

Dateiverarbeitung - Textdateien

- Problem: große Datenmengen eingeben
- Lösung: Daten aus einer Textdatei einlesen statt über Tastatur einzugeben
- Bisherige Ein-/Ausgabe über *cin* und *cout* ist nur der Spezialfall einer Textdatei
- Verallgemeinerung ist leicht möglich
- Alles über *cin* und *cout* Gelernte funktioniert weiterhin!!

Dateiverarbeitung - Textdateien

- Zusätzlich erforderlich:
 - Es muss eine „Verbindung“ zwischen der Datei auf der Festplatte und dem Programm hergestellt werden
- Danach kann aus der Datei (= wie bisher von Tastatur) gelesen
- bzw. in die Datei (= auf den Bildschirm) geschrieben werden
- D.h. Umlenkung der Ein- und Ausgabe

Öffnen der Datei

- Benötigt wird ein Objekt der Klasse *fstream*:
`fstream datei;`
- Beim Öffnen wird dieses Objekt (oder Variable) fest mit einer Datei auf der Festplatte verknüpft:
`datei.open („dateiname.ext“, ios::out);`
- Diese Anweisung öffnet die Datei *dateiname.ext* zum Schreiben
- Gibt es die Datei noch nicht, wird sie automatisch angelegt.
- Existiert die Datei bereits, wird ihr Inhalt ohne Rückfrage gelöscht bzw. überschrieben.
- Die Datei liegt im selben Verzeichnis wie die zugehörige cpp-Datei.

Dirk Seeber , Informatik , Teil 7

3

Schreiben in die Datei

- Die Ausgabe unterscheidet sich nicht vom Schreiben/Ausgabe auf den Bildschirm:
`string s = "Ein String";
int n = 46;
datei << s; // ähnlich zu cout
datei << endl;
datei << n;`
- Nacheinander werden der String, ein Zeilenumbruch und die Zahl 46 in die Datei geschrieben.
- D.h. insgesamt zwei Zeilen

Dirk Seeber , Informatik , Teil 7

4

Schließen der Datei

- Nach dem Schreiben sollte die Datei geschlossen werden.
- Dadurch wird sichergestellt, dass auch wirklich alle Daten auf der Festplatte landen und eventuelle Puffer geleert werden:
`datei.close();`
- Spätestens beim Beenden des Programms wird die Datei automatisch geschlossen

Lesen aus einer Datei

- Soll aus einer **vorhandenen** Datei nur gelesen werden, muss diese so geöffnet werden:
`datei.open („dateiname.ext“, ios::in);`
- Gelesen wird wie von der Tastatureingabe gewohnt:
`datei >> s;`
`datei >> n;`
- Stehen in einer Zeile der Datei mehrere Werte, müssend diese mit einer Leseanweisung abgeholt werden!
`datei >> i, j ,k, n;`

Lesen aus einer Datei

- Die angegebenen Variablen zur Aufnahme der Werte müssen zum Wert passen.
- Die Umwandlung von kleineren in den größeren Datentyp (z. B. `int` nach `double`) funktioniert auch hier, aber nicht umgekehrt.
- Leerzeichen, Tabulatoren, Zeilenumbruch und ähnliche Sonderzeichen trennen die Werte voneinander und können so nicht eingelesen werden!
(Genau wie bei der Eingabe über Tastatur)

Dirk Seeber , Informatik , Teil 7

7

Lesen aus einer Datei

- Es sollte nicht über das Ende der Datei gelesen werden!
- Die Funktion `datei.eof()` liefert `true`, wenn das Ende der Datei erreicht wurde.
- D.h. die Daten werden in einer Schleife eingelesen, solange Dateiende nicht erreicht ist
`while (!datei.eof())`
oder ausführlicher
`while (false == datei.eof())`

Dirk Seeber , Informatik , Teil 7

8

Anfügen an eine Datei

- Eine Datei kann so geöffnet werden, dass der Inhalt erhalten bleibt und neue Daten am Ende angefügt werden können:

```
datei.open („dateiname.ext“, ios::app);
```

- Gelesen wird immer vom Anfang der Datei.
- Geschrieben wird immer an das aktuelle Ende der Datei noch.
- Auch hier Schließen nicht vergessen, um die Puffer des Betriebssystems zu leeren.

Zusatzinformation bzgl. Öffnen einer Datei

- Besonderheit beim Deklarieren der Ein- bzw. Ausgabedatei

```
ifstream eingabeDatei;
```

- Die Datei kann **nur** zum Lesen geöffnet werden:

```
eingabeDatei.open („dateiname.ext“, ios::in);
```

```
ofstream ausgabeDatei;
```

- Die Datei kann **nur** zum Schreiben geöffnet werden:

```
ausgabeDatei.open („dateiname.ext“, ios::out);
```

bzw.

```
ausgabeDatei.open („dateiname.ext“, ios::app);
```

Häufige Probleme

- **Die Datei kann nicht geöffnet werden.**
Das kann viele Gründe haben, z.B. die Datei wird nicht gefunden:
 - meistens ist es ein Fehler im Dateinamen inklusive der Dateierweiterung (wie. Z.B. txt, ttt, ...)
 - Die Datei liegt nicht im richtigen Verzeichnis
- Es gibt keine Fehlermeldung, daher muss der Entwickler selbst Vorsorge treffen:

```
if ( ! datei.good() )  
    cout << "Fehler beim Oeffnen" << endl;
```

unmittelbar nach dem "Öffnungsversuch" hilft dabei.

Dirk Seeber , Informatik , Teil 7

11

Häufige Probleme

- **Die Datentypen in der Datei passen nicht zu den Datentypen im Programm.**
- Hier hilft nur **exaktes** Arbeiten,
der Debugger ist hierbei eine große Hilfe
- Alternativ können alle Werte als String eingelesen und selbst ins geforderte Format umgewandelt werden

```
// Zeile einlesen  
getline( eingabeDatei, zeile, '\n' );  
hausliste[anzahl].preis = atof(zeile.c_str());
```

Dirk Seeber , Informatik , Teil 7

12

Häufige Probleme

- **Beim Schreiben und Lesen werden die Zeilenumbrüche unterschiedlich eingesetzt.**
- **Die Reihenfolge der Werte unterscheidet sich zwischen Schreiben in die Datei und Lesen aus der Datei**
 - Werte werden falsch zugeordnet
 - Werte werden unterschlagen
- Auch hier hilft nur **exaktes** Arbeiten,