Einleitendes Beispiel

Schreiben Sie ein Programm *CntWords*, dem eine Datei übergeben wird, und das alle Worte (die mit einem Buchstaben beginnen und länger als ein Zeichen sind) zählt (d.h. wie oft jedes Wort vorkommt).

Ausgegeben werden sollen alle Worte, die mindestens 3mal vorkommen und zwar sortiert in absteigender Reihenfolge des ASCII-Codes.

```
          zur :    4
           zu :    4
         wird :    4
       werden :    4
          von :    4
          und :    8
         oder :    3
          nur :    3
        nicht :    3
          ist :    5
           in :    4
         fuer :    3
        durch :    3
          die :    8
          der :   12
         dass :    5
          das :    3
          auf :    4
       Klasse :    5
  Hilfsklasse :    3
          C++ :    4
```

Programmsegment zum Datei-Öffnen und Zählen

```
int main() {
    map<string, int, greater<string> > wordCntMap;
    ifstream datei(".\\testFileCpp.txt", ios::in);
    // Von Anfang bis Dateiende
    while (datei.good())    {
        string word;
        // Das nächste Wort aus der Datei einlesen.
        datei >> word;
        // Wort fuer Wort aus der Zeile extrahieren
        if ((word.size() > 1) && isalpha(word[0])) {
            ++wordCntMap[word];
        }
    } // while (datei. good()
```

(*) Wenn nicht vorhanden, wird int mit 0 initialisiert

Programmsegment zur Ausgabe

```
// Ausgabe der Worte alphabetisch absteigend
for (auto& iter : wordCntMap) {
    if (iter.second > 2) {
        cout << setw(15) << iter.first << " : "
             << setw(5) << iter.second << endl;
    }
}
```

Programmkopf

```
#include <iostream>
#include <fstream>
#include <cctype>  // wegen isalpha
#include <string>
#include <map>
#include <functional> //wg. greater
#include <iomanip> // wegen setw
using namespace std;
```

Datei-Öffnen und Zählen mit Fehlerprüfung

```
int mainComplex() {
    typedef map<string, int, greater<string> > T_WordCntMap;
    T_WordCntMap wordCntMap;
    // Datei zum Einlesen oeffnen und abbrechen bei Fehler
    ifstream datei(".\\testFileCpp.txt", ios::in);
    if (!datei.good())  {
       cerr << "datei konnte nicht geoeffnet werden." << endl;
       exit(-1);
    }
    // Von Anfang bis Dateiende
    while (datei.good()) {
        string word;
        // Das nächste Wort aus der Datei einlesen.
        datei >> word;
        // Wort fuer Wort aus der Zeile extrahieren
        if ((word.size() > 1) && isalpha(word[0])) {
           ++wordCntMap[word];
        }
    } // while (datei. good()
```

Übung zu Maps

1. Ändern Sie das Programm, sodass die Worte in absteigender Reihenfolge ausgegeben werden.

2. Ändern Sie die Worte, sodass nicht mehr zwischen Groß- und Kleinschreibung unterschieden wird. „DerDicke“ und „derdicke“ sind das gleiche Wort, aber „Der Dicke“ sind zwei Worte.

3. Geben Sie nur das Wort, das am häufigsten vorkommt, aus.

Übung zu Maps / Sets

Schreiben Sie ein weiteres Programm *CntKeyWords*, das alle Schlüsselworte von C++ (bzw. eine Auswahl davon) in einer übergebenen Datei zählt. Gehen Sie davon aus, dass alle Schlüsselworte durch Blanks voneinander getrennt sind (stimmt zwar nicht, aber wir machen es uns einfacher).

Die Schlüsselworte werden zum effizienten Zugriff in einer Menge abgelegt.

Lösungsidee Übung *CntKeyWords (Ausschnitt)*

```
#include <iostream>
#include <fstream>
#include <string>
#include <set>
#include <map>
using namespace std;

typedef
map<string, int>       WordCntMap;

typedef set<string>  KeyWords;

void FillKeyWordTable
        ( KeyWords& kwTab) {
   kwTab.insert("void");
   kwTab.insert("int");
   . . .
}
```

```
int main() {
   WordCntMap wcMap;
   KeyWords kwTab;
   ifstream datei("xxx.cpp");

   string wort;
   FillKeyWordTable(kwTab);

   //  Alle Vorkommen auf 0 setzen
   KeyWords::const_iterator  siter;
   for (siter = kwTab.begin();
         siter != kwTab.end(); ++siter) {
      wcMap[*siter] = 0;
   }
```

Lösungidee CntKeyWords (2)

```
   datei >> wort;
   while (datei.good()) {
      if (kwTab.find(wort) != kwTab.end()) {
         ++wcMap[wort];
      }
      datei >> wort;
   }

   WordCntMap::const_iterator iter;
   for (iter = wcMap.begin(); iter != wcMap.end(); ++iter) {
      cout << (*iter).first << " : " << (*iter).second << endl;
   }
}
```

Erweiterung der Übung zu Maps/Sets

Schreiben Sie ein weiteres Programm, das alle Schlüsselworte von C++ (bzw. eine Auswahl davon) in einer übergebenen Datei zählt.

Schlüsselworte sind **nicht** mehr unbedingt durch ein Blank vom Rest abgetrennt.

Die Schlüsselworte werden zum effizienten Zugriff in einer Menge (set<string>) abgelegt.
Diese kann man entweder als Konstante zu Beginn im Programm in die Menge einfügen oder auch aus einer Datei einlesen.