Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

направление подготовки: 09.03.01 – «Информатика и вычислительная техника»

**ОТЧЁТ**

**Лабораторная работа №8**

**«Блоковый ввод-вывод»**

**Вариант 8**

Выполнила работу:

студентка гр. ИВТ-24-2б

Малая Алина Александровна

Проверил:

Доцент кафедры ИТАС

Полякова Ольга Андреевна

(оценка) (подпись)

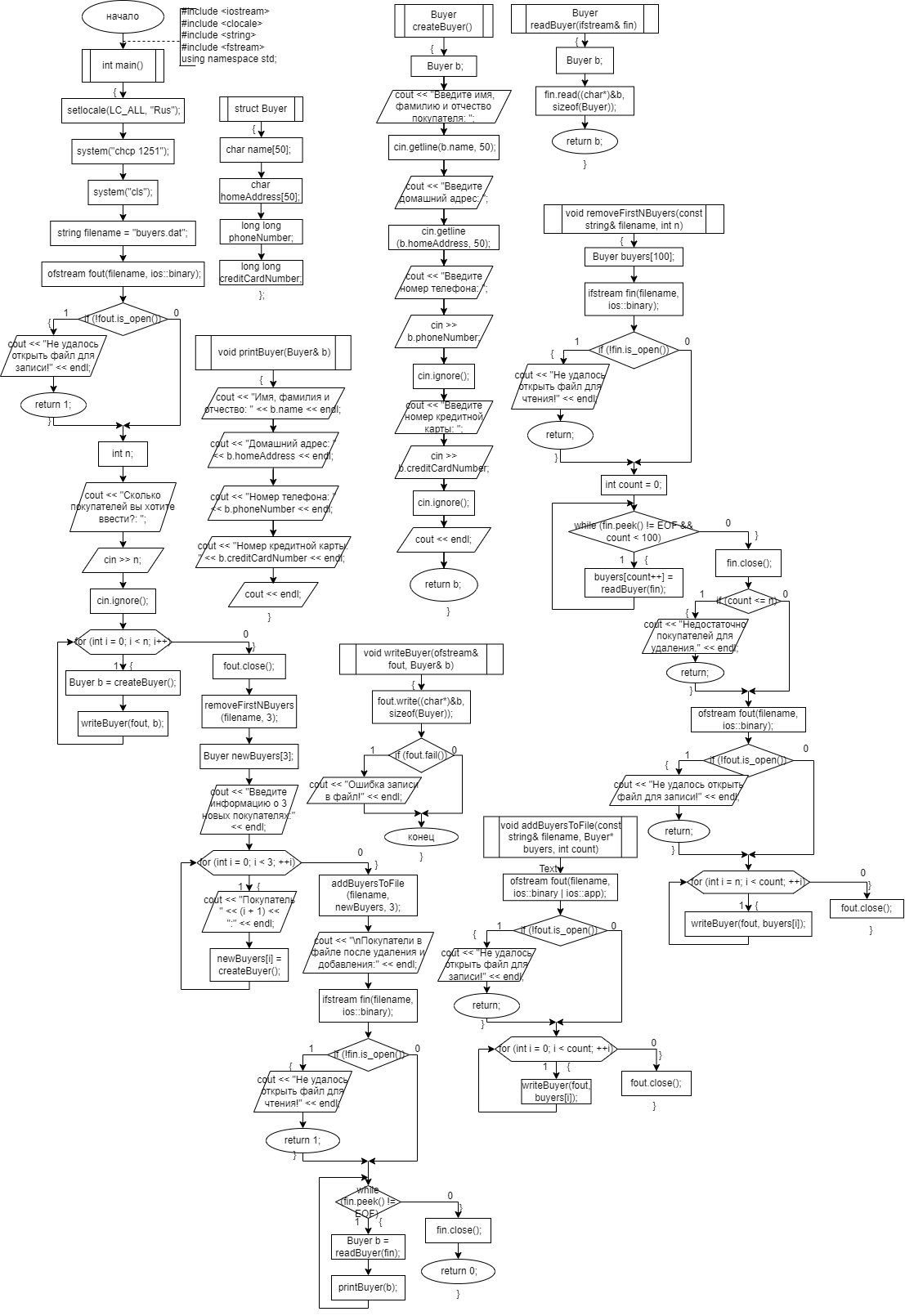
(дата)

г. Пермь, 2025

**Постановка задачи**

Написать программу на С++, которая должна:

Удалить 3 элемента из начала файла и добавить 3 новых элемента в конец файла.

**Блок – схема**

**Код программы**

#include <iostream>

#include <clocale>

#include <string>

#include <fstream>

using namespace std;

struct Buyer // покупатель

{

char name[50]; // имя, фамилия, отчество

char homeAddress[50]; // домашний адрес

long long phoneNumber; // номер телефона

long long creditCardNumber; // номер кредитной карты

};

Buyer createBuyer() // создаем объект Покупатель с помощью ввода и вывода

{

Buyer b;

cout << "Введите имя, фамилию и отчество покупателя: ";

cin.getline(b.name, 50);

cout << "Введите домашний адрес: ";

cin.getline(b.homeAddress, 50);

cout << "Введите номер телефона: ";

cin >> b.phoneNumber;

cin.ignore();

cout << "Введите номер кредитной карты: ";

cin >> b.creditCardNumber;

cin.ignore();

cout << endl;

return b;

