

Entwicklung und grafische Darstellung von Programmabläufen

10.01.2005

Hochschule für angewandte Wissenschaften Hamburg
Prof.Dr.Ing.U.Vogelsang, Prof.Dr.U.Sauvagerd



Programmieren in ANSI C:
K6_C-grafische-Darstellung.ppt, Seite 1

Entwicklung / Darstellung von Programmabläufen

- In diesem Kapitel wollen wir zeigen,

- ➡ dass man den Ablauf von Programmen grafisch darstellen kann
- ➡ auf welche unterschiedliche Art man Programmabläufe grafisch darstellen kann
- ➡ wie man **Struktogramme** erstellt
- ➡ wie man **Programmablaufpläne** erstellt.

10.01.2005

Hochschule für angewandte Wissenschaften Hamburg
Prof.Dr.Ing.U.Vogelsang, Prof.Dr.U.Sauvagerd



Programmieren in ANSI C:
K6_C-grafische-Darstellung.ppt, Seite 2

Struktogramme und Programmablaufpläne

Programmablaufpläne

sind in *DIN 6601 Informationsverarbeitung Sinnbilder und ihre Anwendung* genormt.

Struktogramme

sind in *DIN 66261 Informationsverarbeitung Sinnbilder für Struktogramme nach Nassi-Shneiderman* genormt.

Die Bibliothek der HAW ist Hamburger Auslegestelle für
DIN-Normen.

Jede Norm ist dort verfügbar und darf von Studenten
eingesehen werden.

10.01.2005



Grafische Darstellung von Programmabläufen (1)

- Die grafische Darstellung von Programmabläufen ist historisch gewachsen.
 - Zunächst war man bemüht, die **unübersichtlichen Programme** in den **Maschinensprachen (Assembler)** der Prozessoren anschaulicher darzustellen.

Deshalb wurden
➡ **Programmablaufpläne** zunächst
entwickelt, um die **vorab erstellten**
➡ **Assemblerprogramme**
zu dokumentieren.

10.01.2005



Abstrakte und programmnahe Darstellung (1)

- Bei der Darstellung von komplexen Programmen sollte
 - ➔ die **Zuordnung** der grafischen Symbole zu den **Befehlen**
 - ➔ der **Sinn** der **Befehle** deutlich werden.
- Bei **Programmablaufplänen** kann man dies erreichen, indem man
 - ➔ die **Symbole programmnahe** schreibt und
 - ➔ in den **Kommentaren den Sinn der Befehle** beschreibt.

10.01.2005



Abstrakte und programmnahe Darstellung (2)

- Bei der Darstellung von komplexen Programmen sollte
 - ➔ die **Zuordnung** der grafischen Symbole zu den **Befehlen**
 - ➔ der **Sinn** der **Befehle** deutlich werden.
- Bei **Struktogrammen** sind **keine Kommentare vorgesehen**.
Deshalb muss man sich für **eine der Darstellungsarten** entscheiden.
- Bei **C-Programmen** ist meistens die **abstraktere Darstellung** - Sinn der Befehle - zweckmäßiger.

10.01.2005



Abstrakte und programmnahe Darstellung (3)

- Beispiel für
➔ **abstrakte Programmdarstellung** im Struktogramm
(gleitender Mittelwert)

- C-Ausschnitt

```
...
{
  m1 = (long double) mean * Anzahl;
  Anzahl++;
  mean = (m1 + x) / Anzahl;
}
```

- Struktogramm

alten Mittelwert korrigieren und als long double auswerten
Anzahl der Messwerte erhöhen
neuen Mittelwert berechnen

10.01.2005



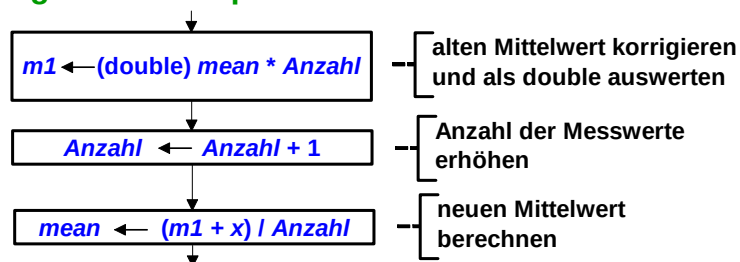
Abstrakte und programmnahe Darstellung (4)

- Beispiel für
➔ **programmnahe Darstellung** mit
abstraktem Kommentar (gleitender Mittelwert)

- C-Ausschnitt:

```
...
{
  m1 = (double) mean * Anzahl;
  Anzahl++;
  mean = (m1 + x) / Anzahl;
}
```

- Programmablaufplan:



10.01.2005



Programmablaufpläne

- Programmablaufpläne sind geeignet, **relativ kleine Programme** zu entwickeln.
- Sie erzwingen aber leider keine **übersichtliche Programmstruktur**.



