

Kontrollstrukturen - Universität Köln

Mario Manno

Was sind Sprachen

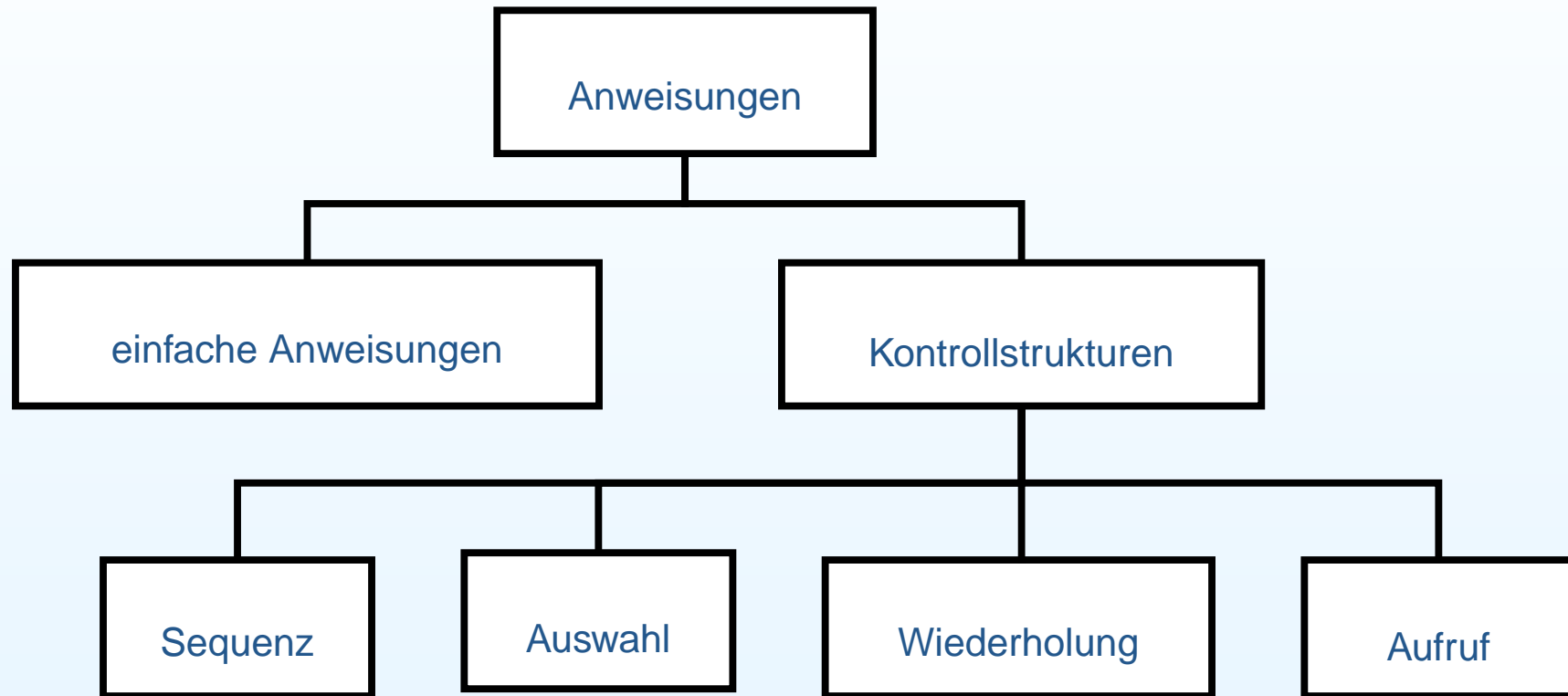
- Auszeichnungssprachen
HTML, XML
- Programmiersprachen
ASM, Basic, C, C++, Haskell, Java, Pascal, Perl, Python, Ruby, XSLT
- Formale Sprache zur Darstellung von Programmen, Algorithmen.
- Unterscheidung in Daten und Kode.

Bestandteile von Programmiersprachen 1

- Ausdrücke
- Bedingungen
a == b
- Zuweisungen
a = 1
- Operatoren
+ - / * ...

-> Anweisungen

Bestandteile von Programmiersprachen 2



Interaktion: Welche Kontrollstrukturen gibt es?

Interaktion: Welche Kontrollstrukturen gibt es?

- Blöcke
- if
- case
- while
- repeat until / do while
- loop
- for
- foreach
- ...

Nassi Shneiderman Struktogramme

- Dienen der Darstellung von Anweisungen.
- Programme lassen sich in Nassi Struktogrammen planen.
- Alternativen:
 - Pseudo-Code
 - Flussdiagramme
 - UML

Sequenz

- Eine Reihe von Anweisungen.
- Anweisungen werden hintereinander, sequentiell, ausgeführt.
- Reihenfolge: oben nach unten, links nach rechts.

