

## Freiwillige Offline-Aufgabe 08-03 (INF & WI & MCD): Wortliste eines Texts, mit Häufigkeiten (geübte C++ Konstrukte: *string*, *Array*, *Schleifen*, *struct*)

Schreiben Sie ein C++ Programm, welches einen Text in Form von mehreren Eingabezeilen einliest. Das Programm soll die in dem Text auftretenden Worte in der Reihenfolge ihres Auftretens mitsamt der Häufigkeit ihres Vorkommens im Text ausgeben.

Einzelne Eingabezeilen und Worte sollen als Strings gespeichert werden. Die „Sammlung aller nicht-leeren Eingabezeilen“ sowie die „Sammlung der Worthäufigkeiten“ sollen als statische Arrays gespeichert werden. Es kommen maximal 5 Eingabezeilen und maximal 1000 Worte vor.

Eine leere Eingabezeile beende die Eingabe. Eingabezeilen können auch Leerzeichen enthalten. Leerzeichen, Punkt, Komma sowie Anfang und Ende jeder Eingabezeile fungieren als Wortbegrenzungen. Im Eingabetext sollen keine anderen Satzzeichen als Punkt und Komma (und Leerzeichen) vorkommen. Ihr Programm kann davon ausgehen, dass der Benutzer nur korrekte Eingaben macht.

Die Worthäufigkeit jedes Worts soll in einer Datenstruktur ...

```
struct w_haeufigkeit {  
    string wort;  
    int haeufigkeit;  
};
```

... gespeichert werden (und davon ein Array mit 1000 Einträgen).

Es sollen für das Ermitteln der Worthäufigkeiten folgende Funktionen definiert werden (diese Funktionen werden von Jenkins ggfs. auch einzeln getestet, sie müssen also vorhanden sein und wie vorgeschrieben definiert werden):

```
string naechstes_wort(string zeile, int& pos);
```

... suche ausgehend von der Position `pos` in der Zeile das nächste Wort und gebe es zurück. `pos` wird dann auf die erste Position (in der Zeile) hinter dem Wort geändert. Sollte es ausgehend von `pos` kein nächstes Wort in der Zeile geben, so werde ein leerer String zurückgegeben.

```
void zaehle_wort(string wort,
                w_haeufigkeit haeufigkeiten[],
                int& w_count);
```

... suche das Wort `wort` in der Wortsammlung `haeufigkeiten[]`, die aktuell bis zur Position `w_count-1` belegt sei. Kommt das Wort schon vor, so werde seine Häufigkeit um 1 erhöht. Kommt das Wort bisher noch nicht vor, so werde es hinzugefügt und seine Häufigkeit auf 1 gesetzt.

Die Arrays für den Eingabetext und die gesammelten Worthäufigkeiten seien lokale Variablen des Hauptprogramms. Benutzen Sie ggfs. folgendes Programmgerüst:

```
#include <iostream>
#include <string>
using namespace std;

struct w_haeufigkeit {
    string wort;
    int haeufigkeit;
};

string naechstes_wort(string zeile, int& pos)
{
}

void zaehle_wort(string wort, w_haeufigkeit haeufigkeiten[], int& w_count)
{
}

int main()
{
    int z_count = 0, w_count = 0;
    string wort = "";
    string eingabe[5];

    /* Eingabe hier, setzt auch z_count auf die Anzahl nicht-leerer Zeilen */ {

    w_haeufigkeit haeufigkeiten[1000];

    for (int k = 0; k < z_count; k++)
    {
        int pos = 0;
        while (true)
        {
            wort = naechstes_wort(eingabe[k], pos);
            if (wort == "")
                break;
            zaehle_wort(wort, haeufigkeiten, w_count);
        }
    }
}
```

```
    for (int k = 0; k < w_count; k++)
        cout << haeufigkeiten[k].wort << ": "
            << haeufigkeiten[k].haeufigkeit << endl;

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

### Testläufe (Benutzereingaben sind unterstrichen):

---

```
Eingabezeile = ? Dies ist die erste Zeile des Texts.
Eingabezeile = ? Und dies, ist, noch eine Zeile.
Eingabezeile = ? .Und noch eine weitere Zeile, im Text.
Eingabezeile = ?
Dies: 1
ist: 2
die: 1
erste: 1
Zeile: 3
des: 1
Texts: 1
Und: 2
dies: 1
noch: 2
eine: 2
weitere: 1
im: 1
Text: 1
Drücken Sie eine beliebige Taste . . .
```

---