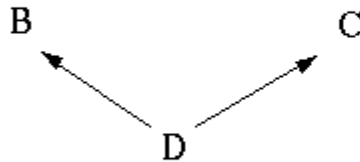


# Mehrfachvererbung

- erleichtert Wiederverwendung -> Funktionen mehrerer Basisklassen können genutzt werden



```
class B {      // ...  
public:      virtual void f (); { }  
};  
  
class C {      // ...  
public:      virtual void f (); { }  
};  
  
class D: public B, public C { // ...  
  
};
```

- **Auflösung von Namenskonflikten**

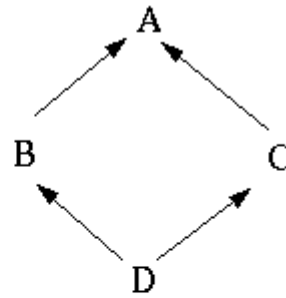
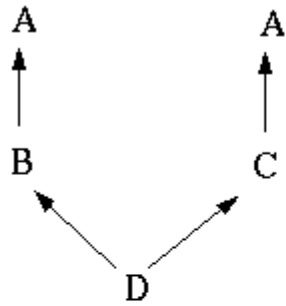
- explizite Klassenangabe beim Funktionsaufruf, z.B.

o.B::f();

- Overriding in abgeleiteter Klasse

- **wird Klasse D mehrfach (indirekt) von derselben Basisklasse A abgeleitet, sind die A-Komponenten mehrfach in D-Objekten repräsentiert**
- **Ausnahme: Verwendung virtueller Basisklassen**

virtuell abgeleitete Klasse D teilt sich Daten der virtuellen Basisklasse A mit anderen Klassen, die von derselben Basis A virtuell abgeleitet sind



```
class A { /* ... */ };  
class B: virtual public A { /* ...*/ };  
class C: virtual public A { /* ...*/ };  
class D: virtual public B,  
        virtual public C { /* ... */ };
```