



Übungsblätter zum Part 2 Grundlagen der Programmierung



GOTO-Problem



Übung



- Recherchieren Sie nach Programmiersprachen die über eine GOTO-Anweisung verfügen?
 - Welche Nachteile sind mit der GOTO-Anweisung verbunden, existieren ggf. auch Vorteile?
 - Durch welche Kontrollstrukturen kann der GOTO-Befehl (z.B. Überführung in andere Programmiersprache) ersetzt werden?
- Einzelüberlegung 5 min.
- Gemeinsame Diskussion



Übung

- Assembler (JMP, JNZ, ...), Fortran, Cobol, Basic, reserviertes Schlüsselwort in Java, aber kein einsetzbarer Befehl.
 - Orientierung an CPU-spezifischen Befehlssätzen
 - Sprung zu einer Zeile (Basic) oder Marke/Label (C, Pascal)
- Nachteile und Vorteile:
 - Nachteile: Spagetticode, Lesbarkeit und Wartbarkeit ist schlecht, Probleme beim Nachweis der Korrektheit, ...
 - Vorteile: effiziente Programmierung, Diskussion im Linux-Diskurs
- Ersetzbar durch Iterationen (while, do, for) bzw. bedingte Verzweigung (If-else)



Übung Struktogramme und Werkzeuge

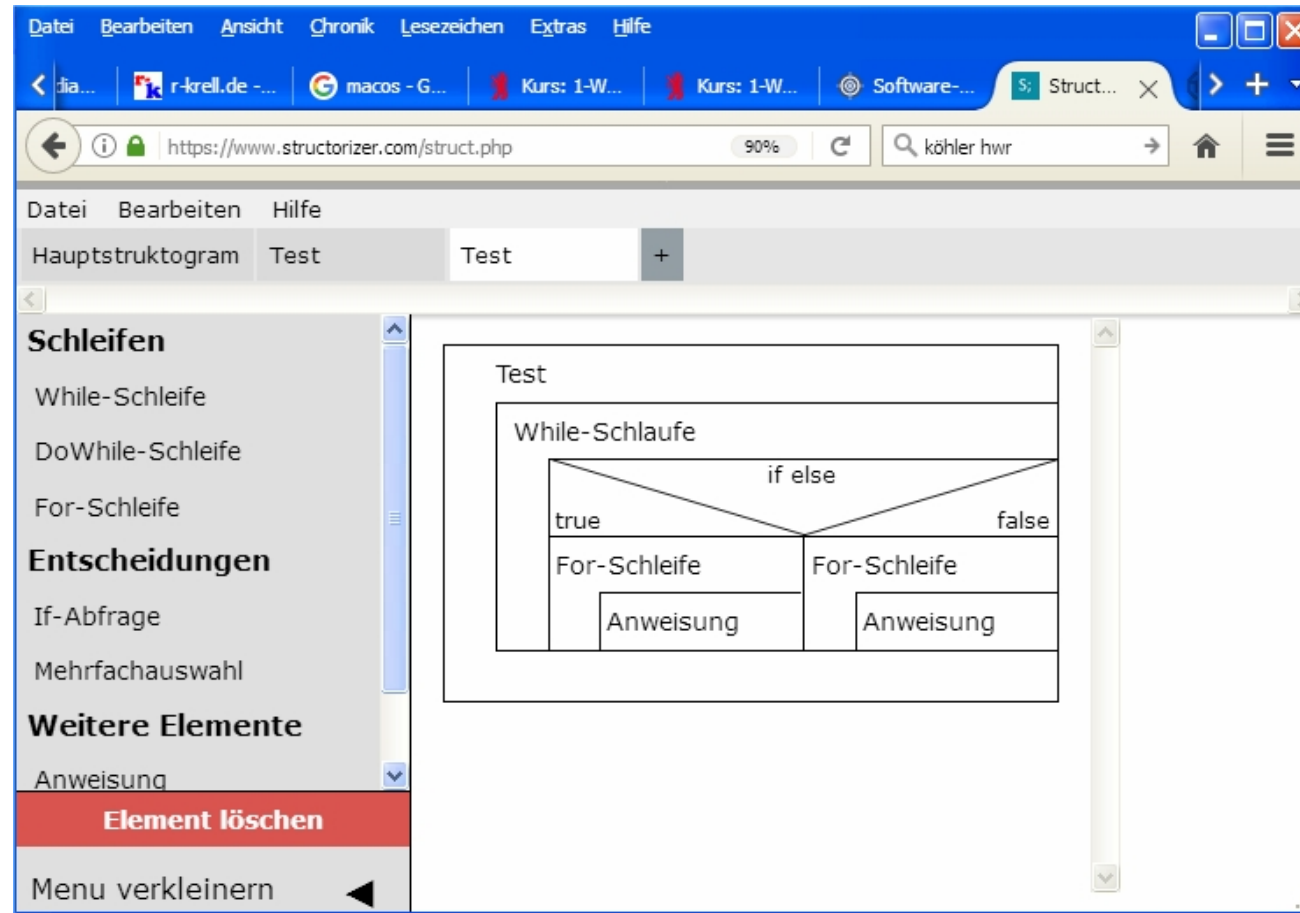


Übung



- Machen Sie sich mit der grafischen Notation entsprechend DIN 66261 bzw. EN 28631 vertraut, welche grundlegenden Elemente stehen Ihnen zur Visualisierung zur Verfügung?
- Warum erzwingt die Verwendung von Struktogrammen den Entwurf gut strukturierter Algorithmen?
- Recherchieren Sie nach potentiellen Werkzeugen die eine Erstellung von Struktogrammen unterstützen.
- Experimentieren Sie mit den Werkzeugen und den darin verfügbaren Struktogramm-Elementen.

Beispiel Structorizer



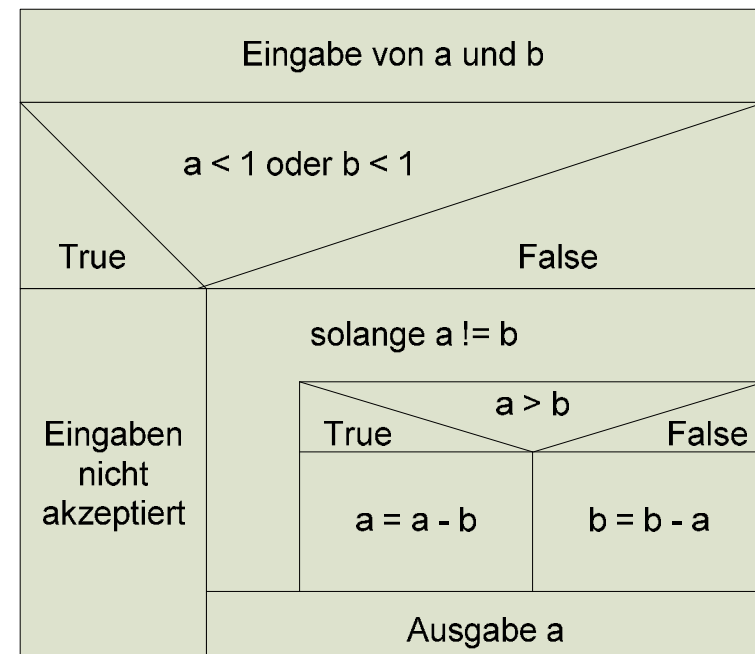
Quelle des Werkzeugs: <https://www.structorizer.com/struct.php>, letzter Abruf Oktober 2020

