

Übung 04: Informationstechnik (IT)

Daniel Grimm

Institutsleitung

Prof. Dr.-Ing. Dr. h.c. J. Becker

Prof. Dr.-Ing. E. Sax

Prof. Dr. rer. nat. W. Stork

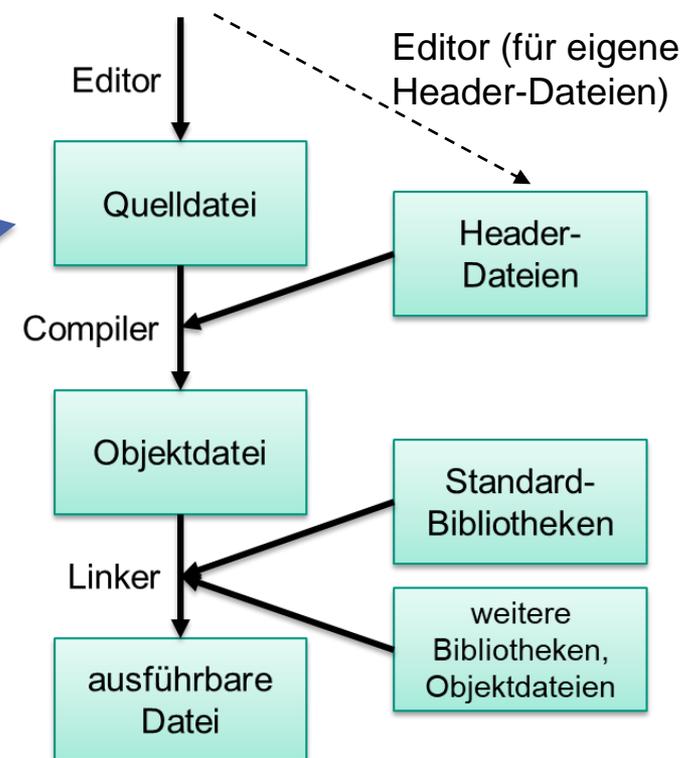
Institut für Technik der Informationsverarbeitung (ITIV)



Teil 1: Besprechung Aufgabenblatt 4

Aufg. 4.01: Verständnisfragen Lsg. (1)

- a) In C++ ist eine Standard-Header-Datei
- eine Objektdatei. → Falsch
 - eine ausführbare Datei. → Falsch
 - eine Textdatei. → Richtig



- b) Zur Ausführung der Anweisung `cout << "Himm...!" << endl;` genügt es, die Header-Datei `iostream` in Ihrem Programm zu inkludieren. → Falsch (using namespace std;)

Aufg. 4.01: Verständnisfragen Lsg. (2)

- c) Mithilfe welcher Anweisungen innerhalb einer Headerdatei kann man vermeiden, dass die Datei mehrfach eingebunden wird? → mithilfe von Include-Guards, die vom Compiler/Präprozessor verarbeitet werden

```
#ifndef _MYHEADER_  
#define _MYHEADER_  
[...]  
#endif
```

- d) In welchen Dateien findet
- die Deklaration → in der Header-Datei, Endung .h
 - die Definition → in der Quelldatei, Endung .cpp
von Funktionen statt?
- e) Was wird durch die Verwendung von Header-Dateien erreicht?
- Größerer Aufwand für Teamarbeit → Falsch
 - Aufteilung des Programms in logische Module → Richtig
 - Wiederverwendung von Code → Richtig
 - Höherer Wartungsaufwand → Falsch

Aufg. 4.01: Verständnisfragen Lsg. (3)

f) Nennen Sie drei Möglichkeiten, um einen Algorithmus zu beschreiben.

→ Nassi-Shneiderman, Flussdiagramm (grafisch)

→ Pseudocode, Programmcode

g) Welche Eigenschaften zeichnen gute Software aus?

→ Zuverlässigkeit: z.B. Stürzt das Programm häufig ab oder hängt sich auf?

