE GAUSSCHE SUMME VON 1 BIS I

```
int start =1;
int ergebnis=0;
int i =IO.IOTol.readInt("Gib natürliche Zahl ein:");

while(i>=start){
    ergebnis=ergebnis+start;
    start++;
}

System.out.prinln(ergebnis);
```

WHILE-SCHLEIFE

- grundsätzlich gleichmächtig zur for-Schleife
- Laufvariable oder boolscher Ausdruck muss im Schleifenrumpf verändert werden.

- while-Schleife wird u.a. für gewollte Endlosschleifen benutzt.
- for-Schleife eher zum strukturierten durchlaufen von Datenmengen benutzen

SCHLEIFEN

Explizite Kontrollanweisungen für Schleifen:

- mit **continue;** wird der aktuelle Schleifendurchlauf beendet und mit dem nächsten begonnen.
- mit **break;** wird die aktuelle Schleife beendet.

BEISPIEL CONTINUE UND BREAK

```
int i=0;

while(i<100){
    if(i%5==0){
        i++;
        continue;
    }
    if(i%2==0){
        System.out.println(i);
    }
    if(i%13==0){
        break;
    }
    i++;
}
```

SCOPES - GÜLTIGKEITSBEREICH VON VARIABLEN

- Variablen, die am Beginn eines Programmblocks deklariert wurden, sind in diesem Block und dessen inneren Blöcken verfügbar.
- Eine bereits deklarierte Variable kann in einem inneren Block nicht neu deklariert werden.
- alle Änderung, die an einer Variable in einem inneren Block vorgenommen werden, bleiben nach Beendung des inneren Blockes erhalten.

BEISPIEL SCOPES

```
public static void main(String[] args){
    //Scope 1
    int a=0;
    {// Scope 2
        int b=2;
        a++;
        System.out.println("a: "+a+" b: "+b);
    }
    System.out.println(a); // b not known
}
```

MASCHINENZAHLEN

WARUM MIT MASCHINENZAHLEN BESCHÄFTIGEN?

BEISPIEL DIVISION

```
public static void main(String[] args){
    int a=5;
    int b=3;
    int ergebnis=a/b;
}
```

BEISPIEL DIVISION

```
00000: 00 00001 00 000 1100 01 01 1100 0
00001: 00 00011 00 001 1101 01 01 1100 0
00010: 10 00111 01 000 0000 01 10 0001 0
            00011: 00 00010 00 011 0000 01 01 1100 0
00100:
00101:
00110: 10 01001 01 001 0000 01 10 0001 0
            00111: 00 11111 00 000 0000 00 00 0000 0
01000: 00 01011 00 001 0001 01 00 0010 0
01001: 00 01010 00 111 1111 01 01 1100 0
            01010: 00 00000 11 111 0000 00 01 1100 0
01011: 10 01100 00 000 0001 01 00 0101 0
01100: 00 01011 00 011 0000 01 01 0101 0
01101: 00 01110 00 111 1111 01 01 1100 0
            01110: 00 01111 11 111 0011 00 00 1100 0
```

BEISPIEL RECHNEN MIT BUCHSTABEN

```
public static void main(String[] args){
    char buchstabe = 'a';
            buchstabe = (char) ((char) buchstabe + 5);
            System.out.println(buchstabe);
}
```

PROBLEME MIT MASCHINENZAHLEN

```
public static void main(String[] args){
    for(double i= 2.0;i<3.0;i=i+0.2){
        System.out.println(i);
    }
}
```

PROBLEME MIT MASCHINENZAHLEN

```
public static void main(String[] args){
    for(double i= 2.0;i<3.0;i=i+0.2){
        System.out.println(i);
    }
}
Ausgabe:
1.0
1.2
1.4
1.5999999999999999
1.7999999999999998
1.9999999999999998
2.199999999999997
...
```