

Übung 11: Struct Datentyp

Mit dem *Struct* Datentyp kann man verschiedene Datentypen in einer einzigen Variable zusammenfassen. Das ist sehr praktisch um Datenbanken zu implementieren.



Verzeichnis von Begriffen

- Datentyp
- Struct
- Komponente
- Initialisierung
- Punkt Operator
- Definition einer Struktur
- Deklaration einer *struct*-Variable
- Datenbank

AUFGABE 0: Betrachten Sie das untenliegende Programm und vor dem Ausführen antworten Sie die nachfolgenden Fragen.

```
#include <iostream>
using namespace std;
struct STank
{
    char Datum[11];
    int Fahrstrecke;
    float Fuellmenge, Durchschnittsverbrauch;
};
int main ( void )
{
    STank Tank, aTank[10], *pTank;

    system("PAUSE");
    return 0;
}
```

a) Versuchen Sie zunächst selbst, folgende Fragen zu beantworten:

Wie groß (in Bytes) ist der Datentyp **STank** ?

Wie groß ist eine Variable bzw. ein Feld mit 10 Elementen vom Typ **STank** ?

Wie groß ist ein Zeiger auf den Typ **STank** ?

b) Schreiben Sie ein kleines Programm, mit dem Sie Ihre Antworten zu a) überprüfen können!

AUSGABE:

```
Groesse des Datentyps STank      : 24 Byte
Groesse der STank-Variablen Tank : 24 Byte
Groesse des STank-Feldes aTank  : 240 Byte
Groesse des STank-Zeigers pTank : 4 Byte
```

3
4
5
6

AUFGABE 1: Wir wollen ein Programm schreiben, in dem eine **main-lokale (!) Variable Tank** vom Typ **STank** definiert wird. Das Programm soll zur Eingabe der Daten einer Kfz-Betankung auffordern, also Datum, Fahrstrecke und Füllmenge. Die Eingabedaten sollen in die Variablen **Tank** eingelesen werden. Danach soll das Programm den Durchschnittsverbrauch berechnen und in der entsprechenden Komponente der Variablen **Tank** speichern. Am Ende soll dann noch einmal der Inhalt der Variablen **Tank** vollständig ausgegeben werden.

```
Berechnung des Durchschnitts-Kraftstoffverbrauchs

Bitte die Tank-Daten eingeben:
Datum < [TT].[MM].JJJJ > = 25.04.2005
Fahrstrecke < in km > = 645
Fuellmenge < Liter > = 38.16

Der vollstaendige Datensatz:
Datum      : 25.04.2005
Fahrstrecke : 645 km
Fuellmenge  : 38.16 Liter
Schnittverbrauch: 5.92 Liter/100km
```



Um maximalen Gewinn aus dieser Übung zu ziehen, sollen außer der Variablen **Tank** keine weiteren Variablen definiert werden!

Aufgabe2

Machen Sie sich eine Kopie des Programms von Aufgabe1 und lagern Sie die Teilaufgaben Eingabe, Berechnung des Durchschnittsverbrauchs und Ausgabe des Inhalts der **STank** Variablen in entsprechende Funktionen aus! Unsere **.cpp**-Datei soll also insgesamt vier Funktionen enthalten:

- **main** soll die beiden Zeilen "Berechnung des Durchschnitts-Kraftstoffverbrauchs" und "Der vollstaendige Datensatz:" ausgeben und sonst nur die drei Funktionen **Eingabe**, **Schnitt** und **Ausgabe** aufrufen!
- **Eingabe** soll den Eingabedialog abwickeln und die drei Eingaben in der **main**-lokalen Variablen **Tank** speichern.
- **Schnitt** soll aus den in der Variablen **Tank** gespeicherten Daten den Durchschnittsverbrauch berechnen und in der entsprechenden Komponente derselben Variablen speichern.
- **Ausgabe** soll den nun vollständigen Inhalt der **main**-Variablen **Tank** ausgeben.

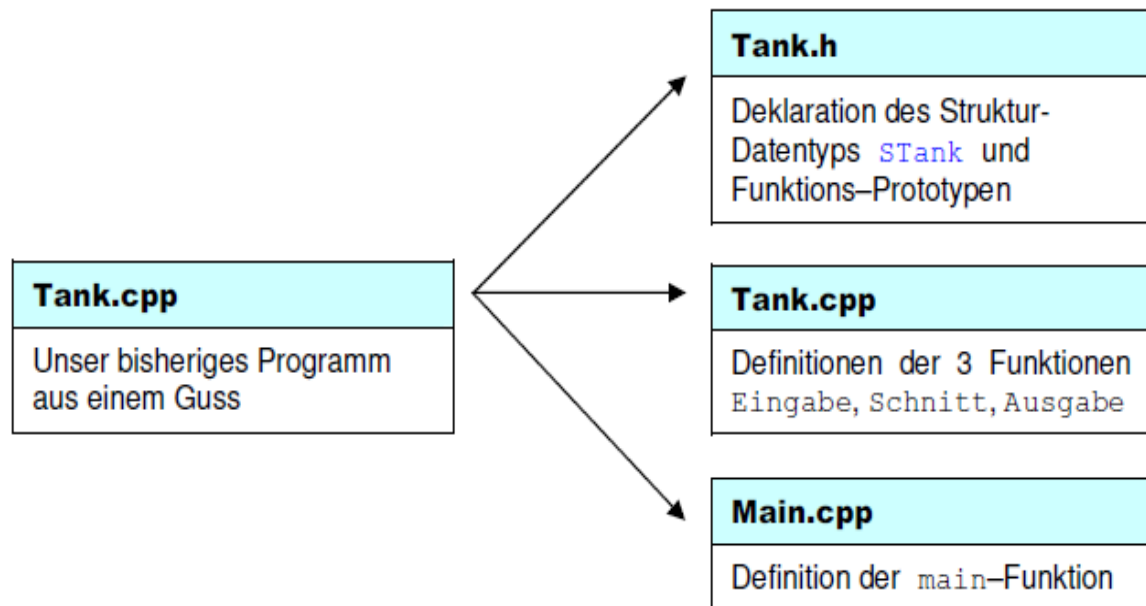


Bitte beachten Sie folgende wichtige **Forderungen**:

- Nach wie vor soll in **main** nur die Variable **Tank** (d.h. keine weiteren Variablen) definiert werden!
- Die drei Funktionen **Eingabe**, **Schnitt** und **Ausgabe** sollen keinen Rückgabewert liefern, also folgende Deklarationen haben:

```
void Eingabe ( ? );           // ? = Ihre Aufgabe !
void Schnitt ( ? );
void Ausgabe ( ? );
```

Aufgabe3 :Machen Sie sich eine Kopie des Programms von Aufgabe2 und verteilen Sie es auf drei Dateien



Dozenten J.L. Muluem, J. Purcalla Arrufí