Vorteile

- leicht veränderbar
- programmnahe Darstellung in den Symbolen
- abstrakte Darstellung im Kommentar



Nachteile

- erzwingen keine Module
- ermöglichen schlecht aufteilbare Programme

- Bei disziplinierter Programmierweise kann man auch mit Programmablaufplänen modulare Programme entwickeln.

10.01.2005

Hochschule für angewandte Wissenschaften Hamburg
Prof.Dr.Ing.U.Vogelsang, Prof.Dr.U.Sauvagerd



Programmieren in ANSI C:
K6_C-grafische-Darstellung.ppt, Seite 9

Struktogramme

- Eine modernere Darstellung von Programmabläufen sind **Struktogramme**.
- Sie erzwingen eine **modulare Programmstruktur**.



Vorteile

- modulare Programme
- aufteilbare Software



Nachteile

- Struktogramm ist schwierig zu korrigieren
- nur abstrakte Darstellung sinnvoll

10.01.2005

Hochschule für angewandte Wissenschaften Hamburg
Prof.Dr.Ing.U.Vogelsang, Prof.Dr.U.Sauvagerd



Programmieren in ANSI C:
K6_C-grafische-Darstellung.ppt, Seite 10

Struktogramme & Programmablaufpläne

- Struktogramme und Programmablaufpläne erlauben
➡ **unterschiedliche Programmkonstrukte.**
 - Struktogramme **erzwingen** eine
➡ **modulare Programmstruktur.**
 - Deshalb sind die Grafiken von
Struktogrammen und Programmablaufplänen
➡ **nicht äquivalent:**

Es gibt Konstrukte in Struktogrammen,
die in Programmablaufplänen nicht vorkommen
und umgekehrt.

10.01.2005



Struktogramm- & Programmablaufplan- Symbole (1)

➡ **Befehlsfolge (Verbundanweisung)**

- **Struktogramm**
- **C-Befehle**
- **Programmablaufplan**

Eingabeaufforderung
auf dem Bildschirm
ausgeben: **printf (..)**

Messwert **x** einlesen:
scanf (..)
x aufsummieren

```
...  
{  
    printf("Messwert x= ");  
    scanf("%f",&x);  
    sum += x;  
}  
...
```

Eingabeaufforderung
auf dem Bildschirm
ausgeben: **printf (..)**
Messwert **x** einlesen:
scanf (..)
x aufsummieren

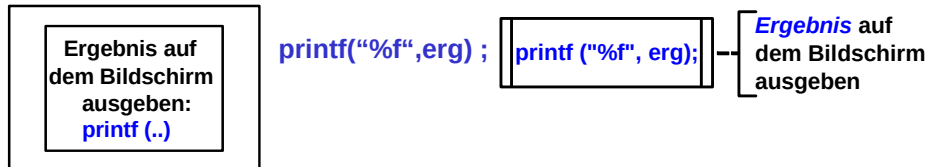
10.01.2005



Struktogramm- & Programmablaufplan-Symbole (2)

➔ Funktion (Unterprogramm)

- Struktogramm
- C-Befehl
- Programmablaufplan



- Die genaue Arbeitsweise eines Unterprogramms (*function*) wird später dargestellt.

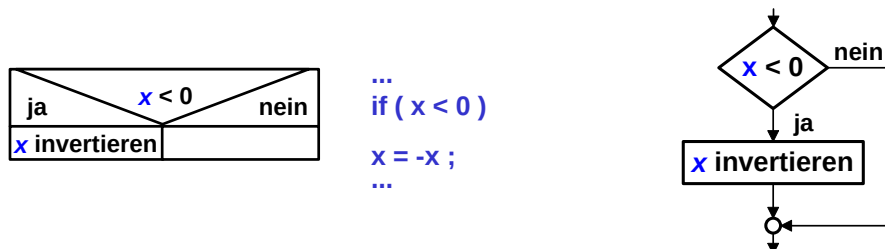
10.01.2005



Struktogramm- & Programmablaufplan-Symbole (3)

➔ Bedingte Verarbeitung

- Struktogramm
- C-Befehle
- Programmablaufplan



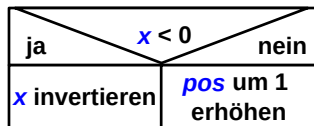
10.01.2005



Struktogramm- & Programmablaufplan-Symbole (4)

