

## 4 Schleifen und Dateien

### 4.1 Übungsaufgabe

#### 4.1.1 Aufgabe 1

Schreiben Sie drei C++ Programme: `pword_for.cxx`, `pword_while.cxx`, `pword_do.cxx`. Die Programme sollen den Benutzer höchstens 5 Mal nach einem Passwort fragen. Die interne Abfrageschleife soll bei den unterschiedlichen Programmen als `while`-, `for`- oder `do`-Schleife implementiert werden.

Achtung: Das interne Passwort soll aus der Datei `mypasswd.txt` gelesen werden. Verwenden Sie als Routinen des File I/O aus Kapitel 3.7.4 des Vorlesungsskriptes.

Die Eingaben des Benutzers, auch die Eingaben aus den letzten Programmläufen, sollen in einer Datei `log.txt` gespeichert werden. Nutzen Sie hierzu `open` `for` `append`.

#### 4.1.2 Aufgabe2

Schreiben Sie ein Programm `count_punct`. Dieses Programm soll eine Textzeile einlesen, diese ausgeben und in der nächsten Zeile die Positionen aller Piktuationszeichen anzeigen.

### 4.2 Wiederholung und zusätzliche Informationen

#### 4.2.1 Arbeit mit Dateien

Zum Lesen aus Dateien verwendet man den Stream `ifstream`, zum Schreiben in Dateien den Stream `ofstream`.

Öffnen der Datei `sz.txt` zum Lesen:

```
include <fstream>
ifstream eingabe ("sz.txt");

oder:

include <fstream>
ifstream sz;
sz.open("sz.txt");
```

oder:

```
include <fstream>
ifstream sz;
string filename = "sz.txt";
sz.open(filename.c_str());
```

Öffnen der Datei `sz.txt` zum Schreiben:

```
include <fstream>
ofstream sz("sz.txt");
```

oder:

```
include <fstream>
ofstream sz;
sz.open("sz.txt");
```

oder:

```
include <fstream>
ofstream sz;
string filename = "sz.txt";
sz.open(filename.c_str());
```

Öffnen der Datei `sz.txt` zum Anhängen: Genau wie Öffnen der Datei zum Schreiben, aber mit dem Zusatz `ios::app` im `open`-Befehl.

```
sz.open("sz.txt", ios::app);
```

#### 4.2.2 Schleifen

Schleifen werden verwendet, wenn man eine Aktion mehrfach ausführen möchte. Die 3 wichtigsten Schleifen sind `for`, `while` und `do`.

Die `for`-Schleife wird verwendet, wenn der Index im Vordergrund steht. Oft hängen Iteration und Abbruchkriterium eng zusammen. Ein Beispiel für den Einsatz von `for`-Schleifen ist das Durchlaufen von Arrays.

Die `while`-Schleife wird verwendet, wenn das Abbruchkriterium im Vordergrund steht. Im Gegensatz zur `do`-Schleife wird bei der `while`-Schleife zuerst das Abbruchkriterium getestet und danach werden die Statements ausgeführt. Ein Beispiel für den Einsatz von `while`-Schleifen ist das zeilenweise Einlesen von Dateien.

Die `do`-Schleife wird verwendet, wenn die Statements im `do`-Block im Vordergrund stehen. Im Gegensatz zur `while`-Schleife werden bei der `do`-Schleife zuerst die Statements ausgeführt, und danach wird die Abbruchbedingung getestet. Ein Beispiel für den Einsatz von `do`-Schleifen ist die Prüfung eines eingegebenen Passworts mit mehreren Versuchen.

