Министерство образования РФ

Пермский национальный исследовательский политехнический университет

Кафедра ИТАС

ЛАБОРАТОРНАЯ РАБОТА № 9

ПО ИНФОРМАТИКЕ ЗА I СЕМЕСТР

Вариант 5

|  |
| --- |
| Выполнил студент:  Отинов Иван Юрьевич  Группа РИС-20-1бз  Шифр 20-ЭТФ-635  Кафедра ИТАС:  Полякова Ольга Андреевна |

ПЕРМЬ 2020

# Лабораторная работа №9 "Хранение данных на внешних носителях”

**Вариант №5**

**Цель:**

1. Получение практических навыков записи структурированной информации в файлы в стиле С;
2. Получение практических навыков записи структурированной информации в файлы в стиле С++;

**Задача:**

1. Используя ввод-вывод в стиле С создать файл и записать в него структурированные данные.
2. Вывести созданный файл на экран.
3. Удалить из файла данные в соответствии с вариантом.
4. Добавить в файл данные в соответствии с вариантом.
5. Вывести измененный файл на экран.
6. Используя ввод-вывод в стиле С++ создать файл и записать в него структурированные данные.
7. Вывести созданный файл на экран.
8. Удалить из файла данные в соответствии с вариантом.
9. Добавить в файл данные в соответствии с вариантом.
10. Вывести измененный файл на экран.

**Вариант задания:**

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | Структура "Человек":   * фамилия, имя, отчество; * год рождения; * рост; * вес. | Удалить все элементы с указанным ростом и весом. | Добавить K элементов в начало файла |

**Текст программы № 1**

#include <stdio.h>

#include <math.h >

int main() {

char str[100];

FILE\* fin;

fin = fopen("D:\\file.txt", "r");

printf("---FILE after load---\n");

int count = 0;

while (1) {

if (NULL != fgets(str, 100, fin)){

if (count == 4) {

count = 0;

printf("\n");

}

printf("%s", str);

count++;

}

else {

break;

}

}

fclose(fin);

int weight;

int height;

printf("\nInput height: ");

scanf("%d", &height);

printf("\nInput weight: ");

scanf("%d", &weight);

fin = fopen("D:\\file.txt", "r");

count = 0;

int currentRecord = 1;

int recordByHeight = 0;

int recordByWeight = 0;

int currentWeight = 0;

while (1) {

if (NULL != fgets(str, 100, fin)) {

if (count == 4) {

count = 0;

currentRecord++;

}

if (count == 2) {

int currentHeight = 0;

int i;

for (i = 0; str[i] != '\0'; i++) {

}

i -=2;

for (int j = 0; str[j] != '\0'; j++) {

currentHeight += ((int)str[j] - (int)'0') \* (int)pow(10, i);

i--;

}

if (currentHeight == height) {

recordByHeight = currentRecord;

}

}

if (count == 3) {

int currentWeight = 0;

int i;

for (i = 0; str[i] != '\n'; i++);

i--;

for (int j = 0; str[j] != '\0'; j++) {

currentWeight += ((int)str[j] - (int)'0') \* (int)pow(10, i);

i--;

}

if (currentWeight == weight) {

recordByWeight = currentRecord;

}

}

count++;

}

else {

break;

}

}

fclose(fin);

int isNeedDeleted = 0;

if (recordByHeight == recordByWeight) {

isNeedDeleted = recordByHeight;

}

FILE\* temp;

if (isNeedDeleted) {

fin = fopen("D:\\file.txt", "r");

temp = fopen("D:\\temp.txt", "wb");

count = 0;

currentRecord = 1;

while (1) {

if (NULL != fgets(str, 100, fin)) {

if (count == 4) {

count = 0;

currentRecord++;

}

if (currentRecord != isNeedDeleted) {

fprintf(temp, "%s", str);

}

count++;

}

else {

break;

}

}

fclose(temp);

fclose(fin);

}

temp = fopen("D:\\temp.txt", "r");

printf("\n---File after delete---\n");

while (1) {

if (NULL != fgets(str, 100, temp)) {

if (count == 4) {

count = 0;

printf("\n");

}

printf("%s", str);

count++;

}

else {

break;

}

}

fclose(temp);

int k = 2;

temp = fopen("D:\\temp.txt", "a");

printf("\n");

char input[50];

for (int i = 0; i < k; i++) {

for (int j = 0; j < 4; j++) {

if (j == 0) {

printf("Input name:\n");