→ Wartungsfreundlichkeit: z.B. ist das Programm so konzipiert, dass neue Funktionen einfach ergänzt werden können?

→ Effizienz: z.B. benötigt das Programm sehr viel Speicher, RAM oder Grafikleistung?

→ Benutzerfreundlichkeit: z.B. bietet das Programm eine übersichtliche Bedienoberfläche und eine gute Hilfefunktion?

→ Nützlichkeit: macht das Programm das, was es soll?

- Aufg. 4.01: Verständnisfragen



Aufgabe 4.02: Funktionen und Header-Dateien

Erstellen Sie ein Programm, das die Oberfläche und das Volumen einer Kugel mit dem Radius r berechnet. Dazu soll der Radius über die Tastatur eingelesen werden und später das Ergebnis auf dem Bildschirm ausgegeben werden.

Die Konstante pi ($= 3,1415927$) und die zwei Funktionen, die jeweils die Oberfläche und das Volumen berechnen, sollen in einer Header-Datei `.h` und der zugehörigen Quelldatei `.cpp` implementiert werden. Die Header-Datei wiederum soll in die Hauptdatei des `main`-Programms eingebunden werden.

```
C:\Dokumente und Einstellungen\Admin\Eigene Dateien\Cp
Geben Sie einen Radius (in cm) ein: 2
Oberflaeche der Kugel: 50.2655 cm^2
Volumen der Kugel: 33.5103 cm^3
```

$$O(r) = 4 \cdot pi \cdot r^2$$

$$V(r) = \frac{4}{3} \cdot pi \cdot r^3$$

Aufgabe 4.02: Funktionen und Header-Dateien

Lsg. (1)

main.cpp

```
#include <iostream>
#include "kugel.h"
```

```
using namespace std;
```

```
int main() {
```

```
    double r = 0;
```

```
    cout << "Geben Sie einen Radius (in cm) ein: ";
```

```
    cin >> r;
```

```
    if (!cin) {
```

```
        cout << "Ihre Eingabe ist nicht korrekt!" << endl;
```

```
    } else if( r < 0 ) {
```

```
        cout << "Sie haben einen negativen Wert eingegeben!" << endl;
```

```
    } else {
```

```
        cout << "Oberflaeche ist: " << oberflaeche( r ) << " cm2" << endl;
```

```
        cout << "Volumen ist: " << volumen( r ) << " cm3" << endl;
```

```
    }
```

```
    return 0;
```

```
}
```

Radius abfragen und einlesen

Eingaben überprüfen – ungültig / negativ
Funktionen aufrufen / Ergebnisse ausgeben

Aufgabe 4.02: Funktionen und Header-Dateien

Lsg. (2)

```
#ifndef _KUGEL_  
#define _KUGEL_
```

kugel.h

```
#include <cmath>
```

```
const double PI = 3.1415927;
```

Konstante

```
double oberflaeche( double a );
```

```
double volumen( double b );
```

Prototypen

```
#endif
```

```
#include "kugel.h"
```

kugel.cpp

```
double oberflaeche( double a ) {  
    double erg = 0;  
    erg = 4 * PI * pow( a, 2.0 );  
    return erg;  
}
```

Implementierung

```
double volumen( double b ) {  
    double erg = 0;  
    erg = 4.0 / 3 * PI * pow( b, 3.0 );  
    return erg;  
}
```