Übung



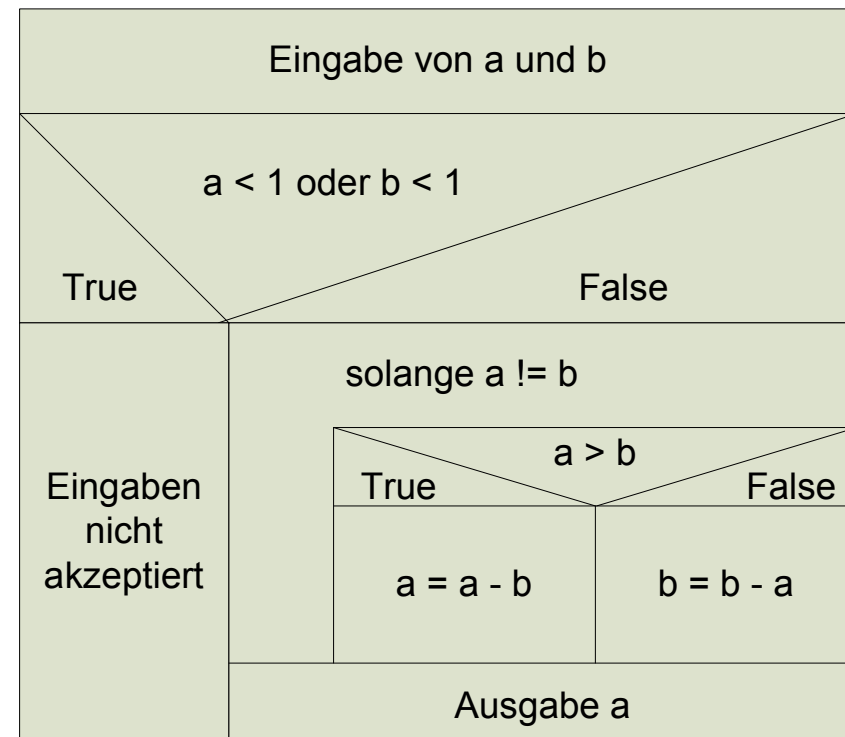
- Identifizieren Sie die verwendeten Strukturblocke?
- Überlegen Sie sich allgemeine Testfälle um alle Anweisungen mindestens einmal zu durchlaufen?
- Prüfen Sie ihre Testfälle durch Wertbelegungen für a und b?
- Fachliche Aufgabenstellung des Algorithmus?

→ 20 min. Einzelübung



Übung

- Kontrollstrukturen:
 - Sequenz von Anweisungen
 - Bedingte Verzweigung (if - else)
 - Kopfgesteuerte Schleife (while)
- Allgemeine Testfälle:
 - a und b müssen natürliche Zahlen und größer 1 sein.
 - a und b müssen natürliche Zahlen und kleiner 1 sein.
 - a und b müssen ungleich sein
 - a muß größer b sein
 - a muß kleiner b sein
- ggT-Problem



Beispielalgorithmen

Übung



- Erstellen Sie ein Struktogramm und Pseudocode zur Lösung der folgenden Problemstellung:
- Ein Fahrscheinautomat gibt Einzel-, Wochen- und Monatskarten für die Tarifzonen 1 – Kurzstrecke und 2 – Normal aus.
- Der Grundpreis für Tarifzone 1 beträgt 2 Euro, für Tarifzone 2 2,80 Euro.
- Die Preise für Wochen- und Monatskarte ergeben sich aus dem Grundpreis multipliziert mit 6 bzw. 25.
- Der Benutzer gibt jeweils eine Nummer für die Tarifzone und für die Art der Zeitkarte (Einzel = 1, Woche = 2, Monat = 3) ein. Bei Fehleingaben wird eine Fehlermeldung ausgegeben.



Übung

Hauptprogramm

Einlesen Strecke			
Strecke =			
1	2	sonst	
Grundpreis GP = 2		Grundpreis GP = 2,80	
berechneEndpreis (GP)	berechneEndpreis (GP)	Fehlerausgabe!	

Unterprogramm – berechneEndpreis(GP)

Einlesen Zeit			
Zeit =			
1	2	3	sonst
EP = GP	EP = 6*GP	EP = 25*GP	Fehler!