## 4.3 Lösungsvorschlag

### 4.3.1 Aufgabe 1

Zum Benutzen der jeweiligen Schleifen können die anderen Schleifen auskommentiert werden.

```
/*Autor: Nicola Greth
 * Uebung 3, Aufgabe 1
 * Programm: Passwort-Abfrage mit for, while und do-while Schleifen
 */
#include <iostream>
#include <string>
#include <fstream>
using namespace std;

int main() {
    string pw;
    string versuch;
    ifstream DATEI("mypasswd.txt");
    ofstream LOG;
    LOG.open("log.txt", ios::app); //ios::app zum Anhängen
    int i = 0;

    //falls die Datei nicht existiert bricht das Programm ab
    if (!DATEI) {
        cerr << "Inputfile existiert nicht!!!" << endl;
        return (-1);
    }

    //Das Passwort wird aus der Datei gelesen
    getline(DATEI, pw);
    DATEI.close();
    LOG << endl << "Start des Programms" << endl;
    cout << "Geben Sie das Password ein:" << endl;

    //Do-While-Schleife
    do {
        i++;
        cout << i << ". Versuch>>";

        //Die Eingabe wird ausgelesen
        getline(cin, versuch);

        //Die Eingabe wird in die Log-Datei geschrieben
        LOG << versuch << endl;

        //Wenn das Passwort nicht korrekt ist
        if (versuch != pw and i < 5) {
            cout << "Das Passwort ist inkorrekt!" << endl;
            cout << "Sie haben noch " << 5 - i << " Versuche! \n" << endl;
        }
    } while (versuch != pw);
}
```

```
}

//Wenn das Passwort korrekt ist
else if (versuch == pw) {
    cout << "Das Passwort ist korrekt. Sie erhalten Zugang" << endl;
    i = 5; //damit die Schleife abgebrochen wird
}

//Wenn das Passwort beim fünften Versuch nicht korrekt ist
//wird eine Fehlermeldung ausgegeben
else {
    cerr << "Kein Zugriff!!! :P" << endl;
}
} while (i < 5);

//For-Schleife
for(i=1;i<=5;i++) {
    cout << i << ". Versuch>>";

    //Die Eingabe wird ausgelesen
    getline(cin, versuch);

    //Die Eingabe wird in die Log-Datei geschrieben
    LOG << versuch << endl;

    //Wenn das Passwort nicht korrekt ist
    if (versuch != pw and i < 5) {
        cout << "Das Passwort ist inkorrekt!" << endl;
        cout << "Sie haben noch " << 5 - i << " Versuche! \n" << endl;
    }

    //Wenn das Passwort korrekt ist
    else if (versuch == pw) {
        cout << "Das Passwort ist korrekt. Sie erhalten Zugang" << endl;
        i = 5; //damit die Schleife abgebrochen wird
    }

    //Wenn das Passwort beim fünften Versuch nicht korrekt ist
    //wird eine Fehlermeldung ausgegeben
    else {
        cerr << "Kein Zugriff!!! :P" << endl;
    }
}

//While-Schleife
while(i<5) {
    i++;
    cout << i << ". Versuch>>";

    //Die Eingabe wird ausgelesen
    getline(cin, versuch);
```

```

//Die Eingabe wird in die Log-Datei geschrieben
LOG << versuch << endl;

//Wenn das Passwort nicht korrekt ist
if (versuch != pw and i < 5) {
    cout << "Das Passwort ist inkorrekt!" << endl;
    cout << "Sie haben noch " << 5 - i << " Versuche! \n" << endl;
}

//Wenn das Passwort korrekt ist
else if (versuch == pw) {
    cout << "Das Passwort ist korrekt. Sie erhalten Zugang" << endl;
    i = 5; //damit die Schleife abgebrochen wird
}

//Wenn das Passwort beim fünften Versuch nicht korrekt ist
//wird eine Fehlermeldung ausgegeben
else {
    cerr << "Kein Zugriff!!! :P" << endl;
}
}
LOG.close();
return 0;
}

```

### 4.3.2 Aufgabe 2

```

/*Autor: Nicola Greth
 * Uebung 3, Aufgabe 2
 * Programm: Entfernt Piktuationszeichen
 */
#include <iostream>

int main(){
    std::string zeile;
    std::cout<<"Geben Sie eine Textzeile ein>>";
    getline(std::cin,zeile);

    //Gibt die unbearbeitete Zeile aus
    std::cout<<zeile<<std::endl;
    int count_punct=0;

    for(int i=0;i<zeile.size();i++){

        //Wenn es ein Piktuationszeichen ist soll die Position markiert werden
        if(ispunct(zeile.at(i))) {
            std::cout<<"^";
            count_punct++;
        }
    }
}

```

```
        //Wenn es kein Punctuationszeichen ist soll es eine Stelle weiter rücken
        else {
            std::cout<<" ";
        }
    }

    std::cout<<std::endl;

    //Wenn es gar kein Punctuationszeichen gab
    if(!count_punct) {
        std::cout<<"Kein Punctuationszeichen gefunden!"<<std::endl;
    }

    return 0;
}
```