Kapitel 2

Einführung in C++

C++ Zeichensatz

- Buchstaben: **a** bis **z** und **A** bis **Z**.
- Ziffern: **0** bis **9**
- Sonderzeichen:

;	:	,	.	‘	„	#	+	-	*	/	%	_	\
!	<	>	|	&	^	~	(	)	{	}	[	]	?

Höhere Elemente

- Mit den im Zeichensatz enthaltenen Zeichen werden folgende Elemente gebildet:
  - Schlüsselwörter
  - Namen
  - Konstanten
  - Operatoren
  - Interpunktionszeichen

Schlüsselwörter

- Schlüsselwörter haben eine durch die Programmiersprache festgelegte Bedeutung.

asm	auto	break	case	catch	char	class	const
continue	default	delete	do	double	else	enum	extern
float	for	friend	goto	if	inline	int	long
new	operator	private	protected	public	register	return	short
signed	sizeof	static	struct	switch	template	this	throw
try	typedef	union	unsigned	virtual	void	volatile	while

Namen

- In einem Programm werden Variablen, Funktionen usw. Namen (Bezeichner) zugeordnet.
- Ein Name beginnt mit einem Buchstaben oder einem Unterstrich.
- Er besteht aus einer Folge von Buchstaben, Ziffern und dem Unterstrich.
- Ein Name hat eine vom Compiler abhängige Länge von z.B. 255 Zeichen.

Namen

- Groß- und Kleinschreibung wird unterschieden.
- Namen von Variablen sollten mit Kleinbuchstaben beginnen.
- Konstanten sollten groß geschrieben werden (z.B.: PI).
- Ein C++ Schlüsselwort darf nicht als Name verwendet werden.

volumen
PI
_anzahl_werte
feldNummer16
FeldNummer16

Konstanten

- Konstanten sind nicht veränderbare Daten innerhalb eines Programms.
- Jeder Konstanten ist eindeutig ein Datentyp zugeordnet.
- Die spezifische Schreibweise der Konstanten hängt von ihrem Datentyp ab.

3
3.14159

Operatoren

- Operatoren dienen der Verknüpfung von Operanden.
- Es sind für jeden Datentyp Operatoren vordefiniert.
- Bsp:

Addition	+
Division	/
Oder	||

Interpunktionszeichen

- Ein Beispiel für ein Interpunktionszeichen ist das Semikolon am Ende jeder Anweisung.

```
a = 1;
cin >> radius;
```

Trennung von Anweisungen

- Es können eine oder mehrere Anweisungen in einer Zeile stehen.
- Am Ende jeder Anweisung **muss** ein Semikolon stehen.

```
a = 1;
b = 2; c = 3;
```

Kommentare

- C++ Kommentare beginnen mit // und reichen bis zum Ende der Zeile.

```
// Dies ist eine Kommentarzeile
a = 1;       // Dies ist ein Kommentar
```

Kommentare

- Kommentare im C-Stil sind auch erlaubt, und dienen meist zur vorübergehenden Ausblendung von Programmcode.
- Dadurch ist das Ausblenden eines ganzen Abschnittes möglich.

```
/* Dies ist ein Kommentar */
/*
    volumen = 4.0 / 3.0 * PI * pow(radius, 3);
    cout << "Volumen = " << volumen << endl;
*/
```

Fortsetzungszeilen

- In C++ ist es möglich, eine Zeile in der darauf folgenden Zeile fortzusetzen.

```
volumen =
        4.0 / 3.0 * PI * pow(radius, 3);
```

Leerzeichen

- Innerhalb von Schlüsselwörtern, Namen, Konstanten (außer Zeichenkonstanten) und Operatoren (z.B. >>) dürfen keine Leerzeichen verwendet werden.

Das „Hallo Welt"-Programm

Zeile	Befehl
1	#include <iostream>
2	using namespace std;
3	void main()
4	{
5	cout << "Hallo Welt "; // Dies ist ein Kommentar
6	}

Das „Hallo Welt"-Programm

- Dieses Programm gibt den Text „Hallo Welt" auf dem Bildschirm aus.
- Ein C++ Programm besteht mindestens aus der globalen Funktion *main*. Sie ist das Hauptprogramm und darf auch nur *main* heißen!
- In **Zeile 1** steht die Präprozessor-Anweisung.
- **Zeile 2** gibt den Namensraum an, in dem die genutzten Befehle zu finden sind. (Der Befehl „cout" ist zu finden im Namensraum „std")
- Ab **Zeile 3** wird die Funktion *main* definiert.

