

Prof. Dr. Verena Majuntke

Grundlagen der Programmierung

Kontrollstrukturen

Prog1

Lernziele

- Was gibt es für Kontrollstrukturen in Java ?
- Was ist eine Sequenz?
- Wie funktionieren Schleifen mit *while*, *do...while*, *for* ?
- Wie verzweigt man mit *if / else* ?
- Wann werden switch/case – Statements genutzt?
- Wie kann Programmcode unterbrochen und weitergeführt werden?

Übersicht: Kontrollstrukturen

- Sequenzen
- Wiederholungen durch Schleifen
(`while`, `do ... while`, `for`)
- Verzweigungen (mittels Ausdrücken)
(`if / else`)
- Fallunterscheidungen
(`switch / case`)

Sequenzen

Sequenzen Struktogramm



Sequenzen

- Werden nacheinander „sequentiell“ von oben nach unten ausgeführt

```
public class HelloWorld {  
  
    public static void main(String[] args)  
    {  
        String message;  
        message = "Hello World!";  
        System.out.println(message);  
    }  
}
```

Sequenzen

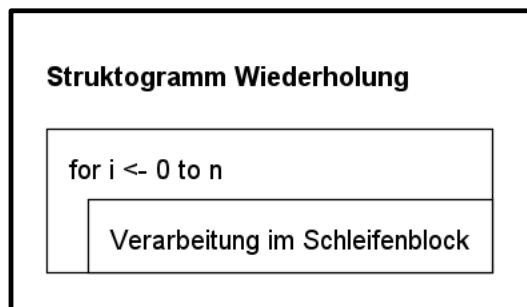
- Werden nacheinander „sequentiell“ von oben nach unten ausgeführt

```
public class HelloWorld {  
    public static void main(String[] args) {  
  
        String message; 1  
        message = "Hello World!"; 2  
        System.out.println(message); 3  
    }  
}
```

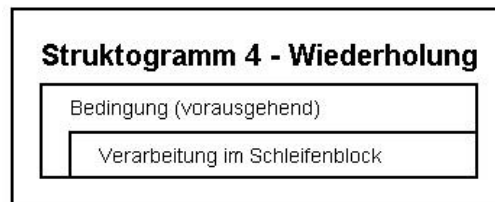
Schleifen

Schleifen

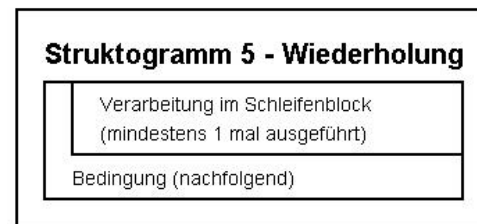
- Wiederholungen durch Schleifen



for-Schleife



while-Schleife



do while-Schleife

for-Schleife

Syntax:

```
for (Initialisierung; Schleifenbedingung; Schrittweite)
{
    Anweisungen;
}
```

for-Schleife

```
for (int i = 0; i < 10; i++) {...}
```

for-Schleife

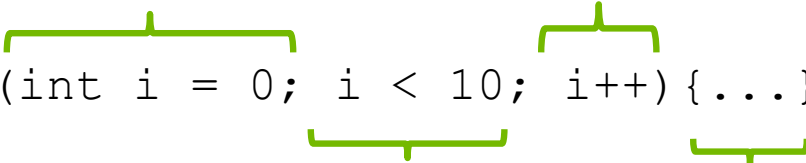
Initialisierung –

wird nur 1mal durchlaufen

Schrittweite –

i.d.R. Inkrementierung des
Schleifenzählers

```
for (int i = 0; i < 10; i++) { ... }
```



Schleifenbedingung –

Bedingung wird auf `true` geprüft

Schleifenblock

for-Schleife: Reihenfolge

```
for (int i = 0; i < 10; i++) {  
  
    System.out.println(i);  
  
}
```

Übung

Multiplikation durch wiederholte Addition:

$$1*a = a$$

$$2*a = a + a$$

$$3*a = a + a + a$$

$$4*a = a + a + a + a$$

...

$$n*a = \underbrace{a + a + a + a + a + \dots + a}_{n\text{-mal}}$$

n-mal

Lösung

```
public static int multiply (int a, int n) {  
    int ergebnis = 0;  
    for (int i = 0; i<n; i++) {  
        ergebnis = ergebnis + a;  
    }  
    return ergebnis;  
}
```

for-Schleife: alle 3 Teile optional

- `for (int i = 0; i < 10; i++) {...}`
- `for (; i < 10;) {...}`
- `for (int i = 0; ; i++) {...}`
- `for (;;) {...}`
- `for (int i = 0; i < 10;) {...}`

while-Schleife

Syntax:

```
while (Schleifenbedingung)
{
    Anweisungen;
}
```

while-Schleife

```
while (i < 10) {...}
```



Schleifenblock

Schleifenbedingung –
Bedingung wird auf `true` geprüft

while-Schleife

Was darf nicht vergessen werden?

```
while (i < 10){  
    System.out.println("i= " +i);  
  
}
```

while-Schleife

Was darf nicht vergessen werden?

```
while (i < 10){  
    System.out.println("i= " +i);  
    i++; //Schleifenzähler inkrementieren  
}
```

while-Schleife

Was darf nicht vergessen werden?

```
int i = 0; //Schleifenzähler initialisieren
while (i < 10){
    System.out.println("i= " +i);
    i++; //Schleifenzähler inkrementieren
}
```

