

# Einführung in die Ein-/Ausgabe von C++

Literatur: Schildt, H.: Teach Yourself C++, Berley: Osborn McGraw-Hill, 1994

## 1 Grundlagen

- C++ E/A-System operiert über **streams**
- **streams** sind logische Geräte, die Informationen produzieren bzw. verbrauchen
- **streams** sind durch das E/A-System mit physikalischen Geräten verbunden
- Standardstreams

stream	Bedeutung	Standardgerät
cin	Standard Eingabe	Tastatur
cout	Standard Ausgabe	Bildschirm
cerr	Standard Fehler	Bildschirm
clog	gepufferte Version von cerr	Bildschirm
- **iostream** stellt Standardfunktionen zur Nutzung des E/A-Systems bereit
- **iostream** ist Teil einer Klassenhierarchie, deren Basisklasse **ios** ist.
- **ios** enthält Memberfunktionen und Variablen, die die grundlegenden Operationen eines Streams steuern bzw. aufzeichnen.

## 2 Formatierte Ein-/Ausgabe

Bisher wurden alle Ein-/Ausgabeoperationen unter Nutzung der C++-Standardformate realisiert. Jeder Stream ist mit einer Anzahl von **flags** verbunden, die bestimmte Standardwerte enthalten:

Name	Beschreibung
skipws	Whitespacebehandlung
left	Linksbündige Ausgabe
right	Rechtsbündige Ausgabe
internal	Füllzeichen
dec	Dezimaldarstellung von Integerzahlen
oct	Oktalдарstellung von Integerzahlen
hex	Hexadezimaldarstellung von Integerzahlen
showbase	Anzeige der Zahlenbasis
showpoint	Anzeige eines Dezimalpunktes für floating-Zahlen
uppercase	Großschreibung des E in floating-Zahlen
showpos	Anzeige eines + für positive Zahlen
scientific	Halblogarithmische Schreibweise von floating-Zahlen
fixed	Festpunktdarstellung von floating-Zahlen
unitbuf	Reaktion des C++-Systems (compilerabhängig)
stdio	meist nicht anwendbar

Die Flags können mit den Methoden **setf** und **unsetf** gesetzt bzw. zurückgesetzt werden.  
Beispiel:

```
#include <iostream>

using namespace std;

main()
{
    // Standardwerte
    cout << 123.23 << " hello " << 100 << '\n';
    cout << 10 << ' ' << -10 << '\n';
    cout << 100.0 << "\n\n";

    cout.setf (ios::hex | ios::scientific);
    cout << 123.23 << " hello " << 100 << '\n';

    cout.setf (ios::showpos);
    cout << 10 << ' ' << -10 << '\n';

    cout.setf (ios::showpoint | ios::fixed);
    cout << 100.00 << '\n';
}
```

Dieses Programm erzeugt die folgende Ausgabe:

```
123.23 hello 100
10 -10
100
```

```
1.232300e+02 hello 100
+10 +4294967286
+100.000
```

### 3 width(), precision(), fill()

In Ergänzung zu den Flags sind in **ios** drei Methoden definiert, die die Feldbreite, das Füllzeichen und die Genauigkeit der Anzeige verändern können:

**width()**

**precision()**

**fill()**

Beispiel:

```
#include <iostream>

using namespace std;

main()
{
    int i;
    cout << " Left Just Right Just \n";
    for (i = 0; i <= 70; i+=10){
        cout.fill('#');
        cout.setf(ios::left);
        cout.width(10);
        cout << i;
        cout.fill ('?');
        cout.unsetf (ios::left);
        cout.width(11);
        cout << i;
        cout << "\n";
    }
}
```

Dieses Programm erzeugt die folgende Ausgabe:

```
0#####?????????0
10#####?????????10
20#####?????????20
30#####?????????30
40#####?????????40
50#####?????????50
60#####?????????60
70#####?????????70
```

## 4 Manipulatoren

Es gibt einen zweiten Weg, um Formatinformationen an des C++ E/A-System zu übergeben: *manipulatoren*. In vielen Situationen ist es einfacher, Manipulatoren zu verwenden als ios-Flags. Um Manipulatoren mit Parameterwerten zu verwenden, muß man die Datei **iomanip** einbinden.