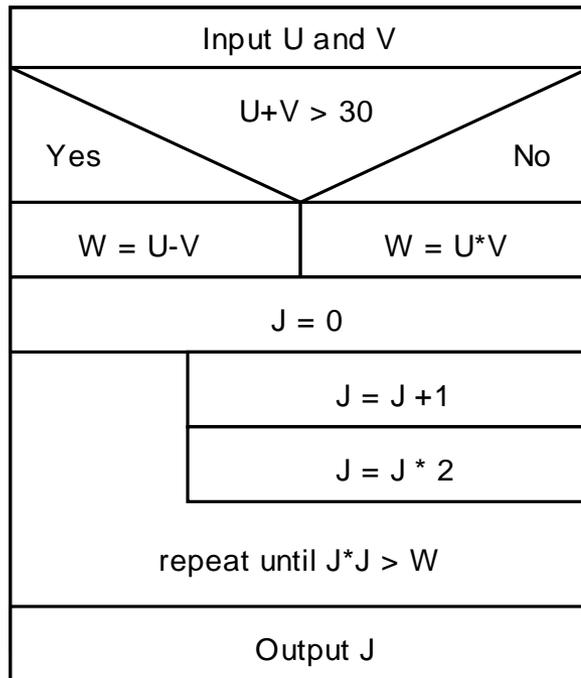
Funktion in <cmath> für x^y

- Aufg. 4.02: Funktionen und Header-Dateien



Aufgabe 4.03: Programmstrukturen Lsg.