➔ Einfache Alternative

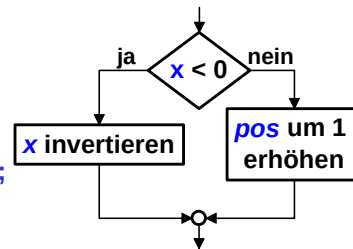
• Struktogramm



• C-Befehle

```
...
if ( x < 0 )
    x = -x ;
else pos += 1 ;
...
```

• Programmablaufplan



10.01.2005



Struktogramm- & Programmablaufplan-Symbole (5)

➔ Schleifen- oder Verzweigungsabbruch

• C-Befehl

```
...
switch ( j )
{ case 0 :
    Geld = 1.95583 * Euro ;
    break ;
...
}
```

• Struktogramm



• Programmablaufplan



10.01.2005



Struktogramm-Symbole (1)

➔ Mehrfache Alternative

• Struktogramm

• switch (...)

... case ...:

• C-Programm

```
enum Waehrung { D-Mark, F-Franc, Lira } ;
...
switch (Waehrung)
{ case 0:
  Geld = 1.95583 * Euro ;
  printf ("%6.2f Euro = %6.2f DM", Euro, Geld ) ; break ;
  case 1:
  Geld = 6.55957 * Euro ;
  printf (" %7.2f Euro = %8.2f FF ", Euro, Geld) ; break ;
  case 2:
  Geld = 1939.27 * Euro ;
  printf (" %7.2f Euro = %12.0f itl ", Euro, Geld) ; break ;
  default:
  printf ("Fehler bei der Umrechnung! " ) ; }
```

Waehrung			
D-Mark	F-Franc	Lira	sonst
Geld = 1.95583 * Euro	Geld = 6.55957 * Euro	Geld = 1939.27 * Euro	Fehler- meldung

10.01.2005

Hochschule für angewandte Wissenschaften Hamburg
Prof.Dr.Ing.U.Vogelsang, Prof.Dr.U.Sauvagerd

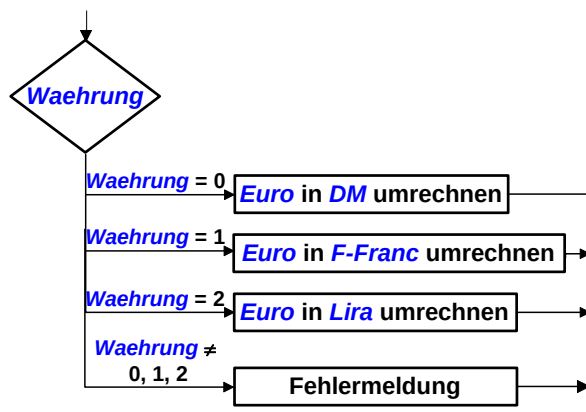


Programmieren in ANSI C:
K6_C-grafische-Darstellung.ppt, Seite 17

Programmablaufplan-Symbole (1)

➔ Mehrfache Alternative

• Abstrakter Programmablaufplan



10.01.2005

Hochschule für angewandte Wissenschaften Hamburg
Prof.Dr.Ing.U.Vogelsang, Prof.Dr.U.Sauvagerd



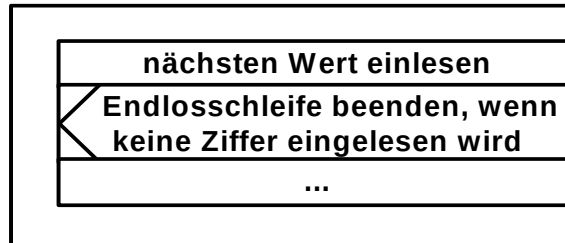
Programmieren in ANSI C:
K6_C-grafische-Darstellung.ppt, Seite 18

Struktogramm-Symbole (2)

➔ Wiederholung ohne Präfix und Postfix

• Struktogramm

- while (1)
- oder
- for (;;)



• C-Programm

```
while ( 1 )
{
    ...
    printf ( "Short-Integerzahl i eingeben ( i>=0, i<=99 ): i= „ ) ;
    ok = scanf ( "%hi",&i ) ;
    if ( ok == 1 ) break ;
    ...
}
```

10.01.2005

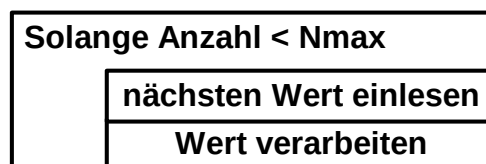


Struktogramm-Symbole (3)

➔ Wiederholung mit Präfix

• Struktogramm

- for
- oder
- while



• C-Programm

```
/* ... */
for ( Anzahl = 1; Anzahl < Nmax; Anzahl++ )
{
    printf ( "Long-Integerzahl i eingeben: i=" ) ;
    scanf ( "%li",&i ) ;
    max = ( i > max ) ? i : max ;
}
/* ... */
```