Übung

Schreiben Sie eine Methode für die Multiplikation durch Addition.
Nutzen Sie dazu eine while-Schleife.


do while-Schleife

Syntax:

```
do {...} while (Schleifenbedingung);
```

do while-Schleife

```
do {...} while (i < 10)
```



Schleifenblock

Schleifenbedingung –
Bedingung wird auf `true` geprüft

do while-Schleife

Was darf nicht vergessen werden?

```
do{  
    System.out.println("i= " +i);  
  
} while (i < 10);
```

do while-Schleife

Was darf nicht vergessen werden?

```
do{  
    System.out.println("i= " +i);  
    i++; // Schleifenzähler inkrementieren  
} while (i < 10);
```

do while-Schleife

Was darf nicht vergessen werden?

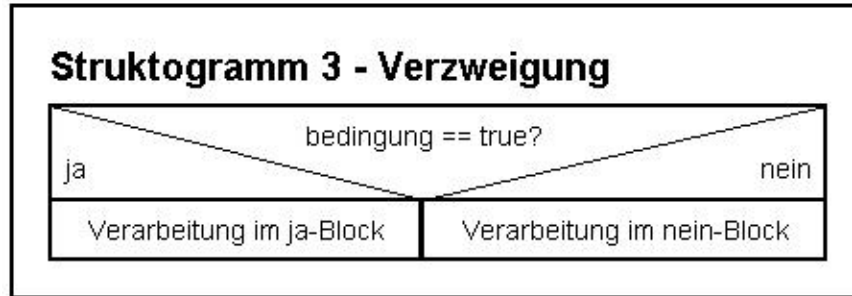
```
int i = 0; // Schleifenzähler initialisieren
do{
    System.out.println("i= " +i);
    i++; // Schleifenzähler inkrementieren
} while (i < 10);
```

Übung

Schreiben Sie eine Methode für die Multiplikation durch Addition.
Nutzen Sie dazu eine do while-Schleife.

Verzweigungen

if/else Alternative



if/else Alternative

Syntax:

```
if (Bedingung) {...} else {...};
```

Bedingung –
wird auf `true` geprüft

if-block –
falls Bedingung zu `true` auswertet

else-block –
falls Bedingung zu `false` auswertet

if/else

```
if ( x > y ) {  
    System.out.println(" Ja : x größer y" );  
}  
else {  
    System.out.println(" Nein : x kleiner/gleich y" );  
}
```


else-Teil optional

```
if ( x > y ) {  
    System.out.println(" Ja : x größer y" );  
}
```

Hier gibt es keinen else-Teil

Übung

Prüfen sie die Eingabevariable `e` vom Typ `char`: Wenn die Eingabevariable `,j`` oder `,J`` enthält geben Sie „Ja“ auf der Konsole aus, andernfalls „Nein“

Tipp: Denken Sie auch an logische Ausdrücke und Operatoren !!!

Baumelndes else

```
if ( x > y )  
  
    if( a == b ){  
        }  
  
else {  
    System.out.println(„“);  
}
```

Baumelndes else

```
if ( x > y )
```

```
    if( a == b ) {  
    }
```

Gehört es zum inneren if?

```
else {  
    System.out.println(„“);  
}
```

Baumelndes else

```
if ( x > y )
```

```
    if( a == b ){  
    }
```

Gehört es zum äußeren if?

```
else {  
    System.out.println(„“);  
}
```

Baumelndes else

```
if ( x > y )
```

```
    if( a == b ) {  
    }
```

Es gehört zum inneren if!

```
else {  
    System.out.println(„“);  
}
```

Baumelndes else

```
if ( x > y ) {
```

```
    if( a == b ) {  
    }
```

```
}
```

```
else {
```

```
    System.out.println(„“);
```

```
}
```

Es gehört zum äußeren if
nur wenn Klammern gesetzt sind!