Ein Algorithmus ist festgelegt durch das folgende Nassi-Shneiderman-Diagramm

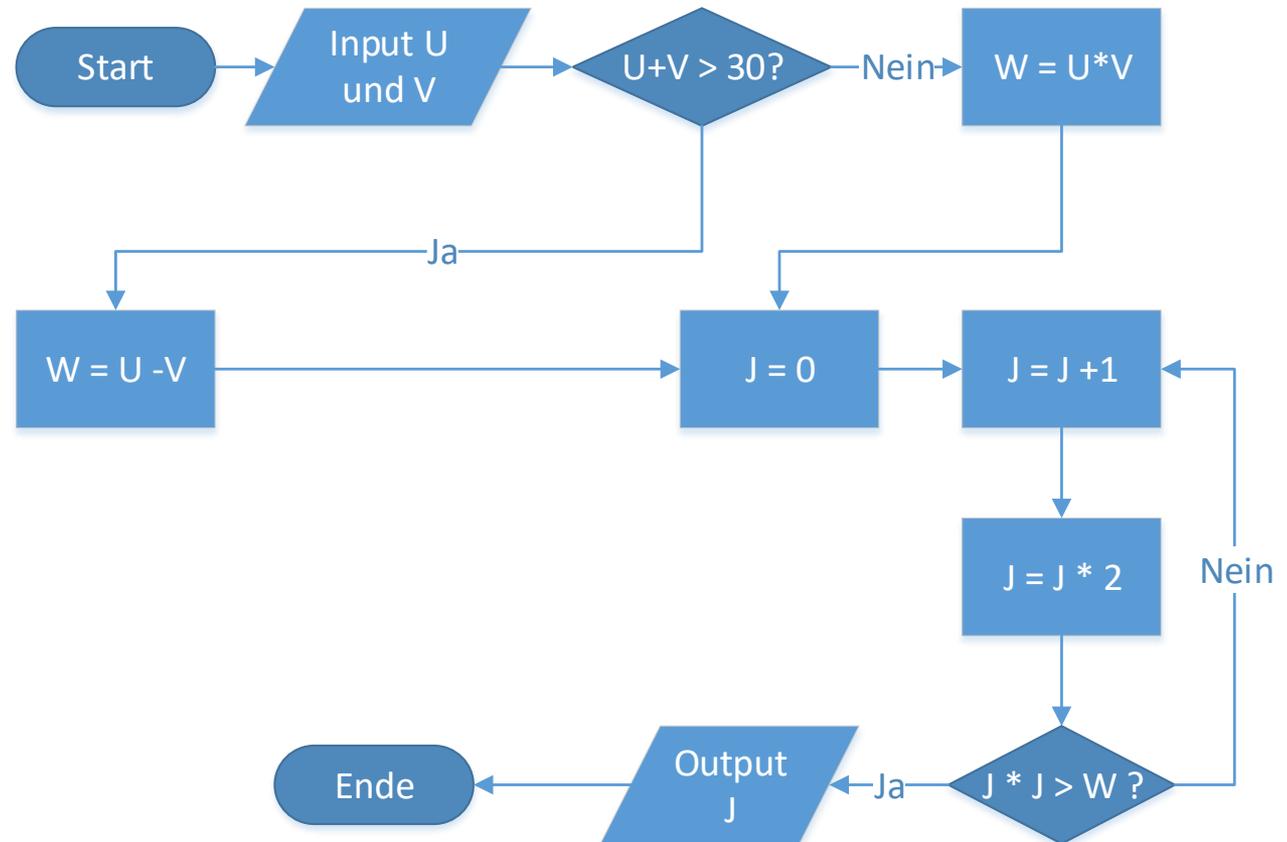


b) Zeichnen Sie ein Flussdiagramm, das die gleiche Funktion erfüllt.

Aufgabe 4.03: Programmstrukturen Lsg.

b) Zeichnen Sie ein Flussdiagramm, das die gleiche Funktion erfüllt.

Input U and V	
$U+V > 30$	
Yes	No
$W = U - V$	$W = U * V$
$J = 0$	
$J = J + 1$	
$J = J * 2$	
repeat until $J * J > W$	
Output J	

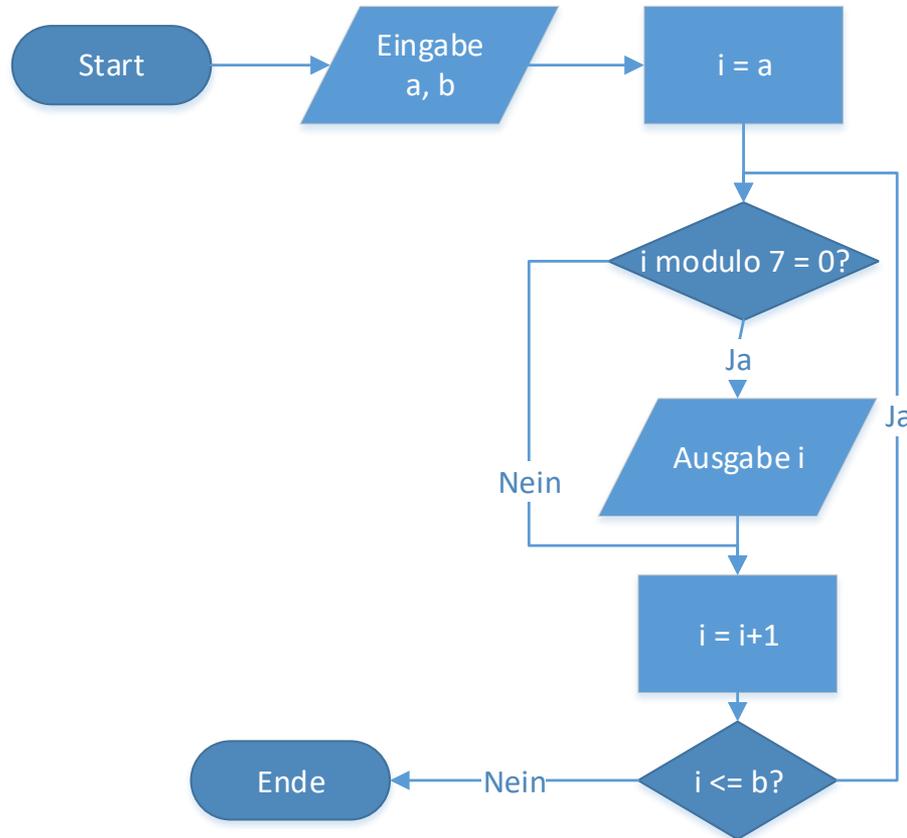


- Aufg. 4.03: Programmstrukturen



Aufgabe 4.04: Programmstrukturen II Lsg.

