

Nassi-Shneiderman-Diagramm

1

STRUKTOGRAMME

28.06.2018

Inhalt

2

- Sequenz
- Schleife
- Iteration
- Verzweigung
- Auswahl

28.06.2018

Sequenz



Sequenz

- Eine Sequenz ist die Aneinanderreihung einzelner Anweisungen.
- Diese werden hintereinander abgearbeitet.
- Abhängig vom Grad der Modularisierung und der später verwendeten Programmiersprache kann ein Element einer Sequenz mehrere Befehle repräsentieren.

3

Sequenz



Sequenz

Nassi-Shneiderman-Notation

Anweisung 1
Anweisung 2
...
Anweisung n

4

Sequenz



Sequenz

Beispiel:

<code>strText = "Hallo Würzburg"</code>
<code>strText</code> ausgeben

```
string strText;  
  
strText = "Hallo Würzburg";  
console.WriteLine(strText);
```

5

Sequenz



Sequenz

Beispiel:

<code>x = 17</code>
<code>y = 4</code>
<code>e = x+y</code>
<code>e</code> ausgeben

```
int x,y,e;  
  
x = 17;  
y = 4;  
  
e = x+y;  
  
console.WriteLine(e);
```

6

Schleife



Schleife

- Mithilfe einer Schleife kann eine Sequenz mehrfach durchlaufen werden.
- Die Anzahl der Durchläufe wird durch eine Bedingung bestimmt.
- Man unterscheidet kopfgesteuerte und fußgesteuerte Schleifen.

7

Schleife



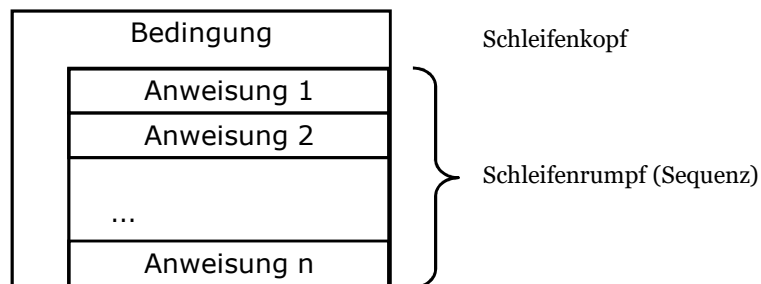
Kopfgesteuerte Schleife

- Bei einer kopfgesteuerten Schleife befindet sich die Bedingung, die zum Durchlaufen der Schleife erfüllt sein muss, vor dem Rumpf, der die Sequenz enthält.
- Wenn die Bedingung beim ersten Aufruf der Schleife nicht erfüllt ist, wird die Sequenz in der Schleife **nie** ausgeführt.
- *"Solange die Bedingung erfüllt ist, tue dies."*

8

Schleife

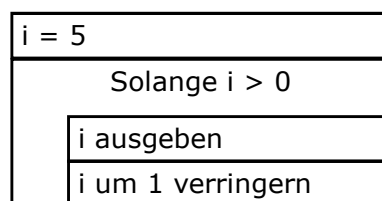
Kopfgesteuerte Schleife Nassi-Shneiderman-Notation



9

Schleife

Kopfgesteuerte Schleife Beispiel



```
int i;
i = 5;

while(i>0) {
    console.WriteLine(i+"\n");
    i = i-1;
}
```

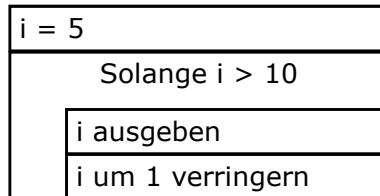
```
5
4
3
2
1
```

10

Schleife

Kopfgesteuerte Schleife

Beispiel



diese kopfgesteuerte Schleife wird nie durchlaufen, da beim ersten Aufruf die Bedingung nicht erfüllt ist!

```
int i;
i = 5;

while(i>10) {
    console.WriteLine(i+"\n");
    i = i-1;
}
```

