

File-Streams

Die Headerdatei `<fstream>` definiert die Objekttypen

```
std::ifstream für die Eingabe aus Dateien  
std::ofstream für die Ausgabe auf Dateien  
std::fstream für beides.
```

Verwendung von File-Streams:

File-Streams müssen ebenfalls zuerst erzeugt werden. Am einfachsten geht das mit einer Initialisierung (ein Konstruktor-Aufruf mit dem Filenamen), der den Filestream auch schon im richtigen Modus öffnet (`ifstream` zum Lesen, `ofstream` zum **Überschreiben**). Falls eine **Ausgabedatei** nicht existiert, wird sie beim Öffnen angelegt (Eingabedateien werden nicht neu angelegt).

Die Eingabe und Ausgabe erfolgt wie gewohnt mit den C++Operatoren `<<` und `>>`. Auch die IO-Manipulatoren lassen sich wie gewohnt verwenden.

```
std::ofstream f{"Myfile.txt"};  
    // öffnet f als eine vorhandene oder neue Datei Myfile.txt  
f << "Der Wert von a ist " << a << '\n'; // schreibt a in die Datei  
  
std::ifstream in{"MyInput"}; // öffnet die existierende Datei MyInput  
double x;  
if (in.is_open())  
    in >> x; // liest den Wert für x aus der Datei
```

Vor dem Einlesen testen wir, ob die Datei überhaupt lesbar ist (das hätte man oben vor dem Schreiben auch tun sollen!).

Zusätzlich zu den Operatoren `<<` und `>>` gibt es noch weitere Objektmethoden, die man für File-Streams verwenden kann:

<code>bool is_open()</code>	Überprüft, ob der Filestream geöffnet ist
<code>bool good()</code>	Überprüft, ob der Stream keinen Fehler gemeldet hat
<code>bool bad()</code>	Überprüft, ob der Stream einen Fehler gemeldet hat
<code>bool eof()</code>	Überprüft, ob der Stream am Dateiende ist
<code>void open(Filename)</code>	Öffnet Filestream
<code>void close()</code>	schließt Filestream

Die Methoden `open()` und `close()` kann man sich in meistens C++ sparen, da man den File-Stream durch die Initialisierung öffnen kann und der File-Stream von seinem Destruktor **vollautomatisch** geschlossen wird.

```
std::ofstream f; // Nicht initialisierter Stream
f.open("Myfile", std::ios::app); // öffne Datei zum „hinten Anhängen“
if (!f.is_open()) // Klappte das Öffnen?
    std::cerr << "Fehler bei open()\n"; // Nein!
else
    f << "Dies ist der Output"; // Ja! Wir können schreiben
```

Man könnte die ersten zwei Zeilen auch durch den Konstruktor ersetzen (ist kürzer!):

```
std::ofstream f{"Myfile", std::ios::app}; //Öffne per „Initialisierung“
```

Das Schließen erfolgt jeweils durch den Destruktor von `f` automagisch.

Die optionale Verwendung von `ios::app` öffnet die Ausgabedatei im Anhängemodus, d.h. der alte Inhalt bleibt erhalten, der neue Output wird hinten angehängt. Ohne dieses Argument werden Ausgabedateien beim Öffnen sofort geleert, d.h. **der alte Inhalt ist verloren!**