10.01.2005



Struktogramm-Symbole (4)

➡ Wiederholung mit Postfix

- **Struktogramm**
 - do ... while
- **C-Programm**

nächsten Wert einlesen
Wert verarbeiten
solange <code>in != 'j'</code> oder <code>in != 'n'</code>

```

/* ... */
do
{
    printf ( " Ist z eine ganze Zahl ?: [ j/n ] " ) ;
    scanf ( "%c",&in ) ;
    /* ... */
}
while ( (in != 'j') || (in != 'n') ) ;
/* ... */
    
```

Hochschule für angewandte Wissenschaften Hamburg
Prof.Dr.Ing.U.Vogelsang, Prof.Dr.U.Sauvagerd

Programmieren in ANSI C:
 K6_C-grafische-Darstellung.ppt, Seite 21

10.01.2005

Weitere Programmablaufplan-Symbole (1)

➡ Grenzstellen

- **Programmablaufplan**

```

graph TD
    A([main]) --> B[...]
    B --> C([return 0])
    
```

- **C-Programm**

```

/* ... */
int main (void)
{
    /* ... */
    printf("Herzlich Willkommen im C-Kurs! \n"); /* Ausgabe: Herz... */
    /* ... */
    return 0 ;
}
    
```

/ Vereinbaren eines Hauptprogramms */*

/ Anfang des Hauptprogramms */*

/ Rueckgabewert des Hauptprogramms */*

/ Ende des Hauptprogramms */*

Hochschule für angewandte Wissenschaften Hamburg
Prof.Dr.Ing.U.Vogelsang, Prof.Dr.U.Sauvagerd

Programmieren in ANSI C:
 K6_C-grafische-Darstellung.ppt, Seite 22

10.01.2005

Weitere Programmablaufplan-Symbole (2)

➡ **Display-Anzeige**

- **Programmablaufplan**



↓

- **C-Programm**

```
/* ... */  
printf ("Herzlich Willkommen im C-Kurs! \n");  
/* ... */
```

10.01.2005

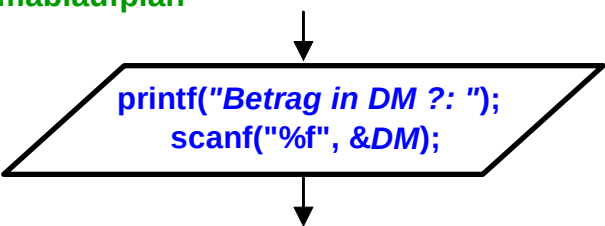
Hochschule für angewandte Wissenschaften Hamburg
Prof.Dr.Ing.U.Vogelsang, Prof.Dr.U.Sauvagerd

Programmieren in ANSI C:
K6_C-grafische-Darstellung.ppt, Seite 23

Weitere Programmablaufplan-Symbole (3)

➡ **Eingabe, Ausgabe**

- **Programmablaufplan**



↓

- **C-Programm**

```
/* ... */  
printf ( "Betrag in DM ?: " ) ; scanf ( "%f", &DM ) ;  
/* ... */
```

10.01.2005

Hochschule für angewandte Wissenschaften Hamburg
Prof.Dr.Ing.U.Vogelsang, Prof.Dr.U.Sauvagerd

Programmieren in ANSI C:
K6_C-grafische-Darstellung.ppt, Seite 24

Struktogramm-Ersatzsymbole (1)

➡ **Alternative**

- Einige Struktogrammsymbole sind grafisch schlecht darstellbar.
- Deshalb sind auch ➡ **Ersatzsymbole** zulässig.

• **Struktogrammsymbol**

Währung			
D-Mark	F-Franc	Lira	sonst
Geld = 1.95583 * Euro	Geld = 6.55957 * Euro	Geld = 1939.27 * Euro	Fehler- meldung

• **Ersatzsymbol**

Währung			
D-Mark	F-Franc	Lira	sonst
Geld = 1.95583 * Euro	Geld = 6.55957 * Euro	Geld = 1939.27 * Euro	Fehler- meldung

• **C-Befehl**

switch (...)