Das „Hallo Welt"-Programm

- **Zeile 4** und **6** begrenzen einen Block. Dieser ist der Funktionsrumpf von *main*.
- Hinter „cout << "Hallo Welt";" steht ein Kommentar
- Kommentare werden mit // eingeleitet. Alles, was hinter den // folgt, wird nicht abgearbeitet.
- Jeder Befehl innerhalb der geschweiften Klammern { } wird durch ein Semikolon abgeschlossen.
  (Bsp.: cout << "Hallo Welt";)

Das zweite C++-Programm

Zeile		Befehl
1	#include <iostream>	// cin/cout sind hier enthalten
2	#include <cmath>	// pow ist hier enthalten
3	using namespace std;	
4	void main()	
5	{	
6	double volumen, radius;	
7	const double PI = 3.14159;	
8	cout << "Radius der Kugel eingeben: ";	
9	cin >> radius;	
10	volumen = 4.0 / 3.0 * PI * pow (radius, 3);	
11	cout << "Volumen = " << volumen << endl;	
12	}	

Das zweite C++-Programm

- Dieses Programm berechnet das Volumen einer Kugel bei gegebenem Radius.
- In **Zeile 1** und **2** stehen die Präprozessor-Anweisungen.
- In **Zeile 3** steht die *using*-Anweisung
- Ab **Zeile 4** wird die Funktion *main* definiert.
- In **Zeile 5** und **12** umschließen den Funktionsrumpf der Funktion *main*.

Das zweite C++-Programm

- In **Zeile 6** werden die Variablen *volumen* und *radius* definiert.
- In **Zeile 7** wird die Konstante **PI** definiert.
- In **Zeile 8** und **11** werden Zeichenfolgen und der Wert von *volumen* ausgegeben.
- In **Zeile 9** steht die Eingabe für die Variable *radius*.
- In **Zeile 10** findet man die Berechnung für *volumen*.
- Die vorhandenen // in Zeile 1 und 2 dienen wieder der Einleitung eines Kommentars.

Blöcke

- Mit Blöcken fasst man mehrere Befehle und Deklarationen zu einer Einheit zusammen.
- Innerhalb eines Blocks können lokale Variablen definiert werden.
- Der Rumpf einer Funktion (zum Beispiel bei *main*) muss ein Block sein.
- Blöcke dürfen beliebig ineinander geschachtelt sein.

Blöcke - Beispiel

```
#include <iostream>
using namespace std;
void main()
{
    int a, b;
    a = 1; b = 2;
    cout << a << b << endl;
    {
        int b = 5, c = 3;
        cout << a << b << c << endl;
        {
            int d = 4;
            cout << a << b << c << d << endl;
        }
    }
}
```

Header-Dateien

- Oberhalb der Funktion *main* werden Header-Dateien eingebunden.

```
#include <iostream>
```

- Diese, mit Doppelgatter # beginnenden Zeilen, sind sogenannte Präprozessoranweisungen.
- Der Präprozessor erzeugt entsprechend der Anweisung den Quelltext, der dann vom Compiler übersetzt wird.
- Am Ende einer Präprozessor-Anweisung steht **kein** Semikolon.

Header-Dateien

- Header-Dateien sind Textdateien, die u.a. weitere Befehle enthalten (zB cout, cin).
- Header-Dateien werden mit der *include*-Anweisung eingebunden, um diese Befehle im Programm nutzen zu können.

Header-Dateien

- Spitze Klammern werden bei Standard-Header-Dateien verwendet.
- Zur Einbindung von selbst verfassten Header-Dateien verwendet man Anführungszeichen.

```
#include <math.h>
#include "Test.h"
```

Header-Dateien

- Alle in einer Header-Datei deklarierten Namen sind global verfügbar.
- Das kann bei großen Programmen zu Namenskonflikten führen.

Header-Dateien

- Daher gibt es in C++ z.B. zur Header-Datei *math.h* noch eine Header-Datei *cmath*, die dieselben Namen in einem Namensbereich *std* deklariert.

```
#include <math.h>
```

entspricht

```
#include <cmath>
using namespace std;
```

Formatierungsregeln für Quelldateien

- In den obigen Beispielen wurden einige Regeln zur Formatierung eines C++ Quellprogramms befolgt:
  - Jeder neue Block wird um eine bestimmte Anzahl an Leerzeichen oder „Tabs" eingerückt.
  - Blockbegrenzungen stehen immer in einer gesonderten Zeile.
  - Jede Funktion beginnt mit der ersten Spalte (steht also ganz links).

Formatierungsregeln für Quelldateien

- Bei längeren Anweisungen sollte der Übersicht halber nicht mehr, als eine Anweisung pro Zeile geschrieben werden.
- Keine Zeile sollte mehr, als 80 Zeichen lang sein, um die Lesbarkeit zu erhöhen.