} else if (j == 1) {

printf("Input birthday:\n");

} else if (j == 2) {

printf("Input height:\n");

} else {

printf("Input weight:\n");

}

scanf("%s", input);

fprintf(temp, "%s", input);

fprintf(temp, "%s", "\n");

}

}

fclose(temp);

temp = fopen("D:\\temp.txt", "r");

printf("\n---File after add---\n");

while (1) {

if (NULL != fgets(str, 100, temp)) {

if (count == 4) {

count = 0;

printf("\n");

}

printf("%s", str);

count++;

}

else {

break;

}

}

fclose(temp);

}

**Результаты работы программы № 1:**



**Текст программы № 2**

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

void readFile(char\* fileName);

void writeInFile(char\* fileName);

void findByHeightAndWeight(char \*fileName, int height, int weight);

void writeToStart(char\* fileName);

int main() {

char fileName[] = "D:\\file.txt";

int n;

cout << "Inputn N: ";

cin >> n;

for (int i = 0; i < n; i++) {

writeInFile(fileName);

}

readFile(fileName);

int height, weight;

cout << "Input height: ";

cin >> height;

cout << "Input weight: ";

cin >> weight;

findByHeightAndWeight(fileName, height, weight);

readFile(fileName);

writeToStart(fileName);

readFile(fileName);

}

void writeInFile(char\* fileName) {

cout << "Write new record:\n";

char name1[50];

char name2[50];

char name3[50];

char year[50];

char height[50];

char weight[50];

cout << "Input name1: ";

cin >> name1;

cout << "Input name2: ";

cin >> name2;

cout << "Input name3: ";

cin >> name3;

cout << "Input birthday: ";

cin >> year;

cout << "Input height: ";

cin >> height;

cout << "Input weight: ";

cin >> weight;

ofstream out(fileName, ios::app);

if (out.is\_open()) {

out << name1 << " " << name2 << " " << name3 << " " << endl;

out << year << endl;

out << height << endl;

out << weight << endl;

}

out.close();

}

void readFile(char \*fileName) {

string line;

ifstream in(fileName);

cout << "FILE " << fileName;

cout << "\n--------------------------\n";

if (in.is\_open()){

while (getline(in, line)){

cout << line << std::endl;

}

}

cout << "--------------------------\n";

in.close();

}

void findByHeightAndWeight(char \*fileName, int height, int weight) {

string line;

ifstream in(fileName);

int count = 0;

int currentRecord = 1;

int recordByHeight = 0;

int recordByWeight = 0;

if (in.is\_open()) {

while (getline(in, line)) {

if (count == 2) {

if (stoi(line) == height) {

recordByHeight = currentRecord;

}

count++;

} else if (count == 3) {

if (stoi(line) == weight) {

recordByWeight = currentRecord;

}

currentRecord++;

count = 0;

} else {

count++;

}

}

}

if (recordByHeight != recordByWeight) {

in.close();

return;

}

int recordToDelete = recordByHeight;

in.close();

in.open(fileName);

ofstream out("D:\\temp.txt", ios\_base::trunc);

currentRecord = 1;

count = 0;

if (in.is\_open()) {

while (getline(in, line)) {

if (currentRecord != recordToDelete) {

out << line << endl;

}

if (count == 3) {

currentRecord++;

count = 0;

} else {

count++;

}

}

}

out.close();

in.close();

out.open(fileName, ios\_base::trunc);

in.open("D:\\temp.txt", ios\_base::in);

if (in.is\_open()) {

while (getline(in, line)) {

out << line << endl;

}

}

out.close();

in.close();

}

void writeToStart(char \*fileName) {

ofstream out;

ifstream in(fileName);

string line;

out.open("D:\\temp.txt", ios\_base::trunc);

if (in.is\_open()) {

while (getline(in, line)) {

out << line << endl;

}

}

in.close();

out.close();

in.open("D:\\temp.txt");

out.open(fileName, ios\_base::trunc);

out.close();

int n;

cout << "Inputn N: ";

cin >> n;

for (int i = 0; i < n; i++) {

writeInFile(fileName);

}

out.open(fileName, ios\_base::app);

if (in.is\_open()) {

while (getline(in, line)) {

out << line << endl;

}

}

}

**Результаты работы программы № 2:**

