Strukturen

Lernziele

Ich kann in einem C-Programm:

- ✓ Eine Struktur als eigenständigen Datentyp deklarieren.
- ✓ Eine Strukturvariable definieren, initialisieren und bearbeiten.

Mit typedef ein Alias definieren

Mit *typedef* kann ein Alias für einen schon existierenden beliebigen Typ definiert werden:

```
typedef   <type>   <alias>
```

Beispiele:

```
typedef int Integer;          //Erstellt ein Alias für den Datentyp int

float preis = 3.99;
Integer zahl = preis;         //zahl = 3

typedef char* String;         //String ist ein Alias für char*

String s = (char*)"Hallo";    //genau wie: char* s = (char*)"Hallo";
printf("%s\n", s);            //Ausgabe: Hallo
```

Typedef.cpp

Wieso Strukturen?

Problemstellung: Wie definiere ich eine Adresse?

Eine Adresse beinhaltet:

- Name
- Vorname
- Telefonnummer
- Strasse
- PLZ
- etc.

Lösung:

Eine Adresse als einen zusammengesetzten Datentyp *Struktur* definieren. Eine Struktur in C kann unterschiedliche, aber logisch zusammenhängende Daten aufnehmen.

Was ist eine Struktur?

- Integer, float, usw. sind einfache Datentypen

- Was hat aber eine Adresse, eine Koordinate für einen Datentyp?
  - Komplexer Datentyp
  - Besteht aus mehreren zusammengehörenden Variablen unterschiedlicher Typen

- Struktur in C:
  - fasst zusammengehörende Variablen zu einer Einheit zusammen.

Definition: Variante 1

Deklaration:

```
struct Koordinate
 {
   int x;
   int y;
 };
```

Definition einer Variablen:

```
struct Koordinate p1;
```

Kombination von Deklaration und Definition:

```
struct Koordinate
 {
   int x;
   int y;
 } p2;
```

StructDef1.cpp

Definition: Variante 2

Definition eigener Datentypnamen mit typedef:

1. Festlegung eines Aliasnamens zu einem existierenden benannten Datentyps:

```
Syntax:   typedef typname aliasname;
```

```
Beispiel:
   typedef unsigned int uint;
```

2. Die Benutzung einer Struktur kann mit **typedef** vereinfacht werden.

```
typedef struct
 {
   uint x;
   uint y;
 } Koordinate;


Koordinate p;
```

StructDef2.cpp

Initialisierung von Strukturvariablen

- Erfolgt in der selben Reihenfolge wie die Auflistung der Felder in der Definition.
- Wenn weniger Werte als Felder in der Initialisierungsliste stehen, werden die restlichen Felder mit 0 gefüllt.

```
typedef struct
 {
   int x;
   int y;
 } Koordinate;

Koordinate p={100, 200};
```

Zugriff auf einzelnes Element

Mit dem Punktoperator kann man auf einzelnes Element (Feld) einer Struktur zugreifen:

```
typedef struct
 {
   int x;
   int y;
 } Koordinate;
Koordinate p = {10, 20};

/*Schreiben von Werten in die Strukturvariable*/
   p.x = 30;
   p.y = 40;

/*Ausgabe von Feldern der Strukturvariablen */
   printf("P:(%d, %d)\n", p.x, p.y);
```

Kopieren einer Struktur

Bei der Kopie einer Strukturvariablen werden alle Einzelelemente (Felder) der Struktur kopiert:

```
typedef struct
{
        char surname[31];
        char name[31];
        char street[41];
        int postcode;
        char place[31];
} Address;
```

```
int main()
{
        // Definition und gleichzeitige Initialisierung der Strukturvariablen person1:
        Address person1 = { "Meier", "Hans", "Sternmatt 11a", 6789, "Luzern" };
        Address person2, teacher;

        //Einzelne Felder von person1 uebernehmen:
        strcpy_s(person2.surname, "Meier");
        strcpy_s(person2.name, "Hans");
        strcpy_s(person2.street, "Sternmatt 11a");
        person2.postcode = 6789;
        strcpy_s(person2.place, "Luzern");

        //Bei der Kopie einer Strukturvariablen werden alle Felder der Struktur kopiert:
        teacher = person2;

        return 0;
}
```

Verschachtelte Struktur

Bei der Kopie einer Strukturvariablen werden alle Einzelelemente (Felder) der Struktur kopiert:

```
typedef struct {
    char tag;
    int monat;
    int jahr;
} Datum;

typedef struct  {
    char name[20];
    int gewicht;
    Datum geburtsdatum;
} Student;
```

```
int main()
{
    Student dummy = { "Hans", 70, {1, 2, 1999} };
    Student me;

    printf("Geben Sie Ihren Name ein>");
    scanf_s("%s", me.name, sizeof(me.name));
    printf("Geben Sie Ihren Geburtsdatum ein
          (dd.mm.yyyy)>");
    scanf_s("%2d.%2d.%4d", &me.geburtsdatum.tag,
        &me.geburtsdatum.monat, &me.geburtsdatum.jahr);



}
```

PersonMitDatum.cpp

Übung 1 Adressbuch Teil 1+2

Aufgabenstellung siehe M411_A_XCH_Adressbuch_Teil1.pdf bzw.

M411_A_XCH_Adressbuch_Teil2.pdf