
```

*----- FILEHANDLER
header -----*
#ifndef FILEHANDLER_H
#define FILEHANDLER_H
#include <vector>

using namespace std;

class FileHandler {
public:
    FileHandler();
    void writetoFile(const char* name);
    void readfromFile(const char* name);
    void search(int a);
    void readandwritetoFile(const char* zaehlungen);
    void writetoPrsort(const char* name);
    void bubbleSort(const char* name);
    void writebubblesort(const char* name);
private:
    int randomNumber = 0;
    vector<int> dataList;
    vector<int> sortList;
    vector<int>::const_iterator i;

};

#endif /* FILEHANDLER_H */

*-----FILEHandler CPP-----*
#include "FileHandler.h"
#include <cstdlib>
#include <iostream>
#include <vector>
#include <ctime>

using namespace std;

FileHandler::FileHandler() {
}

void FileHandler::writetoFile(const char* name) {
    srand(time(NULL));
    int rand();
    FILE *pFile = NULL;
    pFile = fopen(name, "w");
    if(pFile != NULL){
        for(int i = 0; i != 10; i++){
            randomNumber = rand() % 10 ;
            fprintf(pFile,"%d,", randomNumber);
        }
        fclose(pFile);
    }
}

```

```

    }
}

void FileHandler::readfromFile(const char* name) {
    FILE *exampleFile = NULL;
    exampleFile = fopen(name, "r");
    int u = 0;
    int a = 0;
    if(exampleFile != NULL) { // Abfangen, dass die File vorher
        existiert.
        while (!feof(exampleFile) && u <= 9) { // Bis Ende der Datei
            lesen.
            fscanf(exampleFile, "%d, ", &a); // Sucht nach ";" Zeichen
            und speichert die Eintraege in eine Array ab.
            dataList.push_back(a);
            printf("%d: %d \n", u, dataList[u]); // Ausgabe
            auf Konsole zur Kontrolle
            u++;
        }
        fclose(exampleFile); // Datei schliessen.
    }
}

void FileHandler::search(int a) {
    int s = 0;
    for(i = dataList.begin(); i != dataList.end(); i++){
        if(*i == a){
            s++;
        }
    }
    printf("Die Zahl %d kommt in der Datei genau %d mal vor \n", a ,
    s);
}

void FileHandler::readandwritetoFile(const char* zaehlungen) {
    FILE *pFile = NULL;
    int null = 0;
    int eins = 0;
    int zwei = 0;
    int drei = 0;
    int vier = 0;
    int fuenf = 0;
    int sechs = 0;
    int sieben = 0;
    int acht = 0;
    int neun = 0;
    pFile = fopen(zaehlungen, "w");
    if(pFile != NULL){
        for(i = dataList.begin(); i != dataList.end(); i++){
            if(*i == 0){
                null++;
            }
            if(*i == 1){

```

```

        eins++;
    }
    if(*i == 2){
        zwei++;
    }
    if(*i == 3){
        drei++;
    }
    if(*i == 4){
        vier++;
    }
    if(*i == 5){
        fuenf++;
    }
    if(*i == 6){
        sechs++;
    }
    if(*i == 7){
        sieben++;
    }
    if(*i == 8){
        acht++;
    }
    if(*i == 9){
        neun++;
    }
}

fprintf(pFile, " %d mal 0 \n", null);
fprintf(pFile, " %d mal 1 \n", eins);
fprintf(pFile, " %d mal 2 \n", zwei);
fprintf(pFile, " %d mal 3 \n", drei);
fprintf(pFile, " %d mal 4 \n", vier);
fprintf(pFile, " %d mal 5 \n", fuenf);
fprintf(pFile, " %d mal 6 \n", sechs);
fprintf(pFile, " %d mal 7 \n", sieben);
fprintf(pFile, " %d mal 8 \n", acht);
fprintf(pFile, " %d mal 9 \n", neun);
}
fclose(pFile);

}

void FileHandler::writetoPrsort(const char* name) {
    srand(time(NULL));
    int rand();
    FILE *pFile = NULL;
    pFile = fopen(name, "w");
    if(pFile != NULL){
        for(int i = 0; i != 100; i++){
            randomNumber = rand() % 100 ;
            fprintf(pFile, "%d, \n", randomNumber);
        }
        fclose(pFile);
    }
}