else/ if

```
if ( x > y ) {  
    System.out.println("Fall 1");  
} else {  
    if(x == y){  
        System.out.println("Fall 2");  
    } else {  
        if(x < y){  
            System.out.println("Fall 3");  
        }else{  
            System.out.println("Default");  
        }  
    }  
}
```


el(se/) if

```
if ( x > y ) {  
    System.out.println("Fall 1");  
} else  
    if(x == y){  
        System.out.println("Fall 2");  
    } else  
        if(x < y){  
            System.out.println("Fall 3");  
        }else{  
            System.out.println("Default");  
        }  
}
```

el(se/) if

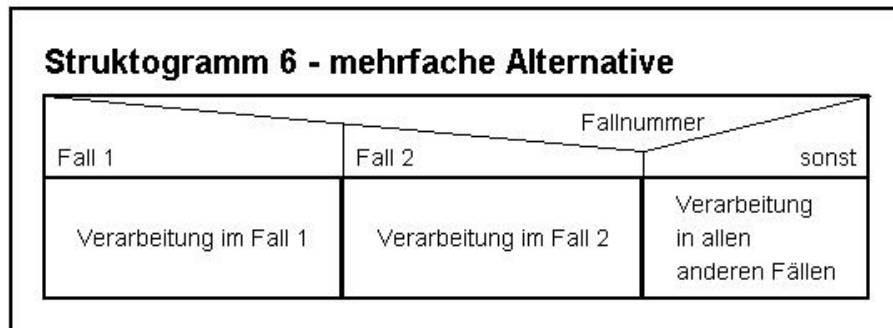
```
if ( x > y ) {  
    System.out.println("Fall 1");  
} else if(x == y){  
    System.out.println("Fall 2");  
} else if(x < y){  
    System.out.println("Fall 3");  
}else{  
    System.out.println("Default");  
}
```

el(se/) if

```
if ( x > y ) {  
    System.out.println("Fall 1");  
  
} else if(x == y){  
    System.out.println("Fall 2");  
  
} else if(x < y){  
    System.out.println("Fall 3");  
  
}else {  
    System.out.println("Default");  
}
```

Fallunterscheidung

Fallunterscheidung



switch/case Fallunterscheidung

Syntax

```
switch(expression)
{
    // case-Ausdrücke, Datentypen müssen übereinstimmen.
    case value1 : // Anweisungen
    case value2 : // Anweisungen
    ...
    default : // Statements
}
```

switch/case

```
int x;  
switch(x)  
{  
    case 0 : System.out.println("Null");  
    case 10 : System.out.println("Zehn");  
    case 20 : System.out.println("20");  
    default : System.out.println("Alles andere!");  
}
```



Kontrollfluss
wird nicht
unterbrochen!

switch/case


```
int x;  
switch(x)  
{  
    case 0 : System.out.println("Null");  
             break;  
    case 10 : System.out.println("Zehn");  
             break;  
    case 20 : System.out.println("20");  
             break;  
    default : System.out.println("Alles andere!");  
}
```



Kontrollfluss
wird
unterbrochen!

switch/case

```
char x = 'J';  
switch(x)  
{  
    case '0' : System.out.println("Nix");  
    case 'J' :  
    case 'j' : System.out.println("Ja");  
    case 'k' : System.out.println("Nein");  
    default : System.out.println("Alles andere!");  
}
```



Kontrollfluss
wird nicht
unterbrochen!

switch/case

```
char x = 'J';  
switch(x)  
{  
    case '0' : System.out.println("Nix"); break;  
    case 'J' :  
    case 'j' : System.out.println("Ja"); break;  
    case 'k' : System.out.println("Nein"); break;  
    default : System.out.println("Alles andere!");  
}
```



Kontrollfluss
wird
unterbrochen!

Übung

Prüfen sie die Eingabevariable `e` vom Typ `char`: Wenn die Eingabevariable `,j'` oder `,J'` enthält geben Sie „Ja“ auf der Konsole aus, wenn `,n'` oder `,N'` „Nein“, andernfalls „Unbekannt“ aus. Jeweils eine Lösung mit `else if` und `case`.

Unterbrechung und Weiterführung

break und continue

break

- Unterbrechung des Kontrollflusses bei Fallunterscheidung (switch/case)
- Unterbrechung und Ende einer Schleife

continue

- Unterbrechung und Weiterführung einer Schleife (mit der nächsten Iteration)

break in switch /case

```
char x;
```

```
switch( x) {  
  case '0' : System.out.println(" Null" );  
             break;  
  case 'j' : System.out.println(" ja" );  
             break;  
  case 'n' : System.out.println(" nein" );  
             break;  
  default: System.out.println(" alles andere" );  
             break;  
}
```

break in while-Schleife

```
int i = 0;

while (i < 10 ) {
    System.out.println("i = " + i);
    if( i == 5 )
        break;
    i++;
}
```

continue in while-Schleife

```
int i = 0;

while (i < 10 ) {
    i++;
    if( i == 5 )
        continue;
    System.out.println("i = " + i);
}
```


continue in for-Schleife

```
for (int i = 0; i < 10; i++ ){  
    if( i == 5 )  
        continue;  
    System.out.println("i = " + i);  
}
```

Lernziele

- Was gibt es für Kontrollstrukturen in Java ?
- Was ist eine Sequenz?
- Wie funktionieren Schleifen mit *while*, *do...while*, *for* ?
- Wie verzweigt man mit *if / else* ?
- Wann werden switch/case – Statements genutzt?
- Wie kann Programmcode unterbrochen und weitergeführt werden?



**Hochschule für Technik
und Wirtschaft Berlin**

University of Applied Sciences

www.htw-berlin.de