Пермский Национальный Исследовательский  
Политехнический Университет

**Лабораторная работа № 10**

Теория алгоритмов

за 2 семестр

Вариант № 1

Выполнил:

Студент группы РИС 20-1-бз

Курганов Н.В.

20-ЭТФ-631

Проверила:

Доцент кафедры ИТАС

Полякова О.А.

2021

**Лабораторная работа № 10**

Базы Данных

### 1. Постановка задачи

Написать приложение для работы с простой базой данных, хранящей информацию об объекте на внешнем носителе. Приложение должно выполнять следующие функции:

1. Создание базы данных, содержащей записи указанного формата.
2. Просмотр базы данных.
3. Удаление элементов из базы данных (по ключу/ по номеру).
4. Корректировка элементов в базе данных (по ключу / по номеру).
5. Добавление элементов в базу данных (в начало / в конец/ с заданным номером).
6. Выполнение задания, указанного в варианте.

### Вариант

1. В типизированном файле хранится информация о средней температуре за месяц. Программа должна
   1. Добавлять, удалять, корректировать, позволять просматривать записи файла.
   2. Определять:
      1. дни, в которые температура поднималась выше средней за месяц;
      2. самый длинный отрезок между днями с отрицательной температурой.

#include <iostream>

#include <conio.h>

#include <fstream>

using namespace std;

//Написать приложение для работы с простой базой данных,

//хранящей информацию об объекте на внешнем носителе.Приложение должно выполнять следующие функции :

//1. Создание базы данных, содержащей записи указанного формата.

//2. Просмотр базы данных.

//3. Удаление элементов из базы данных(по ключу / по номеру).

//4. Корректировка элементов в базе данных(по ключу / по номеру).

//5. Добавление элементов в базу данных(в начало / в конец / с заданным номером).

//6. Выполнение задания, указанного в варианте.

//Вариант 1

//В типизированном файле хранится информация о средней температуре за месяц.Программа должна

//a.Добавлять, удалять, корректировать, позволять просматривать записи файла.

//b.Определять:

//i.дни, в которые температура поднималась выше средней за месяц;

//ii.самый длинный отрезок между днями с отрицательной температурой.

//структура температур по каждому дню

struct Temperature {

int day;//день

int avgD;//средняя температура за день

};

const char\* cureFileName;//файл

const char\* realFileName = "dataBase.bd";//ссылка на основной файл с данными

int cureMassLen = 0;//текущая длина массива

//Функция сохранения данных в файл

void save(const char\* filename, struct Temperature\* mas, int n)

{

fstream f(filename, ios::out);//двунаправленный файловый поток

for (int i = 0; i < cureMassLen; i++)

{

//добавляем данные в файл

f << mas[i].day; f << "\n";

f << mas[i].avgD; f << "\n";

}

f.close();//закрытие потока

}

//Функция изменения данных по ключу

Temperature\* change\_elem(Temperature\* b, int day, int avgT)

{

for (int i = 0; i < cureMassLen; i++)

{

if (b[i].day == day)//если i запись дня равна искомому то меняем

{

b[i].avgD = avgT;

}

}

save(cureFileName, b, cureMassLen);//сохраняем массив в текущий файл

return b;

}

//Функция добавления данных по id

Temperature\* add\_elem(const char\* filename, Temperature\* b,int day, int avgT, int n)

{

int i = 0;

Temperature\* a = new Temperature[cureMassLen + 1];

for (; i < cureMassLen+1; i++)

{

if (i == n)//если номер равен искомому то заносим новые значения в массив a

{

a[i].day = day;

a[i].avgD = avgT;

}//если текущий индекс меньше искомого переписываем массив b в a

else if (i < n) {

a[i].day = b[i].day;

a[i].avgD = b[i].avgD;

}

else //если текущий индекс больше искомого переписываем массив b в a

{

a[i].day = b[i-1].day;

a[i].avgD = b[i-1].avgD;

}

}

cureMassLen++;//увеличиваем текущий размер на 1

save(cureFileName, a, cureMassLen);

return a;

}

//вычисляем дни в которых температура больше средней

void days\_where\_more\_than\_middel(Temperature\* b)

{

int sum = 0;//сумма всех температур

for (int i = 0; i < cureMassLen; i++)

