# Vererbung in mehreren Generationen, Vererbung bei gekapselten Attributen

Vererbung kann in mehreren Generationen stattfinden:

Klasse A vererbt an Klasse B u. diese wieder an Klasse C.

Vererbung wird laut UML mit einem Pfeil mit durchgezogener Linie

aber nicht gefüllter Spitze dargestellt.

Es entstehen Vererbungshierarchiediagramme:

**C**

**B**

**A**

Werden in den Klassen A, B u. C Attribute a, b u. c als "public" definiert,

in UML durch Voranstellen des Pluszeichens gekennzeichnet,

sind in A nur a,

in B: a u. b,

in C: a u. b u. c sichtbar:

Sichtbarkeit:

**C**

+ c: int

**B**

+ b: int

**A**

+ a: int

a

a, b

a, b, c

Gekapselte Attribute werden nicht vererbt.

Sie werden in UML mit einem Minuszeichen dargestellt:

Sichtbarkeit:

**A**

+ a: int  
- private\_a: int

a, privat\_a

**B**

+ b: int  
- private\_b: int

a, b, privat\_b

nicht: private\_a

**C**

+ c: int  
- private\_c: int

a, b, c, privat\_c

nicht: private\_a oder private\_b

Im Quelltext:

**public** **class** A {

**public** **int** a;

**private** **int** privat\_a;

**public** **void** zugriff\_in\_a() {

System.***out***.println(a);

System.***out***.println(privat\_a);

}

}

**public** **class** B **extends** A{

**public** **int** b;

**private** **int** privat\_b;

**public** **void** zugriff\_in\_b() {

System.***out***.println(a);

// System.out.println(private\_a);

// Zugriff auf private\_a nicht möglich!

System.***out***.println(b);

System.***out***.println(privat\_b);

}

}

**public** **class** C **extends** B {

**public** **int** c;

**private** **int** privat\_c;

**public** **void** zugriff\_in\_c() {

System.***out***.println(a);

// System.out.println(private\_a);

// Zugriff auf private\_a nicht möglich!

System.***out***.println(b);

// System.out.println(privat\_b);

// Zugriff auf private\_b nicht möglich!

System.***out***.println(c);

System.***out***.println(privat\_c);

}

}