11

Schleife

Fußgesteuerte Schleife

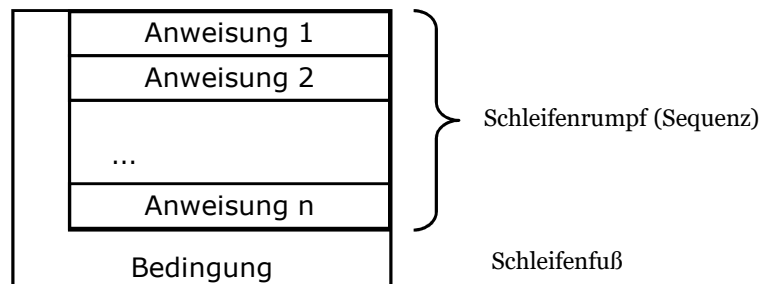
- Bei einer fußgesteuerten Schleife befindet sich die Bedingung, die zum Durchlaufen der Schleife erfüllt sein muss, nach dem Rumpf, der die Sequenz enthält.
- Wenn die Bedingung beim ersten Aufruf der Schleife nicht erfüllt ist, wird die Sequenz in der Schleife trotzdem **mindestens einmal** ausgeführt.
- *"Tue dies. Solange die Bedingung erfüllt ist, tue es nochmal."*

12

Schleife

Fußgesteuerte Schleife

Nassi-Shneiderman-Notation

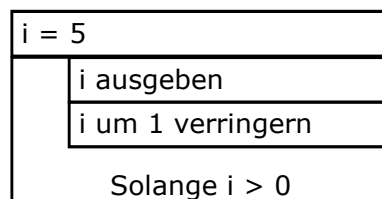


13

Schleife

Fußgesteuerte Schleife

Beispiel



```
int i;
i = 5;

do {
    console.WriteLine(i+"\n");
    --i;
} while(i>0)
```

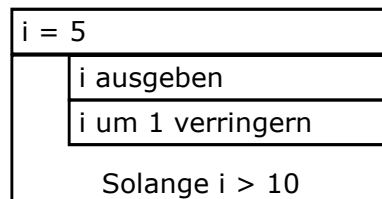
```
5
4
3
2
1
```

14

Schleife

Fußgesteuerte Schleife

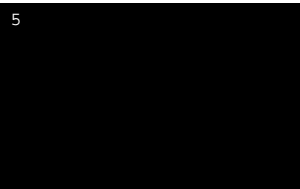
Beispiel



diese fußgesteuerte Schleife wird mindestens einmal durchlaufen, unabhängig davon, ob die Bedingung beim ersten Aufruf erfüllt ist oder nicht!

```
int i;
i = 5;

do {
    console.WriteLine (i+"\n");
    i--;
} while(i>10)
```



15

Iteration

Iteration

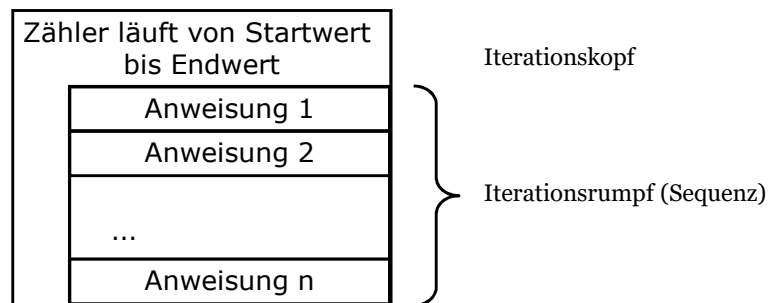
- Die Iteration ist eine Sonderform der kopfgesteuerten Schleife.
- Im Kopf der Iteration ist im Vorhinein festgelegt, wie oft sie aufgerufen werden soll.
- Die Iteration wird häufig auch als For-Schleife bezeichnet.
- *"Lasse den Zähler in festen Schritten von Startwert bis Endwert laufen."*

16

Iteration

Iteration (For-Schleife)