... case ... :

Hochschule für angewandte Wissenschaften Hamburg
Prof.Dr.Ing.U.Vogelsang, Prof.Dr.U.Sauvagerd



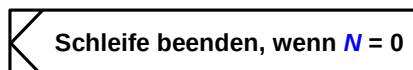
Programmieren in ANSI C:
K6_C-grafische-Darstellung.ppt, Seite 25

10.01.2005

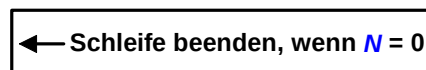
Struktogramm-Ersatzsymbole (2)

➡ **Abbruchanweisung**

• **Struktogrammsymbol**



• **Ersatzsymbol**



• **C-Befehl**

break ;

Hochschule für angewandte Wissenschaften Hamburg
Prof.Dr.Ing.U.Vogelsang, Prof.Dr.U.Sauvagerd



Programmieren in ANSI C:
K6_C-grafische-Darstellung.ppt, Seite 26

10.01.2005

Programmanalyse & grafische Darstellung (1)

- Untersuchen Sie die folgenden 2 Programme!
- Zeichnen Sie jeweils das Struktogramm!
- Geben Sie dazu in der richtigen Reihenfolge den Wert oder die Werte an, die das jeweilige Programm ausgibt. Aus Ihrer Lösung soll erkennbar sein, wie Sie zu Ihren Werten kommen.



Programmanalyse & grafische Darstellung (2)

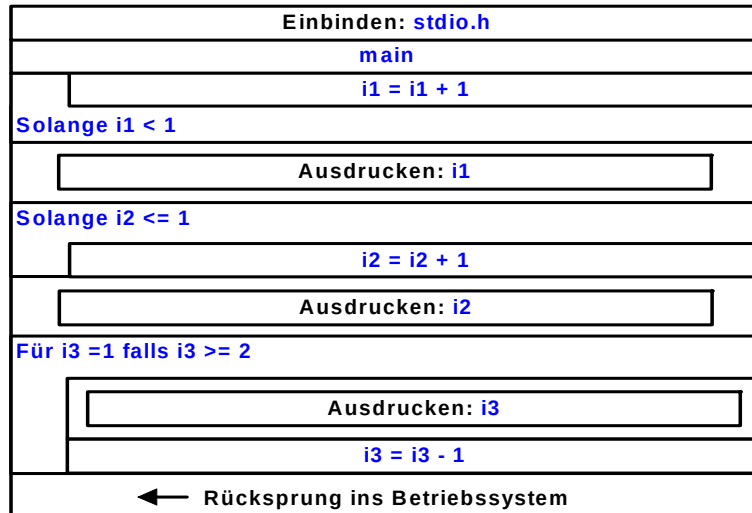
Programm 1:

```
#include <stdio.h>
int main (void)
{
    int i1 = 1, i2 = 1, i3 ;
    do    i1++ ;
    while ( i1 < 1 ) ;
    printf ( "%d\n", i1 ) ;
    while ( i2 <= 1 )
        i2++ ;
    printf ( "%d\n", i2 ) ;
    for ( i3 = 1; i3 >= 2; i3-- )
        printf ( "%d\n", i3 ) ;
    return 0 ;
}
```



✓ Programmanalyse & grafische Darstellung (1)

Programm 1: Struktogramm



✓ Programmanalyse & grafische Darstellung (2)

Programm 1:

• Programmabschnitt

```

int i1 = 1 ; /* ... */
do      i1++ ;
while ( i1 < 1 );
printf ("%d\n", i1) ;
  
```

• Programmabschnitt

```

int i2 = 1 ; /* ... */
/* ... */
while ( i2 <= 1 )
    i2++;
printf ("%d\n", i2);
  
```

• Programmverlauf

```

i1 = 1
i1 = 2
i1 >= 1 ⇒ Schleifenabbruch
Ausgabe: 2
  
```

• Programmverlauf

```

i2 = 1
1. Durchlauf:
i2 <= 1
i2 = 2
Ausgabe: 2

2. Durchlauf:
i2 > 1 ⇒
Abbruch
  
```

✓ Programmanalyse & grafische Darstellung (3)

Programm 1:

- Programmabschnitt
- Programmverlauf

```
for ( i3 = 1; i3 >= 2; i3-- )  
    printf ( "%d\n", i3 ) ;
```

i3 = 1 i3 < 3
⇒ Schleifenabbruch
keine Ausgabe



✎ Programmanalyse & grafische Darstellung (3)

