Министерство образования РФ

Пермский национальный исследовательский политехнический университет

Кафедра ИТАС

ЛАБОРАТОРНАЯ РАБОТА № 10

ПО ТЕОРИИ АЛГОРИТМОВ ЗА II СЕМЕСТР

Вариант 15

Выполнил студент:

Сташков Арсений Дмитриевич

Группа РИС-20-1бз

Шифр 20-ЭТФ-644

Проверила:

Доцент кафедры ИТАС

Полякова Ольга Андреевна

ПЕРМЬ 2022

# Лабораторная работа №10 "Базы данных"

**Вариант №15**

## **Задача:** 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.

15. Типизированный файл содержит данные о клиентах банка, получивших кредит (ФИО клиента, сумма кредита, вид кредита, срок, на который выдан кредит). Клиент может взять несколько кредитов.

* 1. Программа должна
     1. Добавлять, удалять, корректировать, позволять просматривать записи файлов.
  2. Выполнять:
     1. поиск клиентов, взявших кредит на N месяцев;
     2. поиск клиентов по шаблону (3 первые буквы фамилии);

**Анализ задачи:** Для упрощения восприятия программы она была разбита на 3 файла. В 1 файле мы создаем меню и вызываем функции из 2 файла. 3 файл содержит описание этих функций.

**Решение:**

Файл Базы данных.cpp

#include <iostream>

#include "Functions.h"

using namespace std;

int stateMenu;

void Menu() {

cout << "Веберите действие:\n\n"

"0. Выйти из программы\n"

"1. Изменить данные\n"

"2. Показать данные\n"

"3. Удалить данные\n"

"4. Добавить данные\n\n";

cin >> stateMenu;

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

//DataInitialization();

Menu();

int actions;

string fileName;

while (stateMenu != 0) {

switch (stateMenu) {

case 1:

// Чистим консоль

system("cls");

//DataEntry();

DataCleaning();

AddData();

system("pause");

system("cls");

Menu();

break;

case 2:

system("cls");

Print();

system("pause");

system("cls");

Menu();

break;

case 3:

system("cls");

DataCleaning();

system("pause");

system("cls");

Menu();

break;

case 4:

system("cls");

AddData();

system("pause");

system("cls");

Menu();

break;

default:

cout << "Пункт меню выбран неверно!\n";

break;

}

}

system("cls");

cout << "Работа завершена\n";

system("pause");

return 0;

}

Файл Functions.cpp

// Пишем файл с описанием функций

#include "Functions.h"

// Инициализация данных

void DataInitialization() {

ofstream buf("Buffer.txt");

if (!buf)

cout << "Ошибка создания буферного файла!/n";

buf.close();

}

// Ввод данных

void DataEntry() {

string name;

int creditAmount;

string creditType;

int term;

int n;

cout << "Введите количество данных: ";

cin >> n;

// Открываем буферный файл и записываем данные в него

ofstream record("Buffer.txt", ios::app);

if (record) {

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

cout << "Введите имя: ";

cin >> name;

cout << "Введите сумму кредита (в руб.): ";

cin >> creditAmount;

cout << "Введите вид кредита: ";

cin >> creditType;

cout << "Введите срок выдачи кредита (кол-во месяцев): ";

cin >> term;

record << name << "\n";

record << creditAmount << " руб.\n";

record << creditType << "\n";

if (i < n - 1) {

record << term << " мес.\n";

}

else {

record << term << " мес.";

}

record << endl;

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

}

}

else {

cout << "Ошибка открытия файла!\n";

}

record.close();

}

// Чтение данных

void DataReading(string fileName) {

ifstream reading(fileName);

ofstream record("Buffer.txt", ios::out);

if (reading) {

if (record) {

string name;

int creditAmount;

string creditType;

int term;

int n;

reading >> n;

record << n << "\n";

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

reading >> name;

reading >> creditAmount;

reading >> creditType;

reading >> term;

record << name << "\n";

record << creditAmount << "\n";

record << creditType << "\n";

if (i < n - 1) {

record << term << "\n";

}

else {

record << term;

}

}

cout << "Данные считаны!\n";

}

else

cout << "Ошибка открытия буфера!\n";

}

else

cout << "Ошибка открытия файла!\n";

