

Praktische Übung zu DO und IF

Sonderübung

Emily Gasmann & Anneli Richter

PR10

08.11.2021

1. Wiederholung
 - Verzweigungen
 - Schleifen
 - Wiederholungsaufgabe
2. Effizienz von Algorithmen
3. Aufgaben

Was haben wir in der letzten Übung besprochen?

Verzweigungen

- IF - THEN - ELSE - END IF

- SELECT CASE:

Was haben wir in der letzten Übung besprochen?

Verzweigungen

- IF - THEN - ELSE - END IF

```
IF (Bedingung) THEN  
    Anweisungsblock  
ELSE  
    Anweisungsblock  
END IF
```

- SELECT CASE:

Was haben wir in der letzten Übung besprochen?

Verzweigungen

- IF - THEN - ELSE - END IF

```
IF (Bedingung) THEN
    Anweisungsblock
ELSE
    Anweisungsblock
END IF
```

- SELECT CASE:

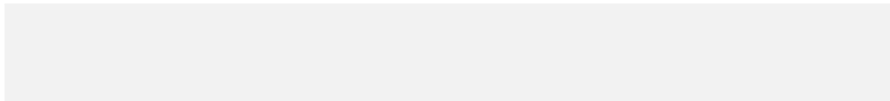
```
SELECT CASE (Variable)
    CASE (Wertebereich)
        Anweisungsblock

    ...
    CASE DEFAULT
        Anweisungsblock
END SELECT
```

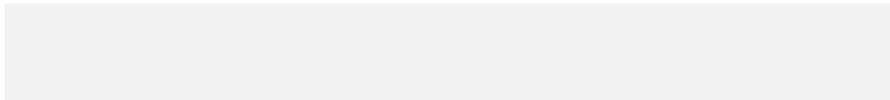
Was haben wir in der letzten Übung besprochen?

Schleifen

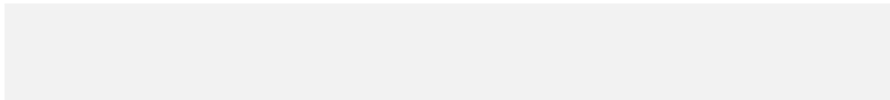
- Klassische DO - Schleife:



- DO WHILE - Schleife:



- Zählschleife:



Was haben wir in der letzten Übung besprochen?

Schleifen

- Klassische DO - Schleife:

```
DO
```

```
    Anweisungsblock
```

```
END DO
```

ACHTUNG: Schleifenabbruch nicht vergessen!

- DO WHILE - Schleife:

- Zählschleife:

Was haben wir in der letzten Übung besprochen?

Schleifen

- Klassische DO - Schleife:

```
DO
    Anweisungsblock
END DO
```

ACHTUNG: Schleifenabbruch nicht vergessen!

- DO WHILE - Schleife:

```
DO WHILE (Bedingung)
    Anweisungsblock
END DO
```

- Zählschleife:

Was haben wir in der letzten Übung besprochen?

Schleifen

- Klassische DO - Schleife:

```
DO
    Anweisungsblock
END DO
```

ACHTUNG: Schleifenabbruch nicht vergessen!

- DO WHILE - Schleife:

```
DO WHILE (Bedingung)
    Anweisungsblock
END DO
```

- Zählschleife:

```
DO k = a, e [, i]
    Anweisungsblock
END DO
```

Schleifen (DO loops)

Aufgabe

- Schreibt ein Programm, das alle Tripel dreier natürlicher Zahlen findet, deren Summe 50 ist.
ZUSATZ: Überlegt euch eine Möglichkeit, wie jede Zahlenkombination nur genau einmal ausgegeben wird (d. h. die Reihenfolge spielt keine Rolle).
TIPP: Nutzt Zählschleifen!
Zählt die Anzahl der Tripel und gebt diese anschließend auf dem Terminal aus.
- Schreibt ein Programm, das alle vierstelligen Zahlen ausgibt, die man aus den Ziffern 1,2,3,4 bilden kann. (Nutzt Zählschleifen!)

Effizienz von Algorithmen

Anmerkungen zur Aufgabe "Alle Tripel, deren Summe 50 ist"

Für dieses Programm gibt es verschiedene Möglichkeiten der Implementierung, die alle das gleiche Ergebnis liefern, aber unterschiedlich schnell sind. Anhand einer Zeitmessung werden wir das verdeutlichen.

```
REAL :: start, end  
  
CALL CPU_TIME(start)  
! do something  
CALL CPU_TIME(end)  
  
WRITE(*,*) 'Gesamtzeit: ', end - start
```

Schachteln oder Verknüpfen von IF-Anweisungen

Für die Berechnung von Schaltjahren gibt es folgende drei Regeln:

- Ist die Jahreszahl durch 4 teilbar, dann ist das Jahr ein Schaltjahr.
- Ist die Jahreszahl jedoch durch 4 und 100 teilbar, dann ist das Jahr kein Schaltjahr.
- Ist die Jahreszahl durch 4, 100 und 400 teilbar, dann ist das Jahr ein Schaltjahr.

Schreibe ein Programm, das eine Jahreszahl einliest und als logischen Wert (LOGICAL) ausgibt, ob das Jahr ein Schaltjahr ist (true) oder nicht (false). Finde geeignete Testwerte, um alle vier Fälle abzudecken.

Zusatz: Löse die Aufgaben in nur einer Anweisung.

Hinweis: $\text{MOD}(a,b)$ liefert den Rest von a bei der Teilung durch b .

Quersumme

Schreibe ein Programm, das die Quersumme einer natürlichen Zahl berechnet. Lies dazu zuerst solange eine Zahl ein, bis eine Zahl ≥ 0 eingegeben wird. Berechne dann mithilfe einer Schleife die Quersumme dieser Zahl und gib diese am Ende aus.

Zusatz: Berechne in der Schleife zusätzlich die umgedrehte Zahl (d.h. die umgedrehte Zahl von 1234 ist 4321) und gib diese ebenfalls am Ende aus.

Zahlenrätsel

100 Scheffel Weizen wurden unter einer gewissen Anzahl von Personen restlos aufgeteilt. Dabei erhielt jeder Mann 3 Scheffel, jede Frau 2 Scheffel und jedes Kind einen halben Scheffel. Berechne alle Möglichkeiten.

Collatz-Vermutung

Betrachte folgende Funktion:

$$f : \mathbb{N} \rightarrow \mathbb{N}$$
$$f(n) = \begin{cases} 3n + 1 & n \text{ ungerade} \\ n/2 & n \text{ gerade} \end{cases}$$

Die Collatz-Vermutung besagt, dass unabhängig von der Zahl mit der man anfängt, irgendwann 1 herauskommt. Zum Beispiel ergibt sich beim Start mit 19 folgende Folge:

19, 58, 29, 88, 44, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1

Schreibe ein Programm, das eine Zahl einliest und für diese berechnet, wie viele Schritte benötigt werden, bis sich 1 ergibt.

Zusatz: Berechne, bei welcher Zahl zwischen 1 und 100 sich die längste Kette ergibt.