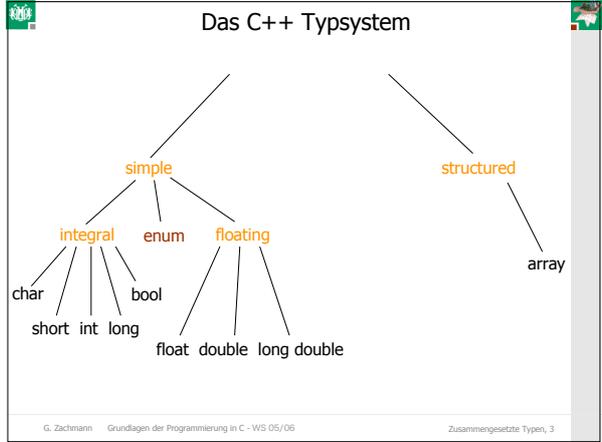


Grundlagen der Programmierung in C++ Zusammengesetzte Typen

Wintersemester 2005/2006
G. Zachmann
Clausthal University, Germany
zach@in.tu-clausthal.de



Aufzählungen (enumeration)

- Einfachste Form eines benutzerdefinierten Typs
- Englisch "enumeration" → keyword `enum`
- Bsp.:

```

enum ColorE
{
    RED, GREEN, BLUE
};

ColorE screen_color = BLUE;
ColorE window_color = RED;

int n = BLUE; // OK
ColorE c = 1; // error!
  
```

G. Zachmann Grundlagen der Programmierung in C - WS 05/06 Zusammengesetzte Typen, 4

Beispiel: Zustandsautomat

Beispiel Ampel:

```

enum ColorE { RED, YELLOW, GREEN };
ColorE color; // current state
...
switch ( color )
{
    // switch to next state
    case RED: color = YELLOW; break;
    case YELLOW: schalte Rot und Gelb aus; color = GREEN; break;
    case GREEN: schalte Grün aus; color = RED; break;
    default: fprintf(stderr, "BUG ...");
}
schalte color ein;
...
  
```

G. Zachmann Grundlagen der Programmierung in C - WS 05/06 Zusammengesetzte Typen, 5

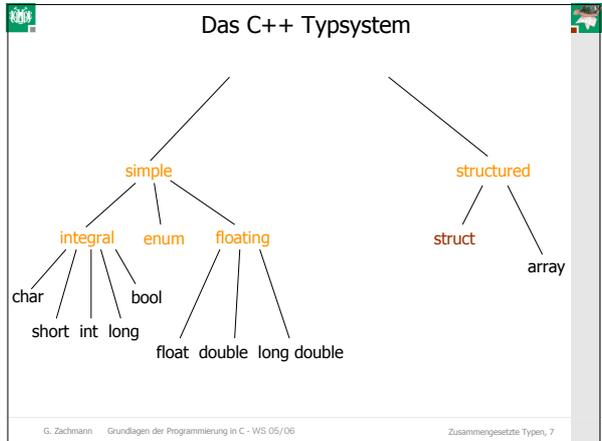
- Was *nicht* geht mit enum's in C++ (ging in C):
 - Rechnen, z.B. `enum++`
- Bestimmung der Anzahl der Enum-Namen:

```

enum ColorE
{
    RED,
    YELLOW,
    GREEN,
    NUM_COLORS
};
  
```

- Funktioniert, weil:
 - Enum-Namen werden fortlaufend numeriert
 - Erster Enum-Name = 0

G. Zachmann Grundlagen der Programmierung in C - WS 05/06 Zusammengesetzte Typen, 6



Struct

- Elemente verschiedenen Typs zusammenfassen zu einem Neuen
- Beispiel: Name, Alter, Größe, ..., einer Person
- Sprach-Feature: "zusammengesetzter Typ"
- Definition:


```
struct NewName
{
    T1 M1;
    T2 M2;
    ...
};
```

wobei T_1, T_2, \dots , bekannte Typen sind,
und M_1, M_2, \dots , die sogenannten *Member*.

G. Zachmann Grundlagen der Programmierung in C - WS 05/06 Zusammengesetzte Typen, 8

- Beispiel:


```
struct Person
{
    unsigned int ID;
    unsigned int age;
    float salary;
    ColorE color;
};
Person person;
Person p2 = { 007, 99, 1000000.0, black };
```
- Benutzung:


```
p2 = person; // assignment
person.age = 17; // member access
printf("%f ", person.color );
```

Member access operator

G. Zachmann Grundlagen der Programmierung in C - WS 05/06 Zusammengesetzte Typen, 9

- Mathematisches Äquivalent: Produkt
Sei S Struct mit Members der Typen T_1, \dots, T_n ,
dann ist der Wertebereich von S

$$W(S) = \prod_i W(T_i)$$

G. Zachmann Grundlagen der Programmierung in C - WS 05/06 Zusammengesetzte Typen, 10

Komposition

- Structs können wieder Structs enthalten
- Vorteil: "Modularisierung" der Datentypen
- Beispiel: Maschinen-Record
 - Datum der Anschaffung
 - Preis
 - Name
 - Statistik der Ausfälle

G. Zachmann Grundlagen der Programmierung in C - WS 05/06 Zusammengesetzte Typen, 11

```
struct Date
{
    int month;
    int day;
    int year;
};

struct Statistics
{
    float failRate;
    Date lastServiced; // composition
    int downDays;
};

struct MachineRec
{
    int id;
    string name;
    Statistics history; // composition
    Date purchaseDate; // composition
    float cost;
};

MachineRec machine;
machine.history.lastServiced.year = 1999;
```

G. Zachmann Grundlagen der Programmierung in C - WS 05/06 Zusammengesetzte Typen, 12

MachineRec

Statistics Date

Date

5719 "PC..." .02 1 25 1999 4 3 21 1995 8000.0

.failrate .lastServiced .downDays .month .day .year

.id .name .history .purchaseDate .cost

machine.history.lastServiced.year

G. Zachmann Grundlagen der Programmierung in C - WS 05/06 Zusammengesetzte Typen, 13