Gegeben ist das Flussdiagramm in der folgenden Abbildung.

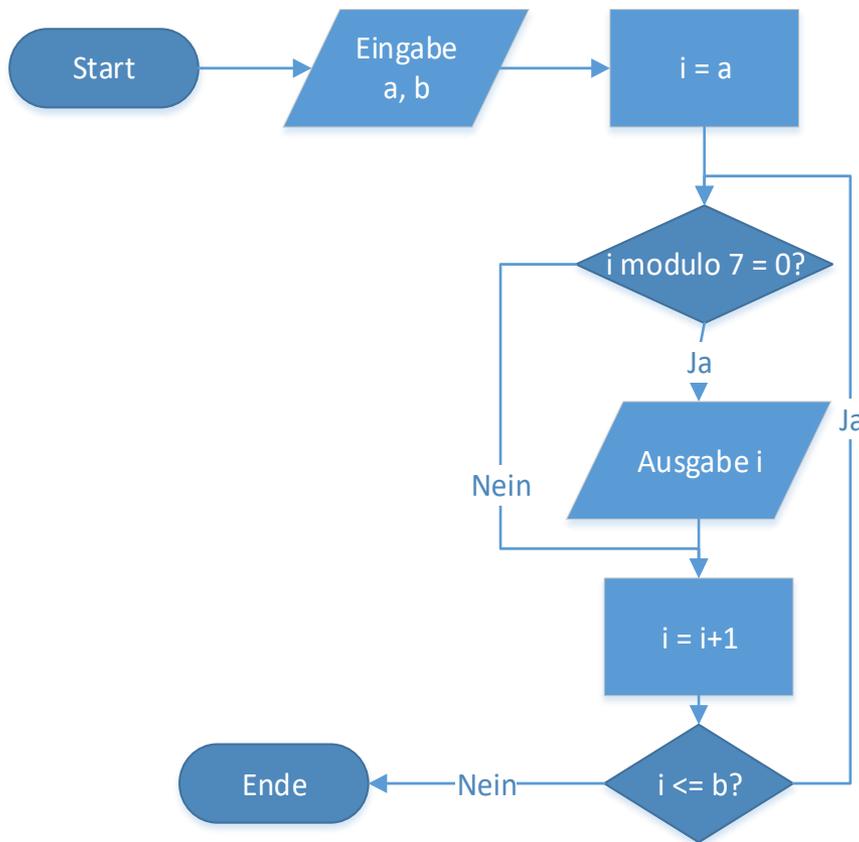


a) Das Flussdiagramm beschreibt einen Algorithmus. Erklären Sie, welche Aufgabe der Algorithmus erfüllt.

→ Berechnung aller durch 7 teilbaren Zahlen, die zwischen a und b liegen

Aufgabe 4.04: Programmstrukturen II Lsg.

Gegeben ist das Flussdiagramm in der folgenden Abbildung.



b) Notieren Sie den Algorithmus in Pseudocode oder C++-Code.

