

Beispiele für Dateioperationen unter C++

Für Datenströme von und zur Festplatte verwenden wir Objekte aus der <fstream>-Bibliothek, die wir (zusätzlich zur <iostream>-Bibliothek) in unser C++-Programm einbinden.

Für lesende Dateioperationen verwenden wir das ifstream-Objekt (Input-File-Stream), das folgende Methoden kennt:

.open(Dateiname)	öffnet eine Datei mit dem Namen „Dateiname“ zum Lesen.
.get(Zeichen)	liest das nächste Zeichen aus der Datei in die Variable „Zeichen“. Zu Beginn steht der Dateizeiger auf 0, es wird das erste Zeichen eingelesen. Der Dateizeiger wird dabei um eins erhöht, beim nächsten get wird also das nächste Zeichen gelesen. Ist nichts mehr zu lesen da (End Of File, EOF), gibt get „false“ zurück.
.close()	schließt die Datei wieder und gibt sie dem Betriebssystem als „nicht mehr verwendet“ frei.

```
// DateiLesen.cpp : Liest aus einer Datei
//
#include <iostream>.....// Allgemein fuer Streams
#include <fstream>          // Zusaetzlich fuer Dateioperationen

using namespace std;

int main()
{
    // Eingabedatei definieren
    char dateiname[30] = "test.txt";
    ifstream datei;
    datei.open(dateiname);      // Datei öffnen

    // Prüfen, ob Datei korrekt geöffnet wurde
    if (datei) {
        // Eingabedatei zeichenweise lesen und am Bildschirm ausgeben
        char c;
        while (datei.get(c)) {
            cout << c;
        }
        cout << endl;

        // Datei schliessen
        datei.close();
    }
    else {          // Datei nicht korrekt geöffnet
        cerr << dateiname << " kann nicht geoeffnet werden!" << endl;
    }

    system("pause");
    return 0;
}
```

Für schreibende Dateioperationen verwenden wir das ofstream-Objekt (Output File Stream), das ebenfalls die Methoden .open(„Dateiname“) und .close() kennt.

Übrigens können wir beim Erzeugen unseres ifstream- oder ofstream-Objekts auch gleich den betreffenden Dateinamen mit angeben und somit gleich initialisieren. Wir sparen uns dann auch eine Zeile C++-Code mit dem expliziten Öffnen der Datei...

Zum Schreiben „schieben“ wir die Dateien statt auf die Konsole (Bildschirm, cout) einfach in unsere Datei: statt cout << zuSchreibendeDaten also einfach: datei << zuSchreibendeDaten

```
// DateiSchreiben.cpp :
// Ausgabe des Programms wird in eine Datei geschrieben.
//
#include <iostream>.....// Allgemein fuer Streams
#include <fstream>          // Zusaetzlich fuer Dateioperationen

using namespace std;

int main()
{
    // Ausgabedatei definieren und oeffnen
    ofstream datei("test.txt");

    if (!datei) {
        cerr << "test.txt" << " kann nicht geoeffnet werden." << endl;
        exit(-1);
    }

    cout << "Die Datei " << "test.txt" << " wurde geoeffnet..." << endl;

    string text = "Neuer Inhalt";
    datei << text << endl;
    datei << "Jetzt schreibe ich direkt in die Datei." << endl;
    datei << "Jetzt schreibe ich nochmal direkt in die Datei." << endl;
    datei << "Danach schreibe ich nichts mehr in die Datei." << endl;

    datei.close();

    cout << "Die Datei wurde wieder geschlossen." << endl;
    system("pause");
    return 0;
}
```

Beim Öffnen einer Datei können wir weitere Parameter übergeben – hier bieten sich vordefinierte Konstanten der Klasse `ios` an. Im folgenden Beispiel wird eine Datei nicht überschrieben (Standard), sondern werden die Daten angehängt („append“) und im Binärmodus gespeichert (damit sind auch Sonderzeichen möglich, die sonst als Steuerzeichen / Escape-Sequenzen interpretiert würden):

```
ofstream datei(„text.txt“, ios::app | ios::binary);
```