


C++ Übungsblatt 5 – fstream

Aufgabe 5-1: Kontoausgänge

CSV ist eine einfache Textdatei, die strukturierte Daten in Form einer Tabelle beinhaltet. Dabei werden in jeder Zeile (= Tabellenreihe) die einzelnen Daten (= Zelle) durch ein Satzzeichen wie Komma oder Semikolon getrennt.

Das folgende Beispiel besteht aus 4 Spalten und 3 Zeilen, wobei die oberste Zeile einer Kopfzeile entspricht.

Datum;Uhrzeit;Ort;Temperatur				
01.10.2019;10:00;Kempten;23,2				
01.10.2019;11:00;Kempten;24,6				



	A	B	C	D
1	Datum	Uhrzeit	Ort	Temperatur
2	01.10.2019	10:00	Kempten	23,2
3	01.10.2019	11:00	Kempten	24,6
4				

Ihre Aufgabe ist es nun, ein Bankkonto zu implementieren, das sämtlichen Zahlungsverkehr in einer csv-Datei lesen und schreiben kann. Dies soll wie folgt ablaufen:

1. Beim Start wird überprüft, ob eine Datei namens „zahlungsverkehr.csv“ existiert
 - a. Existiert die Datei, wird diese eingelesen und in handhabbare Objekte gespeichert
 - b. Existiert die Datei nicht, wird ein neues Konto angelegt
2. Der Benutzer kann nun neue Überweisungen eintragen (nicht bearbeiten oder löschen). Dem Benutzer wird immer der aktuelle Kontostand angezeigt – dieser darf hierbei niemals ins Negative rutschen
3. Beim Beenden des Programms werden die Daten wieder in „zahlungsverkehr.csv“ gespeichert, sodass diese beim nächsten Start wieder eingelesen werden können

Für die Überweisungen sollen **Datum**, **Uhrzeit**, **Empfänger**, **Verwendungszweck** und **Betrag** gespeichert werden (ggf. mit Kontostand nach Erfolgen der Überweisung). Datum und Uhrzeit soll zum Zeitpunkt des Eintrags automatisch ermittelt werden (siehe `ctime`).

Wenn Sie anschließend die csv-Datei haben, können Sie versuchen, diese in Excel zu öffnen und beispielsweise mit Pivot-Tabellen zu analysieren.

Hinweis zu Excel: Excel kann sich unter Umständen schwertun, Trennzeichen richtig zu erkennen. Im Deutschen werden Zellen standardmäßig durch Semikolons und Dezimalzahlen durch ein Komma getrennt, wohingegen im Englischen Zellen durch Kommas und Dezimalzahlen durch einen Punkt getrennt sind. Zur Not ändern Sie die csv-Endung der Datei zu txt, so führt Excel beim Öffnen der Datei den Text-Assistenten aus.

Aufgabe 5-2: Binäre Dateien

Textdateien, die sich in jedem üblichen Editor öffnen lassen, sind einfach zu Handhaben. Wichtig zu berücksichtigen ist jedoch, dass Daten wie Kommazahlen und Uhrzeiten in ASCII Zeichen umgewandelt werden, bevor sie geschrieben werden (bspw. wird die 8 Bit Zahl 01001101 in die zwei Ziffern 77 umgewandelt, repräsentiert in Binär durch 00110111 00110111).

In manchen Fällen bietet es sich an, Binärdateien zu schreiben. Sie benötigen weniger Speicherplatz, können die Leistung verbessern und verringern in manchen Fällen die Komplexität.

Ändern Sie Ihre Lösung aus 5-1 so ab, dass Ihr Programm primär mit einer Binärdatei hantiert!

Folgende Punkte sind zu beachten:

- Der Dateiname sollte nicht mit csv enden. Eine Alternative für rohe Binärdateien wäre .bin oder .dat
- Tabellenüberschriften (Datum, Uhrzeit, ...) können Verworfen werden
- Datum und Uhrzeit wird von `ctime` als eine Zahl gehandhabt, so müssen diese nicht in zwei Spalten aufgeteilt werden
- Da Zahlen eine feste Breite haben, müssen die Spalten nicht durch ein Semikolon getrennt werden, somit können diese ebenfalls verworfen werden. Bei Text ist das Ende automatisch durch das Nullterminierungszeichen gegeben (`'\0'`)
- Der `<<` Operator ist nicht dazu gedacht, um Daten binär zu schreiben und zu lesen. Verwenden Sie stattdessen die Methoden `write` und `read`
- Es soll immer noch die Option geben, eine csv-Datei zu generieren

Vergleichen Sie die Dateigröße der resultierenden Binärdatei mit der des CSVs.

Sie können Programme wie Microhex oder Hex Editor Neo verwenden, um die Daten der Binärdatei in Hex-Format anzeigen zu lassen. Auf UNIX-Systemen (Mac, Linux) gibt es die Möglichkeit, diese in Terminal mittels `hexdump` oder `xxd` anzeigen zu lassen (`xxd dateiname.bin`).