



## 13.20 Strukturen

**Struktur:** Menge von verschiedenartigen Variablen, die einen inhaltlich zusammengehörigen Datensatz beschreiben.

```
/* Deklaration */
struct Student {
    char    Name[25];
    long int MatrikelNr;
    char    Studiengang[3];
};
```

```
/* Definition */
struct Student Stud1;

/* Wertzuweisung */
Stud1.MatrikelNr = 125716;
```

**Anm.:** Die Struktur-Deklaration ist lediglich eine Beschreibung !  
Dabei wird kein Speicherplatz reserviert.

Erst durch die Definition wird für die Struktur Speicherplatz angelegt.



## 13.20 Strukturen (Fortsetzung)

```
/* Deklaration und Definition in einem Schritt */
```

```
struct Student {  
    char    Name[25];  
    long int MatrikelNr;  
    char    Studiengang[3];  
} Std1, Std2, Std3;
```

```
/* Definition und Initialisierung in einem Schritt */
```

```
struct Student Stud1 = {"Zuse", 125716, "TI"};  
struct Student Stud2 = {"Gates", 125609, "AI"};
```



## ÜBUNG: Struktur deklarieren, definieren und initialisieren

Deklarieren Sie eine Struktur "Datum".

Diese soll 3 Integerwerte für den Tag, Monat und das Jahr enthalten sowie ein Textfeld von 50 Zeichen für Stichworte (Memo).

Erzeugen Sie 2 Termine (Typ "Datum") und initialisieren Sie diese:

Silvesterparty : 31.12.2016, "Party ab 20:00."

Geburtstag : 07.02.2016, "Geburtstag Werner"



## 13.21 Arrays von Strukturen

```
/* Definition von 60 Datenstrukturen vom Typ Student */
```

```
struct Student Std[60];
```

```
/* Initialisierung einer Datenstruktur des Arrays */
```

```
strcpy( Std[0].Name , "Meier");
```

```
Std[0].MatrikelNr  = 125716;
```

```
strcpy( Std[0].Studiengang, „AI");
```

```
/* Definition und Initialisierung von 3 Datenstrukturen  
vom Typ Student */
```

```
struct Student Std[3] = {{"Meier", 125716, „AI"},  
                        {"Meiser", 125367, „AI"},  
                        {"Gabski", 123362, "TI"}};
```



## ÜBUNG: Felder von Strukturen definieren und initialisieren

Definieren Sie ein Feld (Name: *Termin*) von 20 Strukturen (Typ "Datum", s.o.) und initialisieren Sie die ersten Strukturen mit

Silvesterparty : 31.12.2016, "Party ab 20:00."

Geburtstag : 07.02.2016, "Geburtstag Werner"

- a) Definition und Initialisierung in einem Schritt.
- b) Definition und Initialisierung in getrennten Schritten.



## 13.22 Zeiger und Strukturen

```
/* Definition der Datenstruktur Student */  
struct Student Std_1;  
  
/* pStd ist Zeiger auf Datenstruktur Student */  
struct Student *pStd = &Std_1;  
  
/* Zugriff auf Strukturelemente über den Zeiger */  
strcpy( (*pStd).Name, "Gallagher");  
(*pStd).MatrikelNr = 125616;  
  
/* ... oder kürzer (und ueblich) ... */  
strcpy( pStd->Name, "Townsend");  
pStd-> MatrikelNr = 125616;
```



## ÜBUNG: Übergabe von Strukturen an Unterprogramme

Es ist ein Programm zu schreiben, welches jeweils nach Ablauf einer Sekunde die Uhrzeit in der Form HH:MM:SS ausgibt.

Die Uhrzeit soll in einer Struktur gespeichert sein:

```
struct tm {  
    int hours;  
    int minutes;  
    int seconds;  
};
```

Neben dem Hauptprogramm soll das Programm aus den beiden Funktionen

update: Aktualisieren der time-Struktur

display: Anzeigen der time-Struktur

bestehen.



## 13.23 Selbstdefinierte Datentypen

In C lassen sich neue Datentypen definieren.

Hierzu dient das Schlüsselwort `typedef`.

```
/* Deklaration exakter Datentypen (s. stdint.h)*/  
/* Achtung: Prozessorabhängig !!                */
```

```
typedef    signed    char        int8_t;  
typedef    signed    short       int16_t;  
typedef    signed    long        int32_t;  
typedef    unsigned  char        uint8_t;  
typedef    unsigned  short       uint16_t;  
typedef    unsigned  long        uint32_t;  
typedef    signed    long long    int64_t;  
typedef    unsigned  long long    uint64_t;
```

```
/* Variablendeklaration */
```

```
int8_t      b1,  b2=0xF1;  
uint16_t    w[10];  
int         a=15;
```





## 13.23 Selbstdefinierte Datentypen (Fortsetzung)

```
/* Deklaration eines neuen Struktur-Typen */
```

```
typedef struct {  
    char      Name[25];  
    long int   MatrikelNr;  
    char      Studiengang[3];  
} Student; /* ← Name des neuen Datentyps */
```

```
/* Anlegen von 20 Strukturen des Typs "Student" */
```

```
Student Std[20]; /* nicht mehr : struct Student */  
strcpy( Std[0].Name, "Winter");
```