Nassi-Shneiderman-Notation

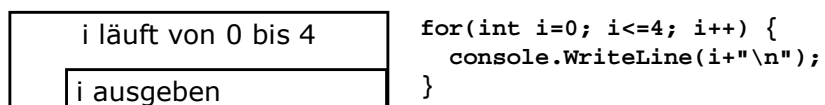


17

Iteration

Iteration (For-Schleife)

Beispiel



```
for(int i=0; i<=4; i++) {
    console.WriteLine(i+"\n");
}
```

```
0
1
2
3
4
```

18

Iteration

Iteration (For-Schleife)

Beispiel

x = 5

i läuft von 1 bis x

i ausgeben

```
int x;
```

```
x = 5;
```

```
for(int i=1; i<=x; i++) {  
    console.WriteLine(i+"\n");  
}
```

```
1  
2  
3  
4  
5
```

19

Verzweigung

Verzweigung

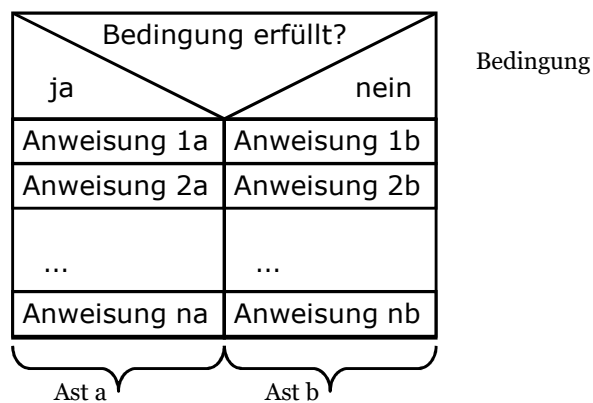
- Eine Verzweigung besteht aus einer Bedingung und zwei alternativen Programmästen.
- Ist die Bedingung erfüllt, wird nur der erste Programmast abgearbeitet.
- Wenn die Bedingung nicht erfüllt ist, wird nur der zweite Ast abgearbeitet.

20

Verzweigung

Verzweigung

Nassi-Shneiderman-Notation

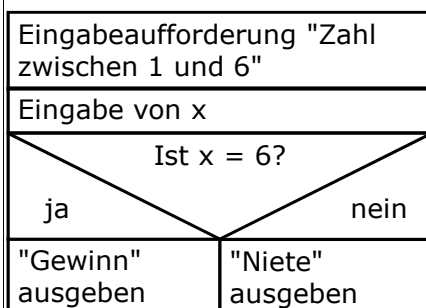


21

Verzweigung

Verzweigung

Beispiel



```
int zahl;
console.WriteLine("Geben Sie eine
Zahl zwischen 1 und 6 ein");
zahl = console.Read();
if (zahl == 6) {
    console.WriteLine("Gewinn");
}
else {
    console.WriteLine("Niete");
}
```

22

Auswahl

Auswahl

- Eine Auswahl wird verwendet, wenn für die möglichen Werte, die eine Variable annehmen kann, jeweils eine eigene Sequenz ausgeführt werden soll.
- Die Auswahl ersetzt damit mehrere verschachtelte Verzweigungen.
- Die Auswahl sollte einen "Default"-Ast besitzen, der ausgeführt wird, wenn die Variable keinen der vorgesehenen Werte annimmt.

23

Auswahl

Auswahl

Hat die Variable den Wert...?			
Wert 1	Wert 2	Wert 3	sonst
Anweisung 1a	Anweisung 1b	Anweisung 1c	Anweisung 1x
Anweisung 2a	Anweisung 2b	Anweisung 2c	Anweisung 2x
...
Anweisung na	Anweisung nb	Anweisung nc	Anweisung nx
Ast a		Ast b	
		Ast c	
		Ast x	

24

Auswahl



Auswahl

Beispiel

Eingabeaufforderung "Geschlecht (m/w)?"		
Eingabe von Geschlecht		
Hat Geschlecht den Wert...?		
"m"	"w"	sonst
"Werter Herr," ausgeben	"Werte Dame," ausgeben	"Hallo," ausgeben
"wir freuen uns, Sie zu sehen!" ausgeben		