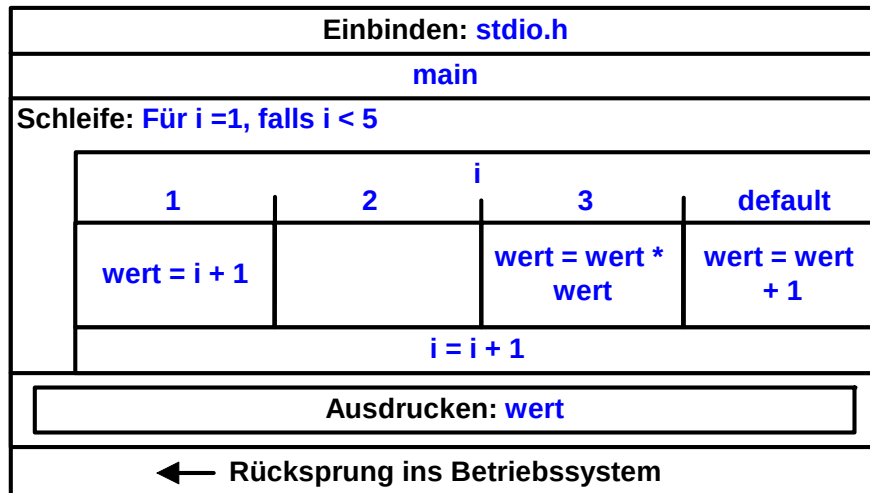
Programm 2:

```
#include <stdio.h>  
  
int main ( void )  
{  
    int i, wert ;  
    for ( i = 1; i < 5; i++ )  
        switch ( i )  
        { case 1: wert = i+1 ;  
          break ;  
          case 2: break ;  
          case 3: wert *= wert ;  
            break ;  
          default: wert += wert ;  
        }  
    printf ( "%d\n", wert ) ;  
    return 0 ;  
}
```



✓ Programmanalyse & grafische Darstellung (4)

Programm 2 : Struktogramm



Hochschule für angewandte Wissenschaften Hamburg
Prof.Dr.Ing.U.Vogelsang, Prof.Dr.U.Sauvagerd

Programmieren in ANSI C:
K6_C-grafische-Darstellung.ppt, Seite 33

✓ Programmanalyse & grafische Darstellung (5)

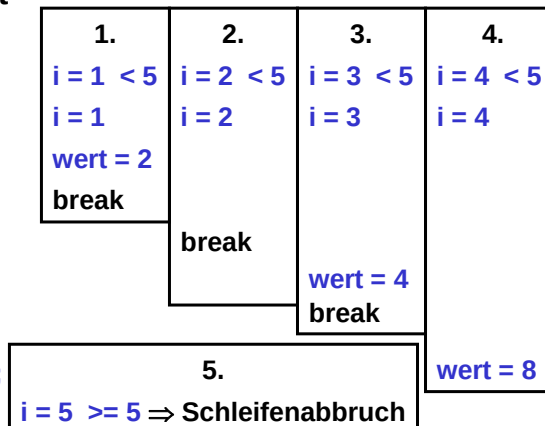
Programm 2:

• Programmabschnitt

```

for ( i = 1; i < 5; i++ )
    switch ( i )
    { case 1: wert = i+1 ;
      break ;
      case 2: break ;
      case 3: wert *= wert ;
      break ;
      default: wert += wert ;
    }
printf ( "%d\n",wert ) ;
    
```

• Schleifendurchläufe



Ausgabe: 8

Hochschule für angewandte Wissenschaften Hamburg
Prof.Dr.Ing.U.Vogelsang, Prof.Dr.U.Sauvagerd

Programmieren in ANSI C:
K6_C-grafische-Darstellung.ppt, Seite 34



Grafische Darstellung & Programmierung (1)

1. Erstellen Sie das Struktogramm eines Programms, das in einer Schleife **Euro**-Beträge in eine jeweils vorher eingegebene andere Währung umrechnet und auf dem Bildschirm ausgibt, solange die Schleife nicht durch eine Fehleingabe abgebrochen wird:

1 € (Euro) =

13,7603	Ats	(Österreichischer Schilling)
40,3399	bf	(Belgischer Franc)
1,95583	DM	(Deutsche Mark)
200,482	Es	(Portugiesischer Escudo)
6,55957	FF	(Französischer Franc)
5,94573	Fm	(Finnmark)
2,20371	hfl	(Niederländischer Gulden)
1939,27	itl	(Italienische Lira)
166,386	Pt	(Spanische Peseta)
0,787564	l£	(Irisches Pfund)

Sorgen Sie für eine verständliche und übersichtliche Darstellung der Ergebnisse auf dem Bildschirm!

Sorgen sie dafür, dass zuviel eingegebene Zeichen nicht zu einem unerwünschten Programmabbruch führen!

Hochschule für angewandte Wissenschaften Hamburg
Prof.Dr.Ing.U.Vogelsang, Prof.Dr.U.Sauvagerd



