

Input / Output in Files

Ein Spezialfall des Schreibens in / Lesens von Files wurde mit den Befehlen `cin` und `cout` behandelt. Es lassen sich allgemeinere Ein- und Ausgabeströme öffnen, die dann ein Schreiben in Dateien erlauben.

- **Syntax**

```
#include iostream
#include fstream
.....
ifstream inFile ( InputFileName );
ofstream outFile ( OutputFileName );
if ( inFile.is_open() ) {
    while ( inFile >> myInteger >> myDouble )
        { myInteger *= 10 ; }
    inFile.clear();
    inFile.seekg(0, ios::beg);
    inFile.close();
}
else {
    cerr << "Can not open file" << InputFileName << endl;
}
```

Einfügen der header files

Öffnen eines Input und Output Streams

War das Öffnen erfolgreich

Lesen bis File Ende

Reset ifstream, z.B. um File erneut zu lesen

Setzen der Fileposition auf Filebeginn

Schliessen des Files

Jetzt könnte das File erneut gelesen werden!

Fehlerbehandlung

Input / Output in Files

Mit den Ein und Ausgabeströmen sind Zustandsbeschreibungen verknüpft, die verwendet werden können.

```
ifstream inFile ( InputFileName );  
... ..  
inFile.bad();  
inFile.good();  
inFile.eof();           true, wenn das File Ende erreicht wurde  
inFile.clear();       Reset der obigen Flags, wird beim erneuten Lesen benötigt  
... ..  
while ( !inFile.eof() ) { Ausführung bis zum Ende }  
.....
```

Die Positionierung im geöffneten File lässt sich steuern

```
inFile.seekg(0, ios::beg);  
inFile.seekg(0, ios::end);
```

Positionierung relativ zum Filebeginn oder zum Fileende. Der 1. Parameter ist der Offset.

Die Dokumentation enthält weitere Funktionen.

Input / Output in Files

Mit Hilfe von Funktionen können Text Files gelesen werden.

```
#include <iostream>
#include <fstream>
#include <string>
.....
string TextZeile;

ifstream inFile ("BeispielTextFile.txt");
if (inFile.is_open()) {
    while ( getline ( inFile , TextZeile ) ) {
        cout << TextZeile << endl;
    }
    inFile.close();
}
else {
    Cerr <<"Fehler, File kann nicht geoeffnet werden " << endl;
}
```

Input / Output in Files

Das folgende Beispiel zeigt noch einmal die wichtigsten C++ Anweisungen, die für den lesenden und schreibenden Zugriff auf Files gebraucht werden.

```
#include <iostream>
#include <fstream>
int main {
```

```
...
```

```
char Filename[] = "myFile.txt";
ofstream outFile ( Filename );
outFile << myVarA << myVarB ;
```

```
ifstream inFile ( Filename );
inFile >> myVarA >> myVarB ;
```

```
...
```

```
outFile.close();
inFile.close();
```

```
}
```

Definitionen für die
Input/Output Klassen

Output File Name
Pointer für Output Stream
Schreiben von 2 Variablen

Schliessen des Output Stream

readFile.cc

Arbeitsvorschlag:

- Schreiben Sie ein Programm, das in ein File schreibt bzw. liest.