Übung



- Formulieren Sie in Pseudocode eine Zählschleife, die von $n=1$ auf 40 mit der Schrittweite 4 zählt.
 - Formen Sie diese Schleife in eine funktional äquivalente kopfgesteuerte Schleife um.
 - Formen Sie diese Schleife in eine funktional äquivalente fußgesteuerte Schleife um.
 - Optional: Geben Sie für alle in Pseudocode erstellten Schleifen die korrespondierenden Struktogramme an. (optionale Aufgabe in Heimarbeit)
- Gruppenüberlegung 20 min.
- Gemeinsame Auswertung und Übungsblätter

Übung

```
int weite = 1;
// Zaeahlschleife
for (int i=weite; i<=40; i+=4) {
    // Ausgabe des jeweiligen i
    System.out.println("i ist"+i);
}

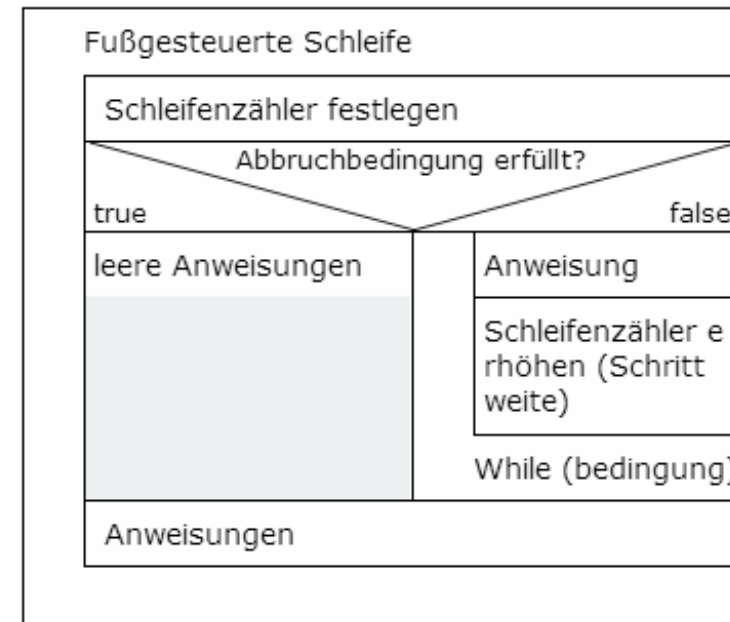
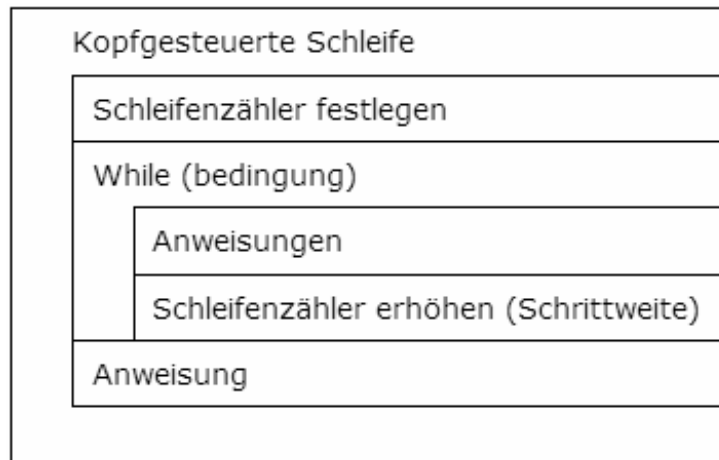
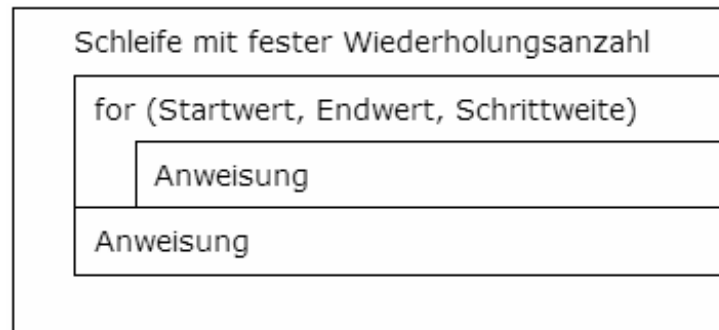
// Kopfgesteuerte Schleife
int j=weite;
while(j<=40){
    // Ausgabe des jeweiligen j;
    System.out.println("j ist"+j);
    // Abbildung der Schrittweite
    j=j+4;
}
```

Bei fussgesteuerten Schleifen
wird der Schleifenkörper mindestens
1 x durchlaufen, das Prüfen der
Abbruchbedingung findet am Ende statt.