}

void printBuyer(Buyer& b) // Функция для вывода информации о покупателе на экран

{

cout << "Имя, фамилия и отчество: " << b.name << endl; // Выводим ФИО

cout << "Домашний адрес: " << b.homeAddress << endl; // Выводим домашний адрес

cout << "Номер телефона: " << b.phoneNumber << endl; // Выводим номер телефона

cout << "Номер кредитной карты: " << b.creditCardNumber << endl; // Выводим номер кредитной карты

cout << endl;

}

void writeBuyer(ofstream& fout, Buyer& b) // Функция для записи данных о покупателе в файл

{

fout.write((char\*)&b, sizeof(Buyer)); // Записываем объект Buyer (b) в файл, используя бинарный формат.

if (fout.fail()) // Проверяем, произошла ли ошибка при записи в файл

{

cout << "Ошибка записи в файл!" << endl; // Если произошла ошибка, выводим сообщение об ошибке

}

}

Buyer readBuyer(ifstream& fin) // Функция для чтения данных о покупателе из файла

{

Buyer b; // Создаем объект Buyer для хранения прочитанных данных

fin.read((char\*)&b, sizeof(Buyer)); // Читаем данные из файла и сохраняем их в объект b

return b; // Возвращаем объект Buyer с прочитанными данными

}

void removeFirstNBuyers(const string& filename, int n)

{

Buyer buyers[100]; // Массив для временного хранения

ifstream fin(filename, ios::binary);

if (!fin.is\_open())

{

cout << "Не удалось открыть файл для чтения!" << endl;

return;

}

// Читаем всех покупателей из файла

int count = 0;

while (fin.peek() != EOF && count < 100)

{

buyers[count++] = readBuyer(fin);

}

fin.close();

// Проверяем, достаточно ли покупателей для удаления

if (count <= n)

{

cout << "Недостаточно покупателей для удаления." << endl;

return;

}

// Открываем файл для записи и записываем оставшихся покупателей

ofstream fout(filename, ios::binary);

if (!fout.is\_open())

{

cout << "Не удалось открыть файл для записи!" << endl;

return;

}

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

{

writeBuyer(fout, buyers[i]);

}

fout.close();

}

void addBuyersToFile(const string& filename, Buyer\* buyers, int count)

{

ofstream fout(filename, ios::binary | ios::app); // Открываем файл в режиме добавления

if (!fout.is\_open())

{

cout << "Не удалось открыть файл для записи!" << endl;

return;

}

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

{

writeBuyer(fout, buyers[i]);

}

fout.close();

}

int main()

{

setlocale(LC\_ALL, "Rus");

system("chcp 1251");

system("cls");

string filename = "buyers.dat";

ofstream fout(filename, ios::binary);

if (!fout.is\_open())

{

cout << "Не удалось открыть файл для записи!" << endl;

return 1;

}

int n;

cout << "Сколько покупателей вы хотите ввести?: ";

cin >> n;

cin.ignore();

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

{

Buyer b = createBuyer();

writeBuyer(fout, b);

}

fout.close();

removeFirstNBuyers(filename, 3); // Удаляем первых 3 покупателей

Buyer newBuyers[3]; // Добавляем новых покупателей

cout << "Введите информацию о 3 новых покупателях:" << endl;

for (int i = 0; i < 3; ++i)

{

cout << "Покупатель " << (i + 1) << ":" << endl;

newBuyers[i] = createBuyer();

}

addBuyersToFile(filename, newBuyers, 3); //Добавляем новых покупателей в файл

// Вывод оставшихся покупателей в файле

cout << "\nПокупатели в файле после удаления и добавления:" << endl;

ifstream fin(filename, ios::binary);

if (!fin.is\_open())

{

cout << "Не удалось открыть файл для чтения!" << endl;

return 1;

}

while (fin.peek() != EOF)

{

Buyer b = readBuyer(fin);

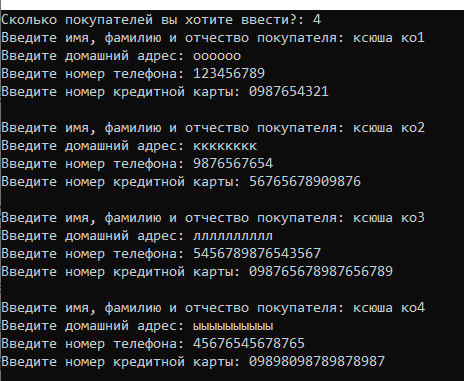
printBuyer(b);

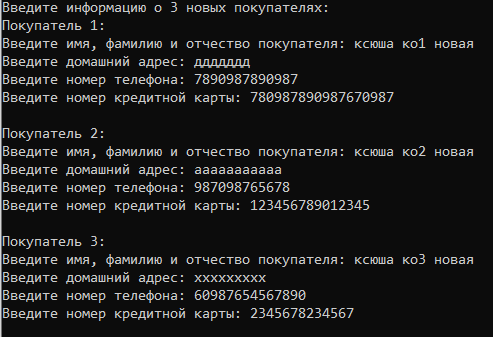
}

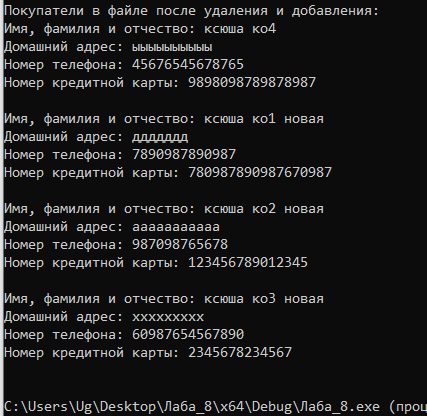
fin.close();

return 0;

}

**Вывод**

****

****

**GitHub**

<https://github.com/amalayaa>