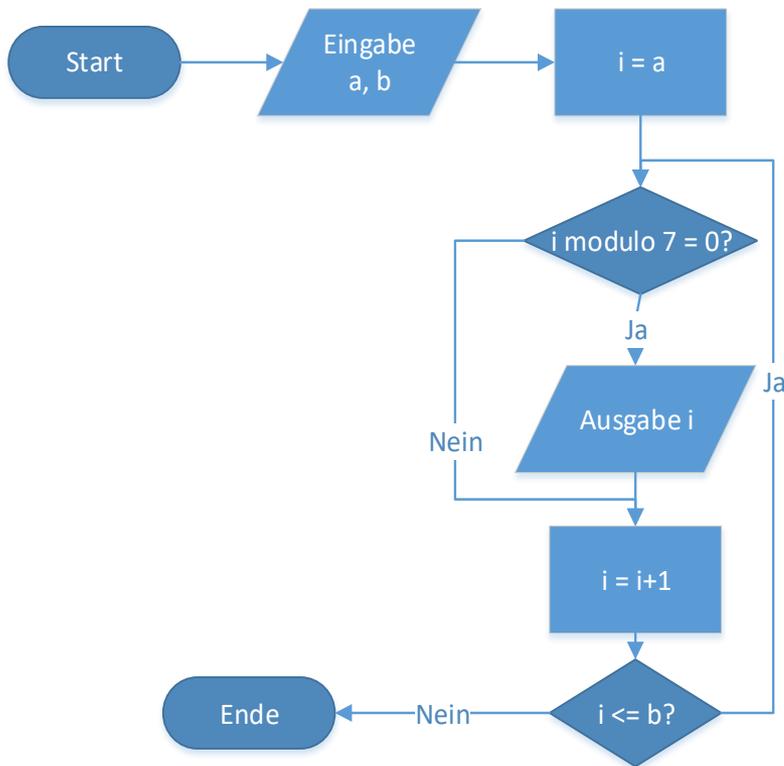
```

int a, b;
cin >> a;
cin >> b;

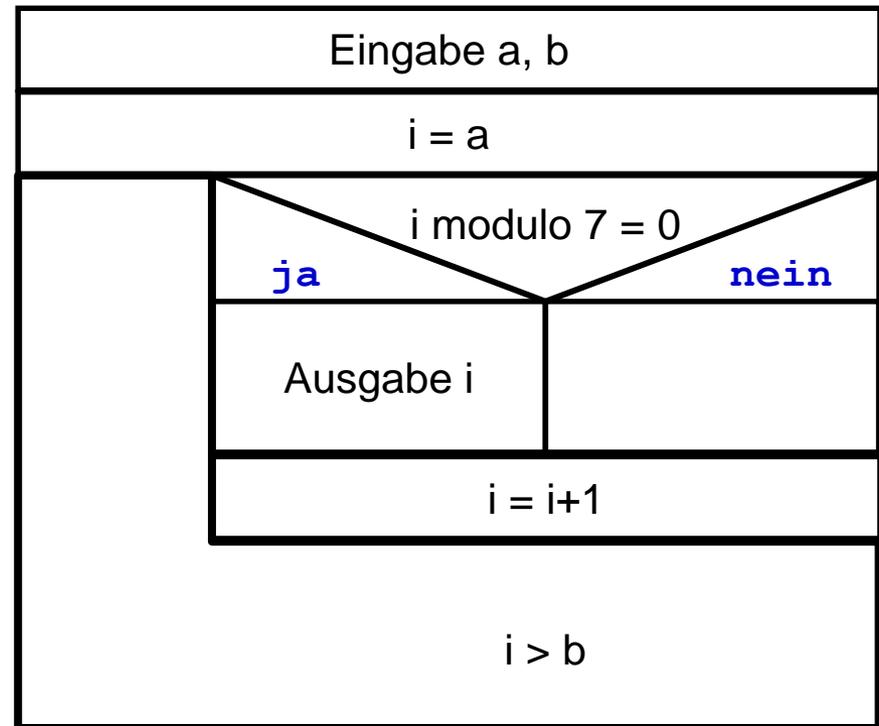
int i = a;
do
{
    if ( i % 7 == 0 )
    {
        cout << i << endl;
    }
    i++;
} while ( i <= b );
  
```

Aufgabe 4.04: Programmstrukturen II Lsg.

Gegeben ist das Flussdiagramm in der folgenden Abbildung.



c) Zeichnen Sie ein Nassi-Shneiderman-Diagramm, das die gleiche Funktion erfüllt.



- Aufg. 4.04: Programmstrukturen II



Vielen Dank für Ihre Aufmerksamkeit



Daniel Grimm
Karlsruher Institut für Technologie (KIT) – ITIV
daniel.grimm@kit.edu