25

Auswahl



Auswahl

Beispiel

```
char Geschlecht;
Console.WriteLine("Geschlecht (m/w)?");
Geschlecht = console.ReadLine()

switch (Geschlecht) {
    case "m" : console.WriteLine("Werter Herr, \n");
               break;
    case "w" : console.WriteLine("Werte Dame, \n");
               break;
    default : console.WriteLine("Hallo, \n");
}

console.WriteLine("wir freuen uns, Sie zu sehen!");
```

26

Übung „Nassi-Shneiderman-Diagramm“

27

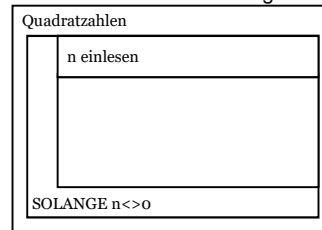
- Vervollständigen Sie das angegebene Nassi-Shneiderman-Diagramm, welches in einer (äußeren) Schleife Ganzzahlen in eine Variable **n** einliest. Die Reaktion des Programms soll davon abhängen, ob der in die Variable eingelesene Wert positiv, negativ oder gleich Null ist. Treffen Sie folgende Teilunterscheidungen:

- Ist die eingegebene Zahl **n** größer als Null, so soll in einer inneren Schleife folgende Ausgabe erzeugt werden:

Zahl	Quadratzahl
1	1
2	4
...	...
n	n*n

Die Anzeige aller Quadratzahlen von 1 bis n.

- Ist die eingelesene Zahl **n** kleiner als Null, so soll ausgegeben werden: „negative Zahl“
- Ist die eingegebene Zahl **n** gleich Null, so soll das Programm (die äußere Schleife) beendet werden.

