

Fachhochschule Bingen

Programmieren

Strukturen

Prof. Dr. Maximilian Mengel,
Professur Programmiermethodik,
Grundlagen der Informatik und Multimedia
Gebäude 1, Raum 212
Tel.: 06721-409 152
E-Mail: mengel@fh-bingen.de

Strukturen

- Mit Strukturen lassen sich unterschiedliche Daten zu einem Datensatz zusammenfassen
 - Weinflasche
 - Lage
 - Jahrgang
 - AP.-Nr.
 - Inhalt, ...
 - Personaldaten
 - Name
 - Alter
 - ...
- Strukturen definieren neue komplexe Datentypen

14.12.2003

2

struct

- Deklaration:
 - Eine Struktur wird vereinbart indem zuerst das Schlüsselwort struct gefolgt von dem neuen Strukturnamen genannt wird. Dann folgt, eingeschlossen in geschweifte Klammern {} und durch Semikolons getrennt, die Liste der zugrunde liegenden Datentypen und Feldnamen.
- BNF *:
<Struktur> ::= struct [<Strukt-Name>]{<Einzeldat-Liste>} [<Var-Liste>];
<Einzeldat-Liste> ::= <typ> <Namensliste>[<Einzeldat-Liste>]

* vereinfacht

14.12.2003

3

struct

- Beispiele:
 - Weinflasche

```
struct wein {
    char Lage[64];
    int Jahr;
    int APNr;
    float Inhalt;
};
```
 - Ein Datum:

```
struct datum {
    int tag, monat, jahr;
    char wochentag[15];
    char monatsname[15];
};
```

14.12.2003

4

Variablen Deklaration

■ Deklaration von Variablen

- Direkt bei der Vereinbarung der Struktur

```
struct datum {  
    int tag, monat, jahr;  
    char wochentag[15];  
    char monatsname[15];  
} heute, morgen;
```

- Analog zu sonstigen Variablen

```
struct datum geburtstag;
```

Variablen Initialisierung

■ Initialisierung

- Ähnlich wie bei Arrays werden die Werte in einer Liste aufgezählt
- Die einzelnen Elemente werden durch Kommata getrennt
- Die Initialisierungsliste beinhaltet je nach Struktur unterschiedliche Datentypen
- Beispiel:

```
struct datum hAbend = {24,12,02,"Dienstag","Dezember"};  
struct wein binger_st = {"B.St.Rochus",1998,12345,0.75};
```

Nutzung von Variablen

- Um auf die Elemente einer Struktur zugreifen wird der Punkt-Operator benutzt:

```
struct datum {  
    int tag, monat, jahr;  
    char wochentag[15];  
    char monatsname[15]; } hAbend;  
int letztesJahr;  
  
hAbend.tag = 24;  
hAbend.monat = 12;  
hAbend.jahr = 2002;  
strcpy(hAbend.wochentag,"Dienstag");  
strcpy(hAbend.monatsname,"Dezember");  
letztesJahr = hAbend.jahr - 1;
```

Alternative Deklaration mittels typedef

- Um nicht immer das Schlüsselwort **struct** schreiben zu müssen kann man das bereits bekannte Verfahren mittels **typedef** benutzen:

```
/* statt      struct datum {  
                int tag, monat, jahr;  
                char wochentag[15];  
                char monatsname[15];}  
wird dann folgendes geschrieben: */  
typedef struct {  
    int tag, monat, jahr;  
    char wochentag[15];  
    char monatsname[15];} t_datum;  
t_datum heute,morgen; //statt struct datum heute,morgen; //
```

Beispiel

- Entwerfen Sie Datentypen zur Personenverwaltung an einer Fachhochschule
 - Studenten
 - Dozenten
 - FH-Bedienstete
- Identifizieren Sie Gemeinsamkeiten und Unterschiede zwischen den Gruppen

Klausuraufgabe (Prog1 SS02)

- Entwerfen Sie eine Datenstruktur mit der Bezeichnung Artikel, die folgende Daten aufnehmen kann:
 - Artikelname (maximal 25 Zeichen)
 - Artikelnummer (eindeutige Identifikationsnummer bei maximal 5000 verschiedenen Artikeln)
 - Nettobetrag (maximal 100.000 Euro, mit zwei Nachkommastellen)
 - Mehrwertsteuer (Zahlenwert zwischen 0,0 und 1,0)
 - Bestand (maximal 5000)
- Entwerfen Sie eine Funktion warenwert(), die ein Feld mit 1000 Artikeln gemäß Ihrer obigen Datenstruktur übergeben bekommt und die als Ergebnis den Gesamtwert aller Waren inklusive Mehrwertsteuer zurückgibt.