Programmieren in ANSI C:
K6_C-grafische-Darstellung.ppt, Seite 35



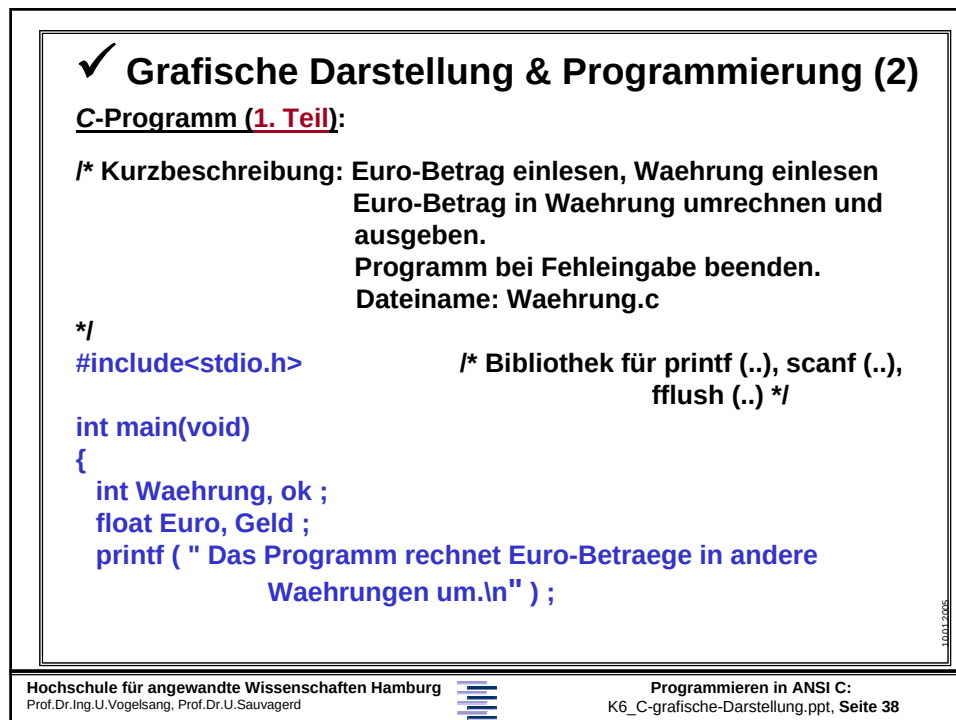
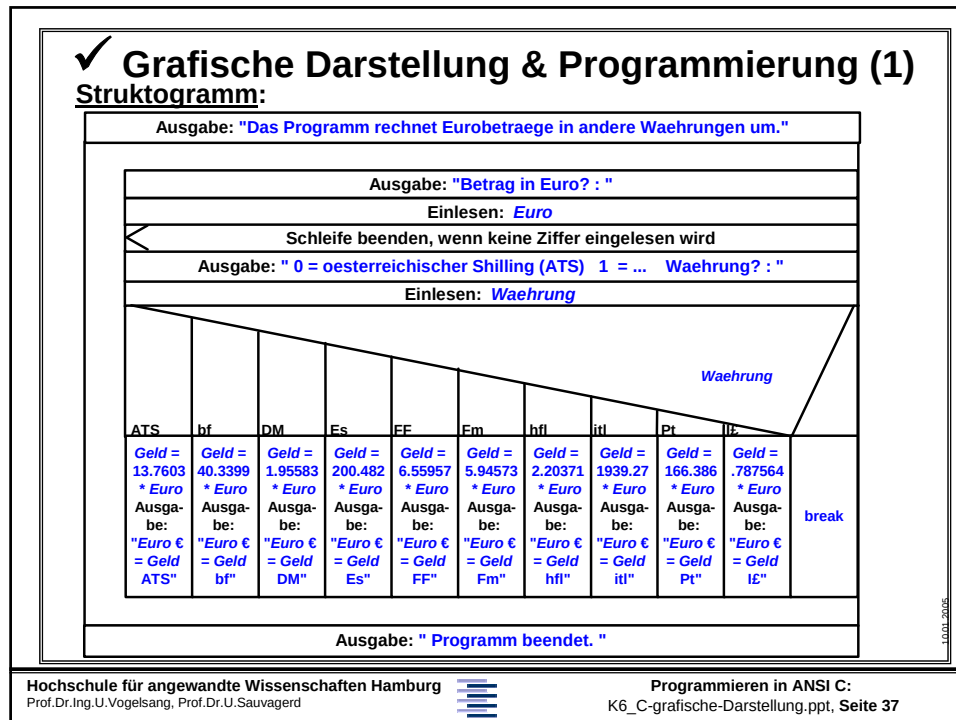
Grafische Darstellung & Programmierung (2)

2. Schreiben Sie das C-Programm entsprechend Ihrem Struktogramm!
3. Erstellen Sie den Programmablaufplan!

Hochschule für angewandte Wissenschaften Hamburg
Prof.Dr.Ing.U.Vogelsang, Prof.Dr.U.Sauvagerd



Programmieren in ANSI C:
K6_C-grafische-Darstellung.ppt, Seite 36



✓ Grafische Darstellung & Programmierung (3)

