

# Fachbereich Medienproduktion

- Herzlich willkommen zur Vorlesung im Studienfach:
  - Grundlagen der Informatik

# Themenübersicht

- Grundlagen der Informatik
  - Grundlagen der Rechnertechnik
  - Algorithmen und Datenstrukturen
    - Primitive Datentypen
    - Abstrakte Datentypen
    - Algorithmen
  - Rechnernetze und das Internet
  - IT Sicherheit

# Arrays

- Haltung von n-Datenwerte des gleichen Typs
- Daten liegen im Speicher hintereinander
- Zeichenketten (Strings) sind char-Arrays

## Beispiel 1

```
int daten[n];           // Deklaration
int i;

for (i=0; i<n; i++)    //Zugriff per Iterator-Schleife
{
    daten[i] = 0;      //Zugriff auf ein Datenelement
}
```

## Beispiel 2

```
char wort[5];          // Deklaration

wort[0] = 't';
wort[1] = 'e';
wort[2] = 's';
wort[3] = 't';
wort[4] = 0;          // „test“
```

# Klassen

- Abstrahiert komplexe Datenhaltung und Verarbeitung
- Dem Nutzer werden definierte Methoden zur Verfügung gestellt -> API (Applications Programming Interface)
- Ermöglicht Erweiterungen, bisheriges Verhalten bleibt unverändert
- Fertige Klassen in Bibliotheken (libraries)
- Eigene Klassen können neu erstellt oder abgeleitet werden

Beispiel: Akkumulator Klasse

Beschreibung	API / Methoden
Addition von Eingabewert	<code>addValue( int x)</code>
Ausgabe der Summe	<code>int getSum()</code>
Ausgabe des Durchschnitts	<code>int getMean()</code>

# Beispiel Akkumulator Klasse

```
class Akkumulator
{
    private int total=0;    //interne Klassenvariablen
    private int n=0;

    public void addValue(int x) //öffentliche Methode (API)
    {
        n++;
        total = total + x;
    }

    public int getSum()      //öffentliche Methode (API)
    { return total; }

    public int getMean()    //öffentliche Methode (API)
    { return total/n; }
}
```

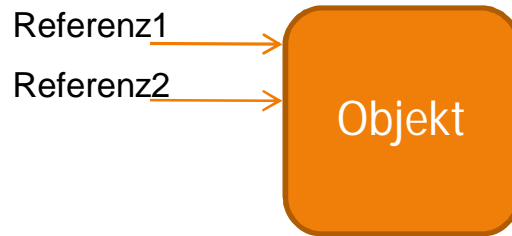
# Referenzen / Pointer

- Referenz zeigt auf das eigentliche Objekt
- Dadurch keine Kopie des Objektes notwendig
- Mehrere Referenzen auf ein Objekt möglich

Beispiel Instanziierung einer Klasse

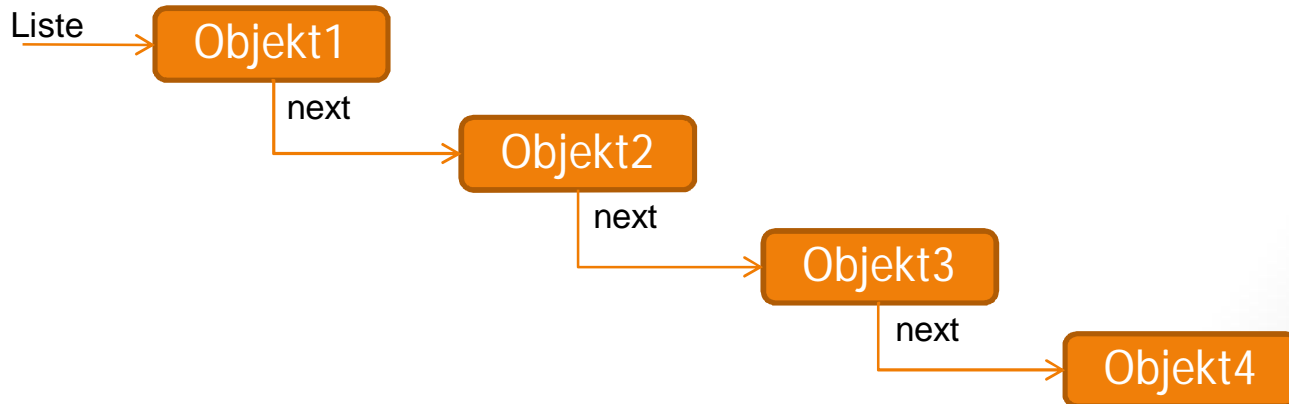
```
Referenz1 = new(Objekt)
```

```
Referenz2 = Referenz1;
```



# Verkettete Listen

- Dynamische Verwaltung von Objekten
- Referenzen zeigen auf „next“ Objekt
- Umsortieren/Einfügen/Entfernen ohne Umkopieren der Daten
- Zugriff auf Elemente nur durch Verfolgung der Kette möglich



- Vielen Dank für Ihre Aufmerksamkeit!