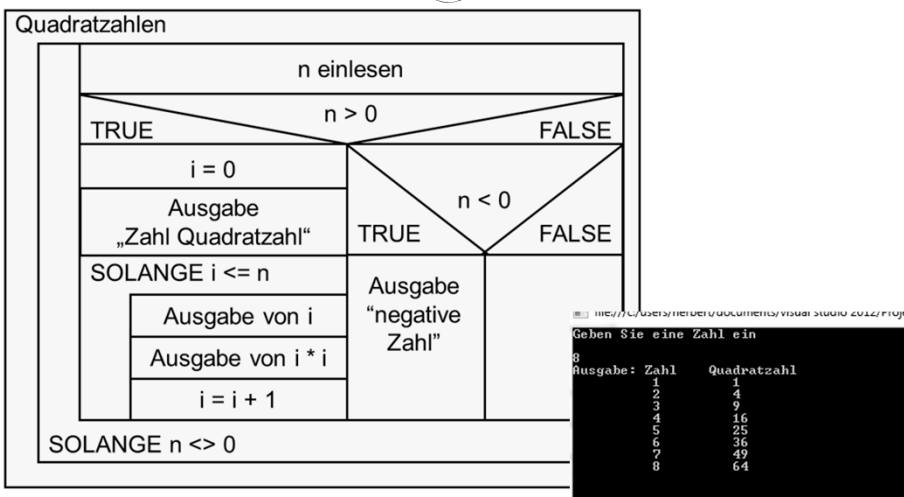


Herbert Pichler, BSc

28.06.2018

Lösung

28



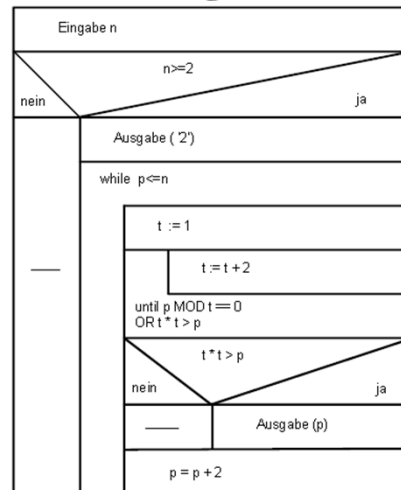
Herbert Pichler, BSc

28.06.2018

Übung „Nassi-Shneiderman-Diagramm“

29

- $p = 3$
- $n = \text{Eingabe}$
- $t = 0$



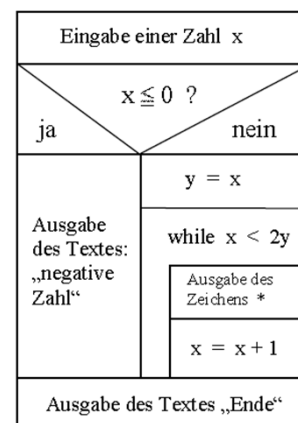
Herbert Pichler, BSc

28.06.2018

Übung „Nassi-Shneiderman-Diagramm“



- Gib genau an, was durch das nebenstehende Struktogramm bewirkt wird.
- Welche Ausgaben erhält man, wenn man für x die Werte 0 bzw. 2 bzw. 8 eingibt?



Funktionen



Funktionen

- Dem Konzept der Modularisierung wird über Funktionen (manchmal auch Prozeduren oder Unterprogramme genannt) Rechnung getragen.
- Befehlsfolgen, die häufiger verwendet werden, werden dabei in Funktionen verpackt.
- Ein Aufruf der Funktion hat zur Folge, dass die Befehle in der Funktion abgearbeitet werden.

31

Funktionen



Funktionen

- Eine Funktionsdeklaration besteht mindestens aus folgenden zwei Teilen:
 - dem Funktionskopf: hier werden der Name der Funktion und die zu übergebenden Parameter festgelegt.
 - dem Funktionsrumpf: hier stehen die Befehle, die abgearbeitet sind.
- Optional enthält eine Funktion im Rumpf noch eine Anweisung, um einen Wert zurück zu geben.

32

Funktionen

Funktionen

- Eine Funktion wird aufgerufen, indem man ihren Namen wie einen Befehl verwendet.
- Gegebenenfalls müssen diesem Befehl noch Werte als Parameter mitgegeben werden.

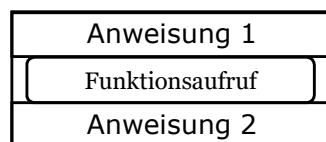
33

Funktionen

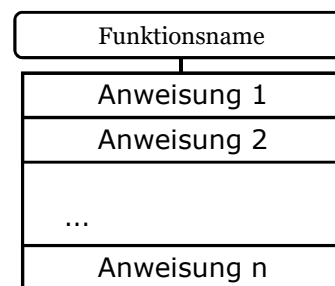
Funktionen

Nassi-Shneiderman-Notation

Funktionsaufruf



Funktionsdeklaration



34

Funktionen

Funktionen

Beispiel ohne Rückgabe

Funktionsaufruf

Anrede ausgeben
Mahntext
"Würzburg, " und Datum ausgeben

Funktionsdeklaration

Mahntext
"in der Hektik des..." ausgeben
...
"Wir sehen uns sonst gezwungen..." ausgeben

35

Funktionen

Funktionen

Beispiel ohne Rückgabe

Funktionsaufruf

Anrede ausgeben
Mahntext
"Würzburg, " und Datum ausgeben

```
string Anrede;

Anrede = "Sehr geehrter Herr...";

console.WriteLine(Anrede+"<br>");
Mahntext();
console.WriteLine("Würzburg, "+Date());
```

36

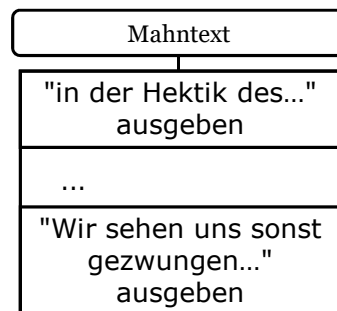
Funktionen



Funktionen

Beispiel ohne Rückgabe

Funktionsdeklaration



```
function Mahntext() {
  document.write("in der Hektik des");
  document.write("...<br>");
  // ...
  document.write("Wir sehen uns ");
  document.write("sonst gezwungen");
  document.write("...<br>");
}
```

