

OOO-Glossar

Basisklasse	<pre>class BASIS { virtuell int vfkt(int); };</pre>	Eine Klasse, von der andere Kind-Klassen abgeleitet werden. Eine abstrakte Basisklasse enthält zwar eine Bauvorschrift, jedoch keine ausformulierten Elementfunktionen. Daher kann von einer abstrakten Basisklasse nur geerbt werden. Virtuelle Funktionen implementieren Polymorphie in C++.
Basisklassen-Pointer	<pre>BASIS *b; b = new KIND; b->vfkt();</pre>	Ein Zeiger auf eine Basisklasse kann auch auf alle Kind-Klassen zeigen. Ruft ein Basisklassenpointer eine Methode auf, wird nicht die Methode der Basisklasse sondern die Methode der Kind-Klasse gerufen, wenn die Methode als virtuell deklariert war.
Bereichs-Bezeichner	<pre>private: protected: public:</pre>	Kennzeichnet einen Bereich in der Klassendeklaration, der geschützt (gekapselt) ist oder für andere Programme zugänglich ist.
C++		Die erfolgreichste Sprache für objektorientiertes Programmieren. Programme für C++ und C können gemischt werden. Auch ohne Verwendung der Objektorientierung bereichern die Erweiterungen von C++ die Möglichkeiten von C.
Container		Datentyp (Klasse), der eine beliebige, nicht von vornherein bekannte Anzahl von Objekten aufnehmen kann. Dabei können die Objekte direkt oder indirekt (über Pointer), sortiert oder unsortiert, mit oder ohne Wiederholung gespeichert werden. Typische Container sind Listen oder Arrays.
Definition	<pre>int x; NAME n; int fkt(int a) {...}</pre>	Die Definition einer Variablen oder einer Funktion ist eine konkrete, verwendbare Instanz eines Typs oder einer Funktion.
Deklaration	<pre>extern int x; struct NAME {...}; class NAME {...}; int fkt(int);</pre>	In Deklarationen werden Variablentypen oder Funktionen beschrieben. Deklarationen benötigen keinen Speicherplatz. Eine Klassenbeschreibung ist eine Deklaration.
Destruktor	<pre>class NAME {... ~NAME(); }</pre>	Ein Destruktor ist eine Funktion, die gerufen wird, wenn der Gültigkeitsbereich eines Objekt abläuft. Bei globalen Objekten nach Ablauf von <code>main()</code> , bei lokalen (automatischen) Objekten nach Beendigung der Funktion, bei dynamischen Objekten bei Aufruf von <code>delete</code> . Eine wichtige Aufgabe des Destruktors ist das Löschen von Speicherbereichen am Heap.
Dynamische Objekte	<pre>new delete</pre>	Dynamische Objekte werden am Heap mit dem Operator <code>new</code> angelegt. Der Operator <code>delete</code> löscht das Objekt wieder.
Eigenschaft (property)	<pre>class NAME {... int eigen; };</pre>	Eine Variable, die innerhalb einer Klasse definiert ist, nennt man eine Eigenschaft dieser Klasse. Je nach Bereichsbezeichner (<code>public</code> , <code>protected</code> , <code>private</code>) kann eine Eigenschaft öffentlich zugänglich, für abgeleitete Klassen zugänglich oder gegen Zugriff geschützt sein.
Elementfunktion (Methode)	<pre>class NAME {... int elem(int); };</pre>	Eine Elementfunktion oder Methode ist eine Fähigkeit die mit den Eigenschaften einer Klasse arbeitet. Je nach Bereichsbezeichner (<code>public</code> , <code>protected</code> , <code>private</code>) kann eine Elementfunktion öffentlich zugänglich, für abgeleitete Klassen zugänglich oder gegen Zugriff geschützt sein.
Gültigkeitsbereich-zugriffsoperator	<code>::</code>	Mit dem neuen Operator <code>::</code> (Scope Resolution Operator) kann der Speicherort einer Variablen oder Funktion durch voranstellen der Klassen oder Objektbezeichnung angegeben werden. (z.B. <code>KLASSE::play()</code>). Entfällt der Klassename, erreicht man globale Größen auch dann, wenn sie durch eine gleichnamige lokale Größe verdeckt werden.
Inline-Funktion		Eine Funktion heißt Inline-Funktion, wenn sie beim Compilieren nicht durch einen Funktionsaufruf mit dem rufenden Programm verbunden wird sondern der Code direkt beim Funktionsaufruf eingefügt wird. In C++ sind alle in der Klassendeklaration definierten Funktionen automatisch Inline-Funktionen
Kapselung (data hiding)	<pre>private, protected, public</pre>	Eigenschaft der Bereichsbezeichner <code>protected</code> und <code>private</code> , die Zugriff auf Variablen und Funktionen von außen verhindern.
Klasse	<code>class</code>	Durch das Zusammenfassen von Variablen und Funktionen entsteht eine Klasse. Klassen und Strukturen (in der C++-Version) unterscheiden sich nur durch den Grad der Öffentlichkeit: Variablen in Klassen sind nicht öffentlich, es sei denn, sie sind durch den Bereichsbezeichner <code>public</code> als solche gekennzeichnet worden.
Konstruktor	<pre>class NAME NAME::NAME();</pre>	Ein Konstruktor ist eine Funktion, die im Zuge der Definition eines Objekts gerufen wird. Ein Konstruktor hat denselben Namen wie die Klasse. Er initialisiert die Variablen der Klasse. Bei Objekten mit dynamischen Speicheranteilen initialisiert er die Variablen am Heap.
Mehrfach-Vererbung	<pre>class KIND:public E1, E2</pre>	Eine Klasse kann die Eigenschaften mehrerer Elternklassen erben.
Objekt		Die Instanz einer Klasse. Im traditionellen Programmierstil entspricht das Objekt einer Variablen, die die Bauform einer Klasse hat. Von einer Klasse können beliebig viele Objekte gebildet werden.
Polymorphie		Fähigkeit, Objekte verschiedenartiger Detaileigenschaft aber mit gleichen Methoden bearbeiten zu können. Beispiel: Bewegung verschiedener Figuren am Bildschirm mit derselben Funktion.
Struktur	<code>struct</code>	Zusammengehörige Variablen können unter einem eigenen Namen angesprochen werden. C++: <code>Strukturname=Typname</code> . Die Deklaration einer Struktur entspricht einer Typvereinbarung. In C++ können Strukturen auch Funktionen enthalten; man nennt sie dann eine Klasse. In einer Struktur sind Daten grundsätzlich öffentlich, es sei denn, sie wurden durch den Bereichsbezeichner <code>private</code> oder <code>protected</code> als geschützt gekennzeichnet.
Typ		Bauvorschrift für eine Variable. Eine Struktur ist die Bauvorschrift für eine aus mehreren Typen zusammengesetzte Anordnung. Eine Klasse definiert darüberhinaus auch einen Zugriffsschutz. (Kapselung)
Vererbung (inheritance)	<pre>class KIND public ELTERN</pre>	Vererbung ermöglicht die Bildung neuer Klassen, die alle Eigenschaften einer Eltern-Klasse übernehmen können. Die Vererbung kann <code>public</code> oder <code>private</code> erfolgen. Kind-Klassen können nur auf Variablen und Funktionen in den Bereichen <code>public</code> und <code>protected</code> der Eltern-Klasse zugreifen.