// Закрытие файла

reading.close();

record.close();

}

// Вывод данных

void Print() {

ifstream reading("Buffer.txt");

if (reading) {

string line;

string name;

string creditAmount;

string creditType;

string term;

int n;

cout << "Введите количество данных: ";

cin >> n;

cout << "\n";

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

reading >> line;

cout << line << "\n";

reading >> name;

cout << "Имя: " << name << "\n";

reading >> creditAmount;

cout << "Сумма кредита: " << creditAmount << " руб.\n";

reading >> creditType;

cout << "Вид кредита: " << creditType << "\n";

reading >> term;

cout << "Срок выдачи кредита: " << term << " мес.\n";

}

}

else {

cout << "Ошибка открытия файла!\n";

}

reading.close();

}

// Очистка данных

void DataCleaning() {

fstream buf("Buffer.txt", ios::out);

if (!buf) {

cout << "Ошибка открытия буферного файла!\n";

}

cout << "Файл очищен\n";

buf.clear();

buf.close();

}

// Изменение данных

/\*void DataChange() {

Copy();

ifstream reading("Buffer2.txt");

ofstream record("Buffer.txt", ios::out);

if (reading) {

if (record) {

string name, creditType;

int creditAmount, term, n, n2;

reading >> n;

cout << "Выберите номер изменяемого элемента (от 1 до " << n << "): ";

cin >> n2;

n2--;

system("cls");

record << n << "\n";

if (n2 >= 0 && n2 < n) {

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

if (i != n2) {

reading >> name;

record << name << "\n";

reading >> creditAmount;

record << creditAmount << "\n";

reading >> creditType;

record << creditType << "\n";

reading >> term;

if (i < n - 1) {

record << term << "\n";

}

else {

record << term;

}

}

else {

cout << "Введите данные №1 (имя): ";

cin >> name;

cout << "Введите данные №2 (Сумма кредита): ";

cin >> creditAmount;

cout << "Введите данные №3 (Вид кредита): ";

cin >> creditType;

cout << "Введите данные №4 (Срок выдачи кредита): ";

cin >> term;

record << name << "\n";

record << creditAmount << "\n";

record << creditType << "\n";

if (i < n - 1) {

record << term << "\n";

}

else {

record << term;

}

reading >> name;

reading >> creditAmount;

reading >> creditType;

reading >> term;

}

}

cout << "Данные изменены!\n";

}

else

"Номер введен неверно!";

}

else

cout << "Ошибка открытия файла!\n";

}

else

cout << "Ошибка открытия буферного файла!\n";

record.close();

reading.close();

remove("Buffer2.txt");

}\*/

// Копирование данных

void Copy() {

ifstream reading("Buffer.txt");

ofstream record("Buffer2.txt", ios::out);

if (reading)

{

if (record)

{

string name;

int creditAmount;

string creditType;

int term;

int n;

reading >> n;

record << n << "\n";

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

{

reading >> name;

record << name << "\n";

reading >> creditAmount;

record << creditAmount << "\n";

reading >> creditType;

record << creditType << "\n";

reading >> term;

if (i < n - 1)

{

record << term << "\n";

}

else

{

record << term;

}

}

}

else

cout << "Ошибка открытия файла!\n";

}

else

cout << "Ошибка открытия буферного файла!\n";

record.close();

reading.close();

}

// Кол-во данных из файла

int AmountOfData() {

ifstream buf("Buffer.txt");

int n;

if (buf)

{

buf >> n;

}

else

{

cout << "Ошибка открытия буферного файла!\n";

}

buf.close();

return n;

}

// Удаление данных

/\*void DeleteData() {

Copy();

ifstream reading("Buffer2.txt");

ofstream record("Buffer.txt", ios::out);

if (reading)

{

if (record)

{

string name, creditType;

int creditAmount, term, n, n2;

reading >> n;

int b = n - 1;

cout << "Выберите номер изменяемого элемента от 1 до " << n << "): ";

cin >> n2;

n2--;

system("cls");

if (n2 >= 0 && n2 < n)

{

for (int i = 0; 1 < n; i++)

{

if (i != n2)

{

reading >> name;

record << name << "\n";

reading >> creditAmount;

record << creditAmount << "\n";

reading >> creditType;

record << creditType << "\n";

reading >> term;

if (i < n - 1)