Achtung – die Forderung der
funktionalen Äquivalenz bedingt die
Notwendigkeit der Prüfung des Inhalts
der Schrittweite! Wir können in realen
Programmen nicht von der Kenntnis des
Inhalts von k ausgehen!

```
int k=weite;
if (k<=40){
    // Fussgesteuerte Schleife
    do{
        // Ausgabe des jeweiligen k;
        System.out.println("k ist"+k);
        // Abbildung der Schrittweite
        k=k+4;
    } while (k<=40);
}
else{
    System.out.println("k ist zu gross"+k);
}
```

Übung



Achtung: Die dargestellten Struktogramme zeigen die grundsätzlichen Lösungsansätze, es fehlen Details (vgl. Quellcode)!

Quelle des Werkzeugs: <https://www.structorizer.com/struct.php>, letzter Abruf Oktober 2020

Übung



- Erstellen Sie ein Struktogramm zur Berechnung der Fakultät.

Die Fakultät ist das Produkt der ersten n natürlichen Zahlen: $n!$

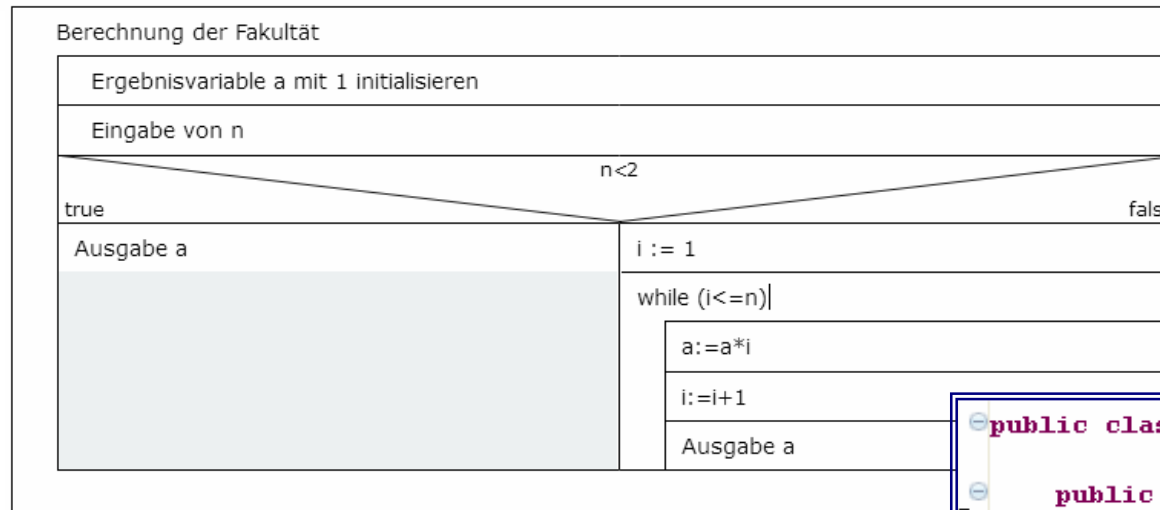
- Geben Sie den korrespondierenden Pseudocode (frei wählbare Programmiersprachkonstrukte) zu diesem Algorithmus an.

$$n! = \prod_{i=1}^n i = 1 \times 2 \times \dots \times n, \quad n \in \mathbb{N}, \quad 0! = 1$$

→ ca. 30 min. (optionale Aufgabe in Heimarbeit)

→ Auswertung siehe Übungsblätter

Übung



Ablauffähiges Programm in Java,
hier als Beispiel für Pseudocode.

```
public class Fakultät {  
  
    public static void main(String[] args) {  
        int a = 1;  
        int n = 6;  
        if (n < 2) {  
            ausgabe(a);  
        } else {  
            int i = 1;  
            while (i <= n) {  
                a = a * i;  
                i = i + 1;  
            }  
            ausgabe(a);  
        }  
    }  
}
```

Quelle des Werkzeugs: <https://www.structorizer.com/struct.php>, letzter Abruf Oktober 2020