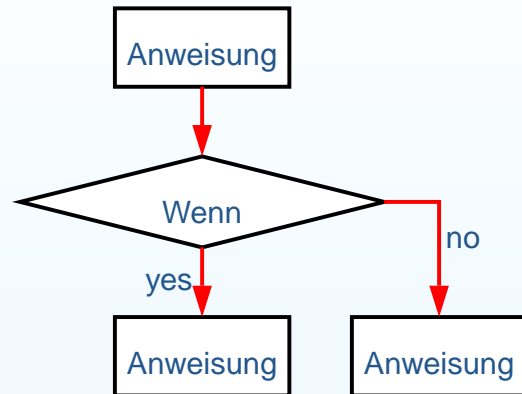
Nassi-Shneiderman — Sequenz, Anweisungs Block

Anweisung
Anweisung

Sequenz - Beispiel

- a=1;
b=2;
print a+b;
- c=3;d=4;print c+d;

Auswahl



- Auch Fallunterscheidung, Verzweigung genannt.
- Ausführung in Abhängigkeit einer Bedingung.
- Ein-, Zwei-, Mehrseitige Auswahl.
- Beispiele: If, Case Anweisung

Auswahl - IF Anweisung

- Häufig auch als mehrseitige Fallunterscheidung anwendbar.
- Zu stark geschachtelte IF Anweisungen sind zu vermeiden.
- Die Logische Bedingung kann äusserst komplex werden, durch Einsatz Boolescher Operatoren.

Nassi-Shneiderman — If-Verzweigung, zweiseitig

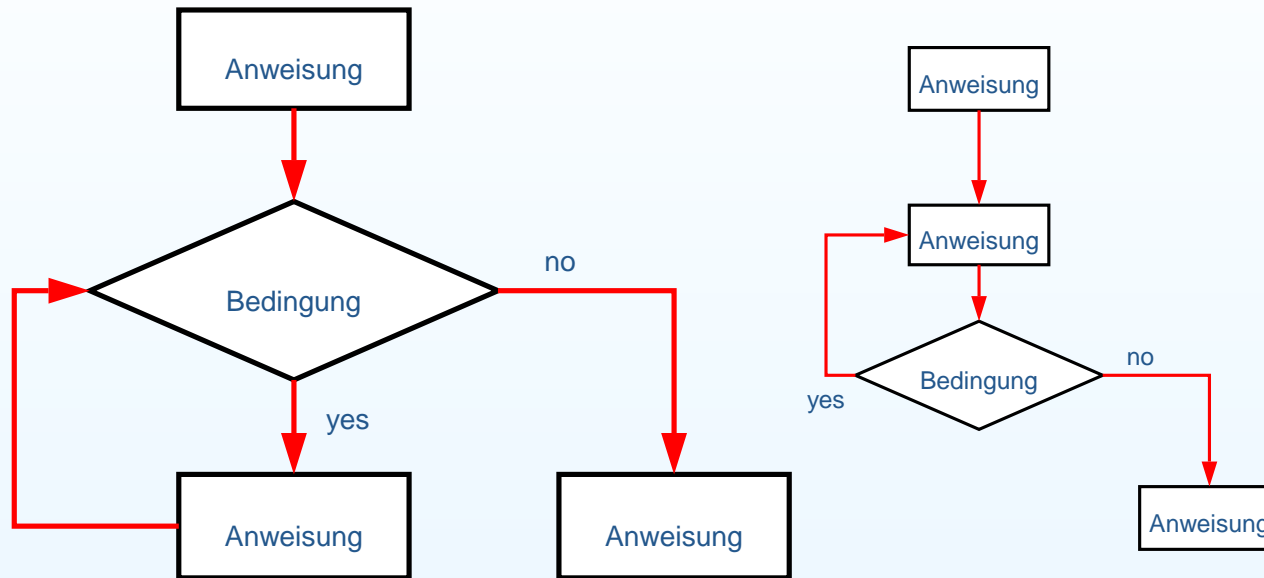


Auswahl - CASE Anweisung

Nassi-Shneiderman — Case-Anweisung

case item		
condition 1	condition 2	condition 3
statement to do	statement 1 to do	statement to do
	statement 2 to do	

Wiederholung / Iteration



- Wiederholung von Anweisungen, meist in Abhängigkeit von logischer Bedingung
- Abfrage vor Wiederholung: “Head controlled“
- Abfrage nach Wiederholung: “Tail controlled“
- Wiederholung mit festgelegter Anzahl von Durchläufen
- Endlosschleife

Wiederholung - FOR Schleife

- Die FOR Schleife enthält meistens eine sogenannte Laufvariable.
- Zählschleife
- Besondere Art der FOR Schleife ist die FOREACH Schleife, sie iteriert über die einzelnen Elemente einer Liste.

