

# Computeranwendung in der Chemie Informatik für Chemiker(innen)

## 7. Organisation von Informationen

# Grundlagen

Programme verarbeiten Daten

⇒ Daten müssen gespeichert werden

- Einfache Datentypen
- Zusammengesetzte Datentypen
- Zeiger
- Felder
- Listen

# Datentypen in C

Variablentyp	Keyword	Byte	Bereich
character	<code>char</code>	1	-128 – 127
integer	<code>int</code>	4	-2147483648 – 2147483647
short integer	<code>short</code>	2	-32768 – 32767
long integer	<code>long</code>	4	-2147483648 – 2147483647
unsigned character	<code>unsigned char</code>	1	0 – 255
unsigned int	<code>unsigned int</code>	4	0 – 4294967296
unsigned short	<code>unsigned short</code>	2	0 – 65536
unsigned long	<code>unsigned long</code>	4	0 – 4294967296
single precision	<code>float</code>	4	1.2E-38 – 3.4E38
double precision	<code>double</code>	8	2.2E-308 – 1.8E308

# Datentypen in C

Tatsächliche Größe von Datentypen abhängig von Compiler

- Character: normalerweise 8 Bit (selten 16 Bit)
- Integer: Registerbreite der CPU (PC: 32 Bit)
  
- Single precision: normalerweise 32 Bit
- Double precision: normalerweise 64 Bit

# Zeiger

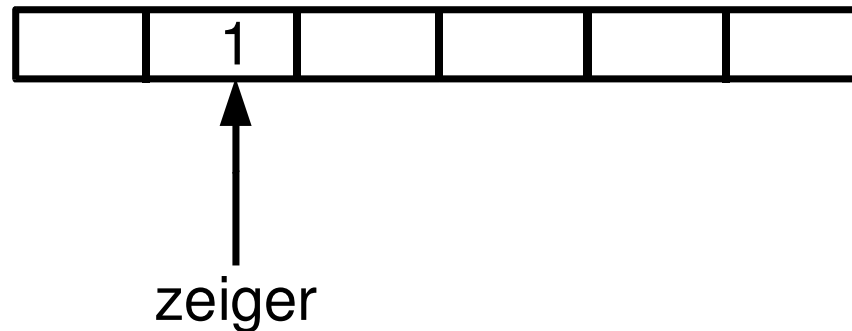
Zeiger (engl. pointer) sind Werte, die auf eine Adresse im Speicher verweisen

- Zeiger beziehen sich auf den virtuellen Adressraum des Prozesses
- Zeiger haben einen Typ
  - Typ gibt Art des Inhalts des Speichers an
  - Beispiel für Deklaration: `int *zeiger;`
  - Generischer Typ: `void *`

# Zeiger

Speicheradresse einer Variable kann durch den  
Addressoperator & ermittelt werden

```
int i, *zeiger;  
i = 1;  
zeiger = &i;
```



Auflösung eines Zeigers durch Dereferenzoperator \*

```
*zeiger == 1 (Inhalt von *zeiger ist 1)
```

# Felder

Felder (engl. arrays) sind ein- oder mehrdimensionale Datenstrukturen mit Daten gleichen Typs

Beispiele:

- Vektor: eindimensional
- Matrix: zweidimensional

# Felder

Größe muß bei Deklaration angegeben werden

⇒ Statische Speicherbelegung

Beispiel:

```
int vektor[3];
```

```
double matrix[4][5];
```

Zugriff: Index in [], z. B. `vektor[1] = 1;`

Index beginnt bei 0!



# Dynamische Speichieranforderung

Statische Speichieranforderung oftmals unbefriedigend, da bei Programmentwicklung Speicherbedarf unbekannt

- Dynamische Speichieranforderung per Bibliotheksfunktion: `malloc()` (Memory Allocation)
- `malloc()` gibt Zeiger auf Speicherbereich zurück
- Angeforderter Speicher muß nach Benutzung wieder freigegeben werden: `free()`

# Strukturen

Variablen können zu Strukturen zusammengefaßt werden

Beispiel: Punkt in zweidimensionalen Raum, x und y

```
struct point {
```

```
double x;
```

```
double y;};
```

```
struct point ursprung;
```