37

Funktionen



Funktionen

```
function Mahntext() {
  document.write("in der Hektik des");
  document.write("...<br>");
  // ...
  document.write("Wir sehen uns ");
  document.write("sonst gezwungen");
  document.write("...<br>");
}
```

```
var Anrede;
```

```
Anrede = "Sehr geehrter Herr...";
```

```
document.write(Anrede+"<br>");
```

```
Mahntext();
```

```
document.write("Würzburg, "+Date());
```

38

Funktionen



Funktionen

- Wie bereits erwähnt, können Funktionen auch einen Wert zurück geben.
- In diesem Fall werden sie wie eine Zahl oder eine Variable auf der rechten Seite des Zuweisungsoperators (=) geschrieben.

39

Funktionen

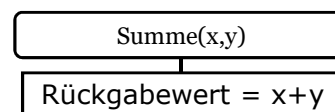
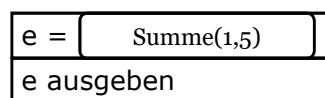


Funktionen

Beispiel mit Rückgabe und Parametern

Funktionsaufruf

Funktionsdeklaration



40

Funktionen



Funktionen

Beispiel mit Rückgabe und Parametern

Funktionsaufruf

e =	Summe(1,5)
e ausgeben	

var e;

e = Summe(1,5);

document.write(e+"
");

41

Funktionen



Funktionen

Beispiel mit Rückgabe und Parametern

Funktionsdeklaration

Summe(x,y)
Rückgabewert = x+y

```
function Summe(x,y) {
  return x+y;
}
```

42

Funktionen



Funktionen

Ablauf des Beispiels. Schritt 1.

- Funktionsdeklaration wird erkannt und übersprungen.

```
function Summe(x,y) {  
    return x+y;  
}  
  
var e;  
  
e = Summe(1,5);  
  
document.write(e+"<br>");
```

43

Funktionen



Funktionen

Ablauf des Beispiels. Schritt 2.

- Variable e wird deklariert.

```
function Summe(x,y) {  
    return x+y;  
}  
  
var e;  
  
e = Summe(1,5);  
  
document.write(e+"<br>");
```

44

Funktionen



Funktionen

Ablauf des Beispiels. Schritt 3.

- Der Variable e soll der Rückgabewert der Funktion Summe zugewiesen werden.

```
function Summe(x,y) {  
    return x+y;  
}  
  
var e;  
  
e = Summe(1,5);  
  
document.write(e+"<br>");
```

45

Funktionen



Funktionen

Ablauf des Beispiels. Schritt 4.

- Die Funktion Summe wird aufgerufen...

```
function Summe(x,y) {  
    return x+y;  
}  
  
var e;  
  
e = Summe(1,5);  
  
document.write(e+"<br>");
```

46

Funktionen



Funktionen

Ablauf des Beispiels. Schritt 5.

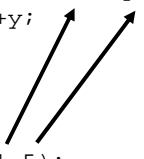
- ...und die Werte der Parameter an die Funktion Summe übergeben.

```
function Summe(x=1,y=5) {
  return x+y;
}

var e;

e = Summe(1,5);

document.write(e+"<br>");
```



47

Funktionen



Funktionen

Ablauf des Beispiels. Schritt 6.

- Die Werte der Parameter werden verwendet, um den Rückgabewert zu berechnen.

```
function Summe(x=1,y=5) {
  return x+y=1+5=6;
}

var e;

e = Summe(1,5);

document.write(e+"<br>");
```

48

Funktionen



Funktionen

Ablauf des Beispiels. Schritt 7.


- Der Rückgabewert wird zurück geliefert und der Variable e zugewiesen.

```
function Summe(x=1,y=5) {
    return x+y=1+5=6;
}

var e;

e = Summe(1,5)=6;

document.write(e+"<br>");
```



49

Funktionen



Funktionen

Ablauf des Beispiels. Schritt 8.

- Der Wert der Variable e wird ausgegeben und einen Zeilenumbruch (
) eingefügt.

```
function Summe(x=1,y=5) {
    return x+y=1+5=6;
}

var e;

e = Summe(1,5)=6;

document.write(e+"<br>");
```

50