Nassi-Shneiderman — FOR-Schleife

for i=n bis i gleich max

Schleifen-Block

Wiederholung - FOR Schleife - Beispiel

Perl Beispiel

```
for ( $i=1;$i<100;$i++) {  
    print $i;  
}
```

Wiederholung - WHILE Schleife

- Die WHILE Schleife ermöglicht eine bessere Kontrolle der Laufvariable.
- Häufig ist hier auch die logische Abbruchbedingung komplexer.
- Wird die Abbruchbedingung geschickt gewählt steigt das Programm u.U. gar nicht in die While Schleife ein.

Nassi-Shneiderman — While(Do)-Schleife



Wiederholung - WHILE Schleife - Beispiel

```
$i=1;
while ($i < 100) {
    print $i;
    $i++;
}
```

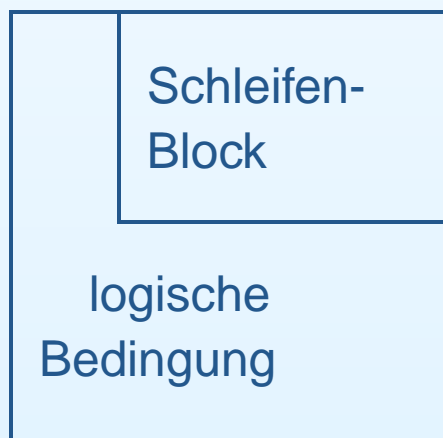
```
while (1) {
    print "Python ist doof.\n";
}
```

```
$dateiname='tmp';
$i=1;
while ( -e $dateiname.$i ) {
    $i+=1;
}
```

Wiederholung - REPEAT/DO Schleife

- Diese Schleife überlässt, wie die WHILE Schleife die Verwaltung der Laufvariable dem Programmierer.
- Jedoch steht die Abbruchbedingung am Ende der Schleife.
- Es gibt also mindestens einen Durchlauf.

Nassi-Shneiderman — Repeat (Do)-Until (While)-Schleife



Wiederholung - REPEAT/DO Schleife - Beispiel

Perl Beispiel

```
$i=0;  
do {  
    print $i;  
    $i++;  
} while ($i<100)
```

Aufruf

- Anwendung einer Operation innerhalb einer anderen Operation
- Möglich durch Aufruf des Operationsnamens und Übergabe der Parameter in der Schnittstelle
- Gibt Daten durch die Schnittstelle zurück oder durch einen Rückgabewert oder gar nicht
- In Java Sprech: Senden einer Botschaft

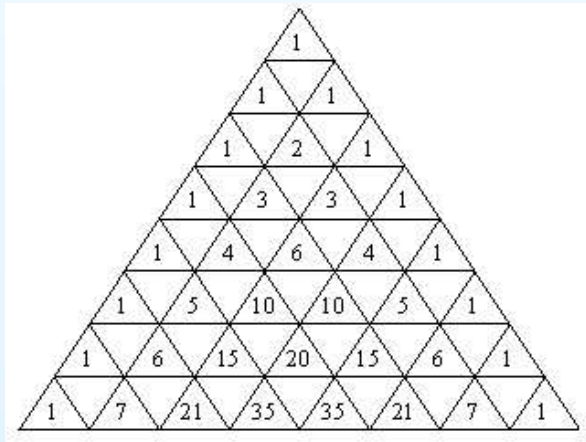
Aufruf - Funktionen und Prozeduren

Historische Unterscheidung in Funktionen und Prozeduren.

- Funktionen liefern Rückgabewert.
- Prozeduren geben bei “Call-by-Reference“ durch die Schnittstelle Daten zurück.
- “Call-by-Value“ ändert die Schnittstellen Werte nicht.
- Heute sind Mischformen üblich, jedoch zu Lasten der Lesbarkeit.

Aufruf - Rekursion

- Selbstbezüglichkeit
- Mathematische Rekursion, Pascalsches Dreieck:
 $C(n,k)=C(n-1,k-1)+C(n-1,k)$
- Wenn eine Funktion sich selbst aufruft spricht man von Rekursion.
- Jede rekursive Funktion hat eine Abbruchbedingung und einen rekursiven Aufruf.
- In der funktionalen Programmierung ersetzt die Rekursion die Schleife.



Dank und Literatur

Vielen Dank !

Literatur:

- Helmut Balzert: “Lehrbuch Grundlagen der Informatik“
- Nassi-Shneiderman:
<http://www.educat.hu-berlin.de/voss/LaTeX/layouts/nassi.phtml>
- Sonstiges: <http://de.wikipedia.org/> , <http://google.com>

Software:

- vim
- tex
- dia