

## Vorlesung „Grundlagen der Informatik und Programmierung 1“

# DATENSTRUKTUREN

## Zeiger, Speicher und Arrays

Prof. Dr. Tom Vierjahn

Visual Computing (<https://vc.w-hs.de>)

Fachbereich Wirtschaft und Informationstechnik – Campus Bocholt



Wintersemester 2020/21



Veröffentlicht unter der Creative-Commons-Lizenz

Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

## Deklaration:

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

- ▶ Die Deklaration erzeugt einen neuen Datentyp (hier struct Date).

- ▶ Es können Zeiger auf Datenstrukturen angelegt und verwendet werden.

```
struct Date today = {14, 11, 2020};
```

```
struct Date* today_pointer = &today;  
(*today_pointer).day = 15;  
today_pointer->month = 12;
```

```
printf("%02d.%02d.%4d\n",  
       (*today_pointer).day, (*today_pointer).month,  
       today_pointer->year);
```

- ▶ Der Adressoperator & ermittelt die Adresse einer Datenstruktur.
- ▶ Der Dereferenzierungsoperator \* gibt Zugriff auf die referenzierte Struktur.
- ▶ Der Points-Operator -> gibt Zugriff auf die Datenfelder einer referenzierten Struktur.

- ▶ Datenstrukturen können per Parameter an Funktionen übergeben werden.
- ▶ Funktionen können Datenstrukturen zurückgeben.
- ▶ Die Daten werden kopiert.

```
void print_date(struct Date d) {  
    printf("%02d.%02d.%4d\n", d.day, d.month, d.year);  
}  
struct Date query_date() {  
    struct Date d;  
    scanf("%d.%d.%d", &d.day, &d.month, &d.year);  
    return d;  
}
```

```
struct Date some_date = query_date();  
print_date(some_date);
```

- ▶ Datenstrukturen können per Zeiger-Parameter an Funktionen übergeben werden.
- ▶ Funktionen können Datenstrukturen per Rückgabeparameter zurückgeben.
- ▶ Lediglich der Zeiger wird kopiert.

```
void print_date(struct Date* d) {  
    printf("%02d.%02d.%4d\n", d->day, d->month, d->year);  
}
```

```
void query_date(struct Date* d) {  
    scanf("%d.%d.%d", &d->day, &d->month, &d->year);  
}
```

```
struct Date some_date;  
query_date(&some_date);  
print_date(&some_date);
```

## Grundlagen der Informatik und Programmierung 1

Fundamentals of Informatics and Programming 1

<b>Kürzel:</b>	GIP1	<b>Workload:</b>	210 h	<b>Leistungspunkte:</b>	7
<b>Semester:</b>	1	<b>Dauer:</b>	1 Semester	<b>Häufigkeit:</b>	Regelmäßig im Wintersemester

```
struct ModuleA {  
    char title[64];  
    char abbreviation[5];  
    int workload;  
    char credits;  
    char semester;  
    char duration;  
};
```

```
struct ModuleB {  
    char title[64];  
    char abbreviation[5];  
    char credits;  
    char semester;  
    char duration;  
    int workload;  
};
```

```
printf("sizeof(ModuleA): %lu\n", sizeof(struct ModuleA));  
printf("sizeof(ModuleB): %lu\n", sizeof(struct ModuleB));
```

# Alignment

0x	00	...	3f	40	...	44	45	46	47	48	49	4a	4b	4c	4d	4e	4f
M.A																	
M.B	title			abbrev.			c	s	d	workload							

- ▶ Datenstrukturen können in Arrays angeordnet werden.
- ▶ Die Datenobjekte liegen hintereinander im Speicher.

```
struct ModuleB core_subjects_is[20] = {  
    {"Algorithmen und Datenstrukturen", "ALG", 6, 3, 1, 180},  
    {"Arbeitstechniken 1", "AT1", 2, 1, 1, 60},  
    {"Arbeitstechniken 2", "AT2", 2, 2, 1, 60},  
    ...  
};
```


## Speicher:





- ▶ Zeiger auf Datenstrukturen
- ▶ Datenstrukturen als Funktionsparameter
- ▶ Speicherbedarf und Alignment
- ▶ Arrays aus Datenstrukturen

Prof. Dr. Tom Vierjahn

▶  tom.vierjahn@w-hs.de

Visual Computing

▶  <https://vc.w-hs.de>

▶  VisualComputingWH

▶  Visual Computing WH

▶  @VisComputingWH

Westfälische Hochschule

Fachbereich Wirtschaft und Informationstechnik

Campus Bocholt



Veröffentlicht unter der Creative-Commons-Lizenz

Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)