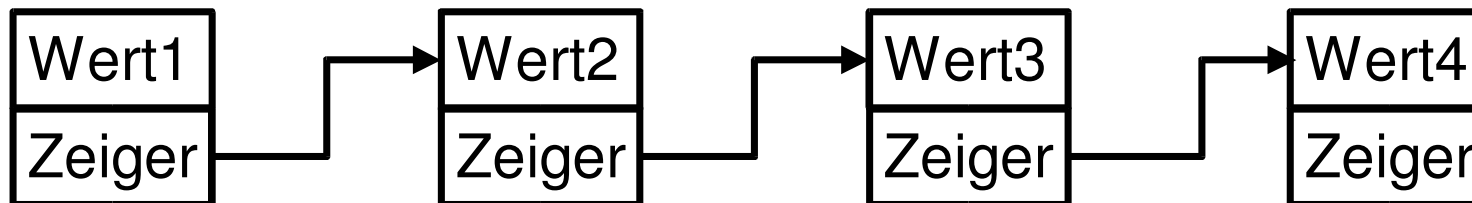
```
ursprung.x = 0; ursprung.y = 0;
```

Deklaration  
der Struktur  
point

Deklaration  
von ursprung

# Verkettete Liste

- Liste besteht aus Elementen
- Jedes Element enthält Wert und Zeiger auf nächstes Element
- Einfaches Einfügen und Entfernen
- Einfaches Speichermanagement: Für jedes Element wird dynamisch Speicher angefordert.



# Verkettete Liste

Beispiel für Implementierung:

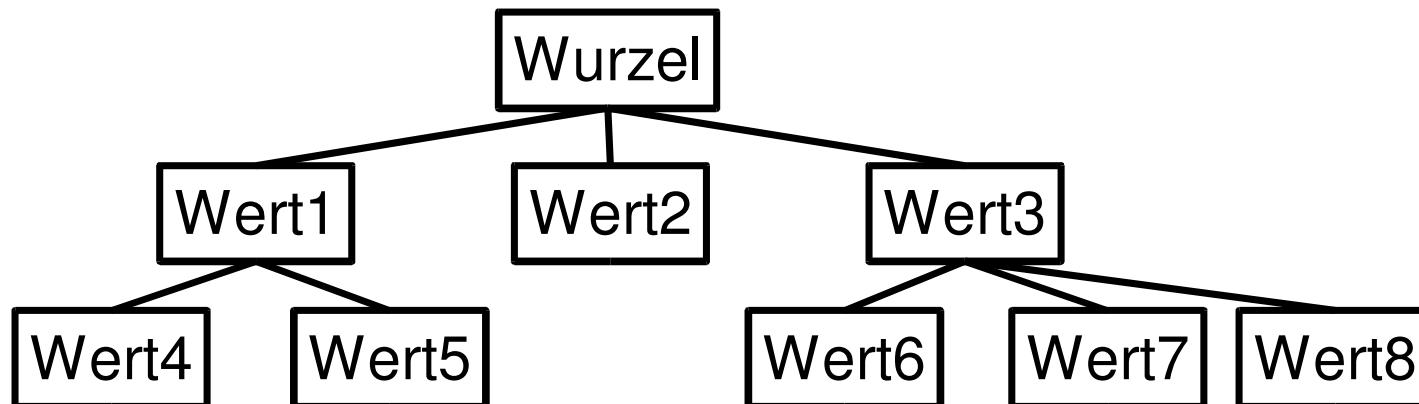
```
struct elem
{
    int wert;
    struct elem *next;
};
```

Zeiger next zeigt auf das nächste Listenelement.

Listenende: next=NULL

# Baum

- Ähnlich der verketteten Liste
- Unterschied: Verzweigt statt linear
- Anwendung: Hierarchische Ablage von Informationen
- Schneller zu durchsuchen als Liste



# Baum

- Im allgemeinen Fall: beliebig viele Verzweigungen von einem Knoten
  - Anzahl der Zeiger auf folgende Knoten variabel
  - Speicherung von Zeigern auf folgende Knoten als verkettete Liste möglich
- Spezialfall: Feste Anzahl von Verzweigungen
  - Binärbaum: Jeder Knoten verzweigt zu zwei Elementen
  - Jeder Schritt durch ein Bit darstellbar

# Darstellung von Grafiken

- Darstellung als zweidimensionales Feld von Werten
  - Rastergrafik, „Bitmap“
  - einfach zu erstellen und zu bearbeiten
  - schlecht größen-skalierbar
- Beschreibung in Vektordarstellung
  - Aufbau durch Punkte, Linien, Kreise, etc.
  - gut größen-skalierbar

# Rastergrafik

- Grafik wird in Feld von Punkten eingeteilt
- Punkte: Pixel (Kunstwort aus Picture Element)
- Jeder Pixel: Ein Wert (Farbe des Pixels)
- Normalerweise Speicherung von Rot-, Grün- und Blauanteilen (RGB, additive Farbmischung)
  
- 1 Bit pro Pixel (Schwarz oder Weiß)
- 8 Bit pro Pixel (256 Farben, Index für Tabelle)
- 24 Bit pro Pixel (je 8 Bit für RGB)



# Vektorgrafik

- Darstellung aus geometrischen Elementen
- Sinnvoll für Strichdarstellungen, z. B.
  - Strichzeichnungen
  - Diagramme
- Darstellung: Erzeugung von Rastergrafik

Beispiel für Vektorformat:

PostScript

- Beschreibung von Seite für Ausgabe auf Drucker