

# Angewandte Mathematik und Programmierung

Einführung in das Konzept der objektorientierten Anwendungen zu  
mathematischen Rechnens

WS 2012/13

Fomuso Ekellem



# Inhalt

- **Überschreiben**
- **Überladen**
- **Operatorüberladung**



# Überschreiben-Wiederholung

- In C++ sind 2 Arten von der Überschreibung
- Durch Kennzeichen der Methodendeklaration mit **virtual** kommt der **dynamische** Typ zum Einsatz
- Standardmässig wird der **statische** Typ zur Ermittlung der zu rufende Methode herangezogen.
- Abstrakte Methoden Anhängen von =0; an die Deklaration

# Polymorphie und Methoden überschreiben

## Wiederholung...

```
class Auto
{
public:
    virtual void marke() = 0;
};
```

```
class DeutschesAuto : public Auto{
public:
    virtual void marke(){cout << "VW" << endl;}
    void fahre250() {cout << "geht nicht" << endl;}
};
```

```
class Audi : public DeutschesAuto{
public:
    virtual void marke(){ cout << "Audi" << endl;}
};
```

```
class Mercedes : public DeutschesAuto{
public:
    virtual void marke(){ cout << "Mercedes" << endl;}
    void fahre250(){ cout << "sicher!" << endl;}
};
```

Identischer Methoden kopf heißt, dass sowohl **Sichtbarkeit, Rückgabewert, Methodename** und die **Datentypen der Methodenparameter** gleich sein müssen.

Beim Auto-Beispiel wird die Methode `marke` von `Auto` in `DeutschesAuto` überschrieben, welche dann wiederum in den Klassen `Audi` und `Mercedes` überschrieben werden.

`fahre250` ist zum ersten mal in `Deutsches Auto` definiert und wird in der Klasse `Mercedes` überschrieben, nicht aber in der Klasse `Audi`, und ist nicht virtuell.



# Überladen

- Methoden können überladen werden
- Methoden mit gleichen Namen müssen sich in den Typen der Parameter unterscheiden.
  
- Operatoren sind auch Methoden
- Operatoren werden nur anders geschrieben
  - a+b    anstatt +(a,b)
  - und
  - +a      anstatt +(a)
- Man kann in C++ für eigene Typen eigen Implementierung der meisten Operatoren angeben(aka Operatorüberladung):

# Operatorüberladung

- Gewöhnliche Operatoren können überladen werden.
- Man versteht schon:  
 $1+2 = 3.$
- Aber wenn wir „+“ in einer Klasse überladen, sieht es so etwa aus:

```
Class MyClass{
private:
int x, y;
public:
MyClass();
MyClass Operator +(MyClass obj1){
MyClass temp;
temp.x = x + obj1.x;
temp.y = y + obj1.y;
return temp;
}
};
```

- In der Main-Methode kann man zwei Objekte von typ MyClass mit „+“ addieren.
- Siehe Beispiele....