```

```

    }

}

void FileHandler::bubbleSort(const char* name) {
    FILE *exampleFile = NULL;
    exampleFile = fopen(name, "r");
    int u = 0;
    int a = 0;
    if(exampleFile != NULL) { // Abfangen, dass die File vorher
        existiert.
        while (!feof(exampleFile) && u <= 99) { // Bis Ende der Datei
            lesen.
            fscanf(exampleFile, "%d, ", &a); // Sucht nach ";" Zeichen
            und speichert die Eintraege in eine Array ab.
            sortList.push_back(a);
            //printf("%d: %d \n", u, sortList[u]); // Ausgabe
            auf Konsole zur Kontrolle
            u++;
        }
        fclose(exampleFile); // Datei schliessen.
    }

    int i, j,tmp;

    for (i = 1; i < 101 ; i++) //Bubble Sort
    {
        for (j = 0; j < 101 - i ; j++)
        {
            if (sortList[j] > sortList[j+1])
            {
                tmp = sortList[j];
                sortList[j] = sortList[j+1];
                sortList[j+1] = tmp;
            }
        }
    }
}

void FileHandler::writebubblesort(const char* name) {
    FILE *pFile = NULL;
    pFile = fopen(name, "w");
    if(pFile != NULL){
        for(int z = 0; z<=99; z++){
            fprintf(pFile,"%d, \n",sortList[z]);
        }
        fclose(pFile);
    }
}

*-----FILEHandler main-----*
#include <cstdlib>
#include <iostream>
#include <math.h>

```

```

#include "FileHandler.h"
#include <vector>

using namespace std;

/*
 *
 */
int main(int argc, char** argv){
    const char* name = "RandomZahlen.cvs";
    const char* zaehlungen = "Zaehlungen.txt";
    const char* pr2sort = "PR2_Sort.txt";
    const char* bubblesort = "PR2_BubbleSorted.txt";

    //Aufgabe Zaehlungen
    FileHandler myFile;
    myFile.writetoFile(name);
    myFile.readfromFile(name);
    myFile.search(6);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%

    *----- DataManagement.cpp -----*

/*
 * File:   DataManagement.cpp
 * Author: chabi
 *
 * Created on 11. November 2019, 18:57
 */
#include <cstdlib>
#include <iostream>
#include <math.h>
#include <vector>
#include "DataManagement.h"

DataManagement::DataManagement() {

}

void DataManagement::writetoFile(const char* name) {
    int x = 0;
    for(int i = 0; i <=MAX ; i++){
        x = x*2;
        dataList.push_back(x);
        if(x == NULL){
            x++;
        }
    }
    FILE *pFile = NULL;
    pFile = fopen(name, "w");

```

```

        if(pFile != NULL){
            for(i = dataList.begin(); i != dataList.end(); i++){
                fprintf(pFile,"%d,", *i);
            }
            fclose(pFile);
        }
    }

void DataManagement::readFromFile(const char* name) {
    // Auslesen einer Datei. Inhalt in einen Array abspeichern.
    FILE *exampleFile = NULL;
    exampleFile = fopen(name, "r"); // Datei
    oeffnen zu lesen. Modus "r" für readable.
    int myArray[MAX] = {0};
    int i = 0;
    if (exampleFile != NULL) { // Abfangen, dass die File vorher
    existiert.
        while (!feof(exampleFile) && i <= MAX) { // Bis Ende der
        Datei lesen.
            fscanf(exampleFile, "%d, ", &myArray[i]); // Sucht
            nach ";" Zeichen und speichert die Eintraege in eine Array ab.
            printf("%d: %d \n", i, myArray[i]); // Ausgabe
            auf Konsole zur Kontrolle
            i++;
        }
        fclose(exampleFile); // Datei schliessen.
    }
}

myFile.readandwritetoFile(zaehlungen);

//Sortieralgorithmen
myFile.writetoPrsort(pr2sort);
myFile.bubbleSort(pr2sort);
myFile.writebubblesort(bubblesort);

// printf("Die Zahl 4 kommt in der Datei genau %d mal vor \n",
myFile.search(4));

return 0;
}

*----- DataManagement.h -----*

/*
* File:   DataManagement.h
* Author: chabi
*
* Created on 11. November 2019, 18:57