{

record << term << "\n";

}

else

{

record << term;

}

}

else

{

reading >> name;

reading >> creditAmount;

reading >> creditType;

reading >> term;

}

}

cout << "Данные удалены!\n";

}

else

cout << "Номер введен неверно!\n";

}

else

cout << "Ошибка открытия файла!\n";

}

else

cout << "Ошибка открытия буферного файла!\n";

record.close();

reading.close();

//remove("Buffer2.txt");

}\*/

// Добавление данных

void AddData() {

string name;

int creditAmount;

string creditType;

int term;

int n = AmountOfData() + 1;

ofstream record("Buffer.txt", ios::app);

ofstream record2("Buffer.txt", ios::out | ios::in);

if (!record2) {

cout << "Ошибка открытия буферного файла!\n";

}

if (record) {

cout << "Введите имя: ";

cin >> name;

cout << "Введите сумму кредита (в руб.): ";

cin >> creditAmount;

cout << "Введите вид кредита: ";

cin >> creditType;

cout << "Введите срок выдачи кредита (кол-во месяцев): ";

cin >> term;

record << "----------------------\n";

record << name << "\n";

record << creditAmount << "\n";

record << creditType << "\n";

record << term << "\n";

}

else {

cout << "Ошибка открытия файла!\n";

}

record.close();

record2.close();

}

// Сохранение данных

void SaveData(string fileName) {

ifstream reading("Buffer.txt");

ofstream record(fileName, ios::out);

if (reading) {

if (record) {

string name;

int creditAmount;

string creditType;

int term;

int n;

reading >> n;

record << n << "\n";

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

reading >> name;

record << name << "\n";

reading >> creditAmount;

record << creditAmount << "\n";

reading >> creditType;

record << creditType << "\n";

reading >> term;

if (i < n - 1) {

record << term << "\n";

}

else {

record << term;

}

}

cout << "Данные сохранены в файле" << fileName << "\n";

}

else

cout << "Ошибка открытия буферного файла!\n";

}

else

cout << "Ошибка открытия файла!\n";

record.close();

reading.close();

}

Файл Functions.h

// Пишем файл с прототипами функций

#pragma once

#include <iostream>

#include <fstream>

#include <string>

#include <Windows.h>

using namespace std;

// Прототипы функций

void DataInitialization();

void DataEntry();

void DataReading(string fileName);

void Print();

void DataCleaning();

//void DataChange();

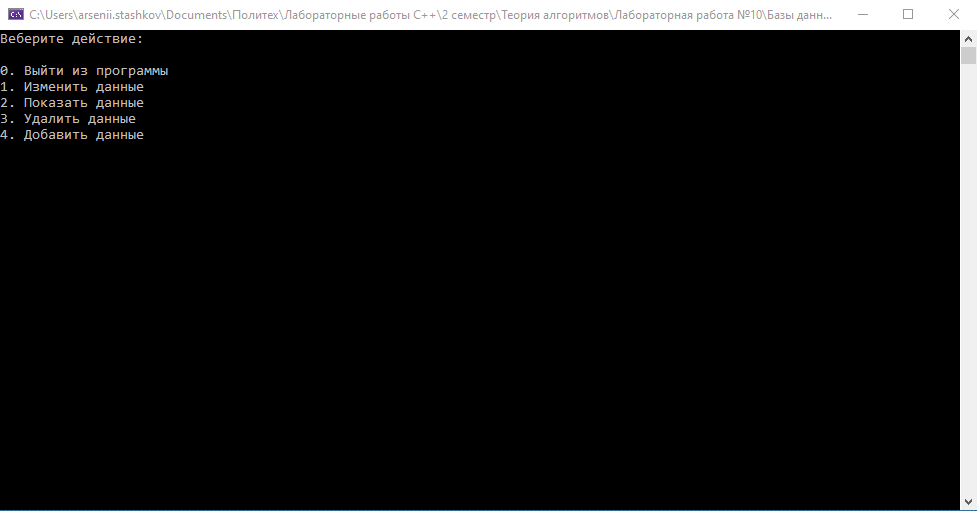
void Copy();

int AmountOfData();

//void DeleteData();

void AddData();

void SaveData(string fileName);



# 