Manipulator	Zweck	Ein-/Ausgabe
dec	Dezimale Ausgabe	Ausgabe
endl	End of Linea und leeren des Streambuffers	Ausgabe
ends	End of String	Ausgabe
flush	Leeren des Streambuffers	Ausgabe
hex	Hexadezimale Ausgabe	Ausgabe
oct	Oktale Ausgabe	Ausgabe
resetiosflags(long f)	Rücksetzen eines Flags f	Ein-/Ausgabe
setbase(int base)	Setzen der Zahlensbasis auf base	Ausgabe
setfill(int ch)	setzen der Füllzeichen	Ausgabe
setiosflags(long f)	Setzen eines Flags f	Ein-/Ausgabe
setprecision(int p)	Setzen der Anzahl von Stellen p für die Genauigkeit der Anzeige	Ausgabe
setw(int w)	Setzen der Feldbreite auf w	Ausgabe
ws	Überspringen von führenden Whitespaces	Eingabe

Beispiel:

```
#include <iostream>
#include <iomanip>

using namespace std;

main()
{
    int i;
    cout << " Left Just Right Just Hexadecimal \n";
    for (i = 0; i <= 70; i+=10){
        cout << setfill('#');
        cout << setiosflags(ios::left) << setw (10) << i;
        cout << setfill ('?');
        cout << resetiosflags (ios::left) << setw (11) << i;
        cout << hex << i <<setw (13) << setfill ('*') << i;
        cout << "\n";
        cout << setfill (' ');
        cout << dec;
    }
}
```

## 5 Grundlagen der Dateiein- und -ausgabe

Um die Datei-Ein-/Ausgabe auszuführen muß die Headerdatei **fstream.h** importiert werden. Diese definiert verschiedene Klassen wie **ifstream**, **ofstream**, **fstream**. Diese Klassen sind von **istream** bzw. **ostream** abgeleitet, welche wiederum von **ios** abgeleitet sind. Man hat also vollen Zugriff auf die Möglichkeiten, die bisher in diesem Kapitel diskutiert wurden.

**Beispiel 1: File erzeugen, Inhalt ausgeben**

```
#include <iostream>
#include <fstream>

using namespace std;

main()
{
    ofstream fout ("test");

    if(!fout)
    {
        cout << "Outputfile nicht zu oeffnen \n";
        return 1;
    }

    fout << "Hello\n";
    fout << 100 << ' ' << hex << 100 << endl;

    fout.close();

    ifstream fin("test");

    if(!fin)
    {
        cout << "Inputfile nicht zu oeffnen\n";
        return 1;
    }

    char str[80];
    int i,j;

    fin >> str >> i >> j;
    cout << str << ' ' << i << ' ' << j << endl;

    fin.close();

    return 0;
```

```
}
```

## Beispiel 2: File kopieren

```
#include <fstream>
#include <libc.h>      // fuer exit()

using namespace std;

void error(char *s)
{
    cerr << s << endl;
    exit(1);
}

main()
{
    ifstream von("files.c"); // Anlegen eines Objektes der
                             // Klasse ifstream
    if(!von)
        error("Fehler beim Oeffnen der Quelle");
    ofstream nach("ziel");   // Anlegen eines Objektes der
                             // Klasse ofstream
    if(!nach)
        error("Fehler beim Oeffnen des Ziels");
    char c;
    while(von.get(c))        // solange ohne Fehler
        nach.put(c);         // zeichenweise kopieren
    if(!von.eof())
        error("Fehler beim Kopieren Quelle ");
    von.close();
    nach.close();
    return(0);
}
```

### Beispiel 3: Wahlfreier Zugriff

```
#include <iostream>
#include <fstream>
#include <cstdlib>

using namespace std;

main(int argc, char * argv[])
{
    if(argc != 4) {
        cout << "Usage : change <filename> <byte> <char> \n";
        return 1;
    }

    fstream out(argv[1], ios::in|ios::out);
    if(!out){
        cout << "Cannot open file \n";
        return 1;
    }

    out.seekp(atoi(argv[2]), ios::beg);

    out.put(*argv[3]);
    out.close();

    return 0;
}
```

Weitere Beispiele finden Sie unter

<http://www.informatik.tu-chemnitz.de/Grundl-Inf/Aufgaben/beispiele/>