```

```

    */

#ifndef DATAMANAGEMENT_H
#define DATAMANAGEMENT_H
#define SIEBEN 7

#include <cstdlib>
#include <iostream>
#include <math.h>
#include <vector>

#include <unordered_set>

using namespace std;

class DataManagement {
public:
    DataManagement();
    void writetoFile(const char* name);
    void readfromFile(const char* name);

private:
    vector<int> dataList;
    vector<int>::const_iterator i;
    const int MAX = 7;
};

#endif /* DATAMANAGEMENT_H */

*----- main Datamanegement -----*

#include <cstdlib>
#include <iostream>
#include <math.h>
#include <vector>
#include "DataManagement.h"

using namespace std;

/*
 *
 */
int main(int argc, char** argv) {

    const char* name = "Quadrante.cvs";
    DataManagement myData;
    myData.writetoFile(name);
    myData.readfromFile(name);

    return 0;
}

```

```

}

*----- tests -----*

#include "gtest/gtest.h"
#include "DataManagement.h"

TEST (TS1, TC1){

    const char* name = "Quadrante.csv";

    //    DataManagement myData;
    //    myData.writetoFile(name);

    // Auslesen einer Datei. Inhalt in einen Array abspeichern.
    FILE *exampleFile = NULL;
    exampleFile = fopen(name, "r"); // Datei
    oeffnen zu lesen. Modus "r" für readable.
    const int MAX = 7;
    int myArray[MAX] = {0};
    int i = 0;
    if (exampleFile != NULL) { // Abfangen, dass die File vorher
    existiert.
        while (!feof(exampleFile) && i <= MAX) { // Bis Ende der
        Datei lesen.
            fscanf(exampleFile, "%d,", &myArray[i]); // Sucht nach ";"
            Zeichen und speichert die Eintraege in eine Array ab.
            printf("%d: %d \n", i, myArray[i]); // Ausgabe
            auf Konsole zur Kontrolle
            i++;
        }
        EXPECT_EQ(myArray[0],0);
        fclose(exampleFile); // Datei schliessen.
    }

}

TEST (TS2, TC2){

    const char* name = "Quadrante.csv";

    // Auslesen einer Datei. Inhalt in einen Array abspeichern.
    FILE *exampleFile = NULL;
    exampleFile = fopen(name, "r"); // Datei
    oeffnen zu lesen. Modus "r" für readable.
    const int MAX = 7;
    int myArray[MAX] = {0};
    int i = 0;
    if (exampleFile != NULL) { // Abfangen, dass die File vorher
    existiert.
        while (!feof(exampleFile) && i <= MAX) { // Bis Ende der
        Datei lesen.

```

```

        fscanf(exampleFile, "%d,", &myArray[i]); // Sucht nach ";"
        Zeichen und speichert die Eintraege in eine Array ab.
        printf("%d: %d \n", i, myArray[i]); // Ausgabe
        auf Konsole zur Kontrolle
        i++;
    }
    EXPECT_EQ(myArray[4],16);
    fclose(exampleFile); // Datei schliessen.
}

}

TEST (TS3, TC3){

    const char* name = "Quadrante.csv";

    // Auslesen einer Datei. Inhalt in einen Array abspeichern.
    FILE *exampleFile = NULL;
    exampleFile = fopen(name, "r"); // Datei
    oeffnen zu lesen. Modus "r" für readable.
    const int MAX = 7;
    int myArray[MAX] = {0};
    int i = 0;
    if (exampleFile != NULL) { // Abfangen, dass die File vorher
    existiert.
        while (!feof(exampleFile) && i <= MAX) { // Bis Ende der
        Datei lesen.
            fscanf(exampleFile, "%d,", &myArray[i]); // Sucht nach ";"
            Zeichen und speichert die Eintraege in eine Array ab.
            printf("%d: %d \n", i, myArray[i]); // Ausgabe
            auf Konsole zur Kontrolle
            i++;
        }
        EXPECT_EQ(myArray[7],128);
        fclose(exampleFile); // Datei schliessen.
    }

}

```

Error using dbstatus

*Error: File: C:\PR2\FileHandlingChabir.m Line: 1 Column: 1
Invalid use of operator.*

Published with MATLAB® R2018b