{

sum += b[i].avgD;

}

double avgM = ((double)sum )/ ((double)cureMassLen);//вычисляем среднюю температур

cout << "Средняя температура: " << avgM;//выводи её

cout << "\nДень\tТемпература\n";

for (int i = 0; i < cureMassLen; i++)//выводи все дни в которых температура больше средней

{

if (b[i].avgD > avgM) {

cout << b[i].day << "\t" << b[i].avgD << "\n";

}

}

}

//вычисляем наибольший промежуток времени с отрицательной температурой

void cold(Temperature\* b) {

int countMax = 0;//счётчик

int tmp = 0;//функция для сохранения

int id = -1;//id

for (int i = 0; i < cureMassLen-1; i++)

{

if (b[i].avgD < 0 && b[i + 1].avgD < 0) //если текущий и следующий элемент меньше 1

{

countMax++;//увеличиваем кол-во подряд идущих отрицательных значений

if (i == cureMassLen - 2) //обрабатываем последний ответ

{

if (countMax > tmp)

{

id = i+1;

tmp = countMax;

}

}

}

else if (b[i].avgD < 0 && b[i + 1].avgD >= 0) //если следующий элемент не отрицательный

{

if (countMax > tmp) {

//если кол-во подсчитанного больше предыдущего сохраняем кол-во и id последнего элемента

tmp = countMax;

id = i;

}

countMax = 0;//обнуляем для подсчёта

}

}

if (id >= 0) //если найден хотя бы 1 элемент

{

cout << "\nДень\tТемпература\n";//выводим все элементы

for (int i = id - tmp; i <= id; i++) {

cout << b[i].day << "\t" << b[i].avgD << "\n";

}

}

else {

cout << "\nВсе дни с положительной температурой\n";

}

}

//удалить k-ый элемент из файла

Temperature\* del\_num(const char\* filename, Temperature\* b, int k, int n)

{

fstream f(filename, ios::out);//вспомогательный файл

//считываем данные из исходного файла в буфер

int i = 0;

if (cureMassLen > 0) {

Temperature\* a = new Temperature[cureMassLen - 1];//создаём новый массив в котором на 1 меньше элементов

for (; i < cureMassLen; i++)

{

if (i != k)//если номер записи не равен k

{

f << b[i].day; f << "\n";

f << b[i].avgD; f << "\n";

a[i].day = b[i].day;

a[i].avgD = b[i].avgD;

}

}

f.close();//закрываем исходный файл

cureMassLen--;

if (n >= cureMassLen \* 2)

{

remove(realFileName);//удаляем исходный файл

rename(filename, realFileName);

}//переименовываем временный файл

return a;

}

return new Temperature[0];

}

//удалить по ключу

Temperature\* del\_key(const char\* filename, Temperature\* b, int k, int n)

{

fstream f(filename, ios::out);//вспомогательный файл

//считываем данные из исходного файла в буфер

int i = 0;

Temperature\* a = new Temperature[cureMassLen-1];

for (int j = 0; i < cureMassLen-1; i++, j++)

{

if (b[i].day == k)//если день записи равен k

{

j++; //увеличиваем индекс для основного массива чтобы не переписывать k элемент в новый массив

}

f << b[j].day; f << "\n";

f << b[j].avgD; f << "\n";

a[i].day = b[j].day;

a[i].avgD = b[j].avgD;

}

f.close();//закрываем исходный файл

cureMassLen--;

if (n >= cureMassLen \* 2){

remove(realFileName);//удаляем исходный файл

rename(filename, realFileName);

}//переименовываем временный файл

return a;

}

// загрузка из файла массива структур

Temperature\* load(const char\* filename)

{

fstream f;

Temperature a;

Temperature\* p;

f.open(filename, ios::in);//открываем поток для чтения

if (f) //если файл существует

{

int num = 0;//кол-во элементов

do

{

//достаём данные

f >> a.day;

f >> a.avgD;

if (f.eof())break;

num++;

} while (!f.eof());

f.close();//закрытие потока

f.open(filename, ios::in);

p = new Temperature[num];

if (num > 0) {

num = 0;

do

{

//достаём данные

f >> a.day;

f >> a.avgD;

if (f.eof())break;

p[num] = a;//записываем в массив

num++;