C-Programm (2. Teil):

```
while ( 1 )
{   printf ( "\n\n Betrag in Euro? (Format xxx xxxx.xx) : " ) ;
    fflush(stdin);
    ok=scanf ( "%f", &Euro ) ;
    if ( ok != 1 ) break;
    printf ( "\n Eingabe Waehrung: \n" ,
        " 0 = oesterreichischer Schilling (ATS)\n"
        " 1 = belgischer Franc (bf)\n 2 = deutsche Mark (DM)\n"
        " 3 = portugiesischer Escudo (Es)\n"
        " 4 = franzoesischer Franc (FF)\n"
        " 5 = finnische Mark (Fm)\n"
        " 6 = hollaendischer Gulden (hfl)\n"
        " 7 = italienische Lira(itl)\n"
        " 8 = spanische Peseta (Pt)\n"
        " 9 = irisches Pfund (£)\n" ) ;
    scanf ("%i",&Waehrung) ;
```



✓ Grafische Darstellung & Programmierung (4)

C-Programm (3. Teil):

```
switch (Waehrung)
{case 0:
    Geld = 13.7603 * Euro ;
    printf ( " %7.2f Euro = %9.2f ATS", Euro, Geld ) ;
    break ;
case 1:
    Geld = 40.3399 * Euro;
    printf ( " %7.2f Euro = %9.2f bf ", Euro, Geld ) ;
    break ;
case 2:
    Geld = 1.95583 * Euro;
    printf ( " %7.2f Euro = %8.2f DM ", Euro, Geld ) ;
    break ;
case 3:
    Geld = 200.482 * Euro;
    printf ( " %7.2f Euro = %10.0f Es ", Euro, Geld ) ;
    break ;
```



✓ Grafische Darstellung & Programmierung (5)

C-Programm (4. Teil):

```
case 4:
    Geld = 6.55957 * Euro ;
    printf ( " %7.2f Euro = %8.2f FF ", Euro, Geld ) ;
    break ;
case 5:
    Geld = 5.94573 * Euro;
    printf ( " %7.2f Euro = %8.2f Fm ", Euro, Geld ) ;
    break ;
case 6:
    Geld = 2.20371 * Euro;
    printf ( " %7.2f Euro = %8.2f hfl ", Euro, Geld ) ;
    break;
case 7:
    Geld = 1939.27 * Euro;
    printf ( " %7.2f Euro = %12.0f itl ", Euro, Geld ) ;
    break ;
```



✓ Grafische Darstellung & Programmierung (6)

C-Programm (5. Teil):

```
case 8:
    Geld = 166.386 * Euro;
    printf ( " %7.2f Euro = %11.0f Pt ", Euro, Geld ) ;
    break;
case 9:
    Geld = 0.787564 * Euro;
    printf ( " %7.2f Euro = %7.2f I£ ", Euro, Geld ) ;
    break ;
default:
    break ;
}
}
printf ( "\n\n Programmende! " ) ;
return 0 ;
}
```



✓ **Grafische Darstellung & Programmierung (7)**
Programmablaufplan

????

Hochschule für angewandte Wissenschaften Hamburg
Prof.Dr.Ing.U.Vogelsang, Prof.Dr.U.Sauvagerd



Programmieren in ANSI C:
K6_C-grafische-Darstellung.ppt, Seite 43

 **C-typische Programmierung**

Folgende Ausdrücke kann man in C kürzer schreiben:

1. `if ( x <= y ) Max = x ;`
 `else Max = y ;`
2. `if ( LC == 0 ) continue ;`
3. `X3 = 2*k/m + X3 - 3j ;`
4. `while ( Zaehler != 0 ) Zaehler = Zaehler - 1 ;`
5. `while ( Zaehler != -1 ) { k1 = Zaehler ;`
 `Zaehler = Zaehler-1 ; }`

Geben sie jeweils die C-typische Kurzschreibweise an und überprüfen Sie das Ergebnis ggf. mit einem Testprogramm in MSVC!

Nach der letzten (sinnvollen) Ausführung der Anweisung sollen alle Variablen in der Kurzschreibweise dieselben Werte aufweisen wie bei den oben angegebenen Befehlen!

Hochschule für angewandte Wissenschaften Hamburg
Prof.Dr.Ing.U.Vogelsang, Prof.Dr.U.Sauvagerd



Programmieren in ANSI C:
K6_C-grafische-Darstellung.ppt, Seite 44

✓ C-typische Programmierung

C-typische Kurzschreibweise:

1. `Max = ( x <= y ) ? x : y`
2. `if ( !LC ) continue;`
3. `X3 += ( 2*k/m - 3j ) ;` ⚡* (Im Zweifelsfall: Klammern setzen!)
4. `while ( --Zaehler ) ;` ⚡* man beachte Prädecrement
5. `while (k1 = Zaehler-- ) ;`
⚡* man beachte das Postdecrement