} while (!f.eof());

}

cureMassLen = num;

f.close();//закрытие потока

return p;

}

return new Temperature[0];

}

//создаём массив из n элементов

Temperature\* create(int n = 30) {

Temperature\* p = new Temperature[n];

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

cout << "Введите среднюю температур в " << i + 1 << " день: ";

cin >> p[i].avgD;

p[i].day = i + 1;

}

return p;

}

//вывод массива

void show(Temperature\* dataBase,int n) {

cout << "\nДень\tТемпература\n";

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

cout << dataBase[i].day <<"\t"<<dataBase[i].avgD<<"\n";

}

}

int main()

{

Temperature\* mas = load(realFileName);//загружаем

if (cureMassLen == 0)

mas = new Temperature[0];

int n = 0;

setlocale(LC\_ALL, "Rus");

while (1) {

cout << "1 - Создать БД, 2 - Показать базу данных, 3 - Удалить элемент, 4 - Добавить элемент, 5 - изменить элемент, 6 - вывести дни в которых температура выше средней, 7 - Вывести самый длинный отрезок времени с отрицательной температурой, 8 - Отменить удаления, 9 - Сохранить изменения";

cout << "\n";

char c = \_getch();//получаем нажатую кнопку

int tmp, day, id;

switch (c) {

case '1':

//создаём массив дней

cout << "Введите размер массива: ";

cin >> cureMassLen;

mas = create(cureMassLen);

n = cureMassLen;

cureFileName = realFileName;

save(cureFileName, mas, n);

break;

case '2':

//выводим массив

if (cureMassLen != 0) {

cout << "\n";

show(mas, cureMassLen);

}

else {

cout << "Нет данных ";

}

break;

case '3':

//удаляем данные

if (cureMassLen != 0) {

cout << "1 - Удалить по ключу, 2 - Удалить по номеру ";

cout << "\n";

char c = \_getch();

switch (c)

{

case '1':

//по ключу

cout << "Введите день: ";

int day;

cin >> day;

cureFileName = "temp";

mas = del\_key(cureFileName, mas, day, n);

cout << "\n";

break;

case '2':

//по номеру

cout << "Введите ID: ";

int num;

cin >> num;

cureFileName = "temp";

mas = del\_num(cureFileName, mas, num, n);

cout << "\n";

break;

default:

break;

}

}

else {

cout << "Нет данных ";

} break;

case '4':

//добавление данных

cout << "1 - Добавить в начало, 2 - Добавить после номера, 3 - Добавить в конец списка ";

c = \_getch();

switch (c)

{

case '1':

//вставить в начало

cout << "Введите день: ";

cin >> day;

cout << "Введите температуру: ";

cin >> tmp;

mas = add\_elem(cureFileName, mas, day, tmp, 0);

break;

case '2':

//вставить после k-ого элемента

cout << "Введите на какое место добавить: ";

cin >> id;

cout << "Введите день: ";

cin >> day;

cout << "Введите температуру: ";

cin >> tmp;

mas = add\_elem(cureFileName, mas, day, tmp, id);

break;

case '3':

//вставить в конец

cout << "Введите день: ";

cin >> day;

cout << "Введите температуру: ";

cin >> tmp;

mas = add\_elem(cureFileName, mas, day, tmp, cureMassLen-1);

break;

default:

break;

}

break;

case '5':

//изменить элемент по ключу

cout << "Введите день, температуру которого нужно изменить: ";

cin >> day;

cout << "Введите температуру: ";

cin >> tmp;

mas = change\_elem(mas, day, tmp);

break;

case '6':

//вывести все дни в которых температура больше средней

days\_where\_more\_than\_middel(mas);

break;

case '7':

//вывести наибольшее подряд идущее кол-во дней с отрицательной температурой

cold(mas);

break;

case '8':

//отменяем изменения загружая основной файл

mas = load(realFileName);

cureFileName = realFileName;

break;

case '9':

//сохраняем изменения и удаляем текущий файл если он существует

save(cureFileName, mas, cureMassLen);

if (cureFileName != realFileName){

remove(realFileName);

rename(cureFileName, realFileName);

cureFileName = realFileName;

}

break;

case 27:

return 1;

}

}

}

