Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №8**

Дисциплина: основы алгоритмизации и программирования

Тема: “Блоковый ввод-вывод” в C++

Вариант: 16

Выполнил работу

студент группы РИС-20-2б

Мехоношин В.А.

Проверила

Доцент кафедры ИТАС

Полякова О.А.

Пермь, 2021

**1. Цель работы**

Цель: Работа с двоичными файлами, организация ввода-вывода структурированной информации и ее хранение на внешних носителях.

**2. Постановка задачи**

16 Вар. Создать структуру "Владелец автомобиля":

-фамилия, имя, отчество;

-номер автомобиля;

-телефон;

-номер техпаспорта.

Удалить элемент с заданным номером, добавить 2 элемента

перед элементом с заданной фамилией.

**3. Анализ задачи**

1. Определить какие операции должны быть выполнены по заданию:

* Создать поле с ФИО владельца машины как массив из 3 элементов типа string
* Создать поле с номером автомобиля типа int
* Создать поле с номером телефона типа int
* Создать поле с номером техпаспорта типа int
* Создать массив, каждый элемент которого является структурой
* Произвести заполнение полей
* Удалить элемент массива с заданным номером
* Произвести поиск по фамилии
* Добавить два элемента массива после элемента с заданной фамилией

2. Какой тип данных будет у переменных с которыми эти действия будут произведены:

Необходимо объявить структуру Auto\_owner:

struct Auto\_owner {

string fio[3]; // Фио

int num; // Номер машины

int phone\_number; // Номер телефона

int number\_of\_techpass; // Номер тех. паспорта

};

И выделить память под массив из некоторого количества элементов:

const int size = 30;

Auto\_owner base[size];

3. Вывод данных будет производиться в консоль консоль через цикл for и с использованием объекта cout и оператора вывода.

void output(Auto\_owner base[], int elements) {

for (int i = 0; i < elements; i++) { // Вывод

if (base[i].num != NULL) {

cout << "\nФИО: ";

cout << base[i].fio[0] << " " << base[i].fio[1] << " " << base[i].fio[2];

cout << "\nНомер машины: ";

cout << base[i].num;

cout << "\nНомер телефона: ";

cout << base[i].phone\_number;

cout << "\nНомер тех. паспорта: ";

cout << base[i].number\_of\_techpass;

}

}

}

4. В структуре будут использоваться 4 поля:

* Поле с ФИО владельца машины как массив из 3 элементов типа string
* Поле с номером автомобиля типа int
* Поле с номером техпаспорта типа int
* Поле с номером телефона типа int

struct Auto\_owner {

string fio[3]; // Фио

int num; // Номер машины

int phone\_number; // Номер телефона

int number\_of\_techpass; // Номер тех. паспорта

};

5. Ввод данных будет производиться через цикл for с использованием объекта cin и оператора ввода.

for (int i = 0; nExT; i++) { //Ввод

safe = 999;

cout << "Введите фамилию: ";

cin >> base[i].fio[0];

cout << "Введите имя: ";

cin >> base[i].fio[1];

cout << "Введите отчество: ";

cin >> base[i].fio[2];

cout << "Введите номер машины: ";

cin >> base[i].num;

cout << "Введите номер телефона: ";

cin >> base[i].phone\_number;

cout << "Введите номер тех. паспорта: ";

cin >> base[i].number\_of\_techpass;

elements++;

while (!(safe > -1 and safe < 2)) {

cout << "\nВведите 1 чтобы продолжить и 0 чтобы прекратить: ";

cin >> safe;

if (safe == 0) {

nExT = false;

}

if (safe == 1) {

nExT = true;

}

}

}

6. Поставленные задачи необходимо решить с помощью операторных структур:

* Циклов
* Ветвлений

Удаление элементов реализовано с помощью сдвигов.

Для вставки элементов был реализован поиск по фамилии, сдвиг элементов в право на 2 и заполнение образовавшегося пустого места.

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

if (base[i].fio[0] == fam) {

for (int c = 0; c < 2; c++) { //Двойной сдвиг

for (int d = elements + c; d > i; d--) {

base[d] = base[d - 1];

}

}

for (int b = i + 1; b <= i + 2; b++) { //Ввод

cout << "Введите фамилию: ";

cin >> base[b].fio[0];

cout << "Введите имя: ";

cin >> base[b].fio[1];

cout << "Введите отчество: ";

cin >> base[b].fio[2];

cout << "Введите номер машины: ";

cin >> base[b].num;

cout << "Введите номер телефона: ";

cin >> base[b].phone\_number;

cout << "Введите номер тех. паспорта: ";

cin >> base[b].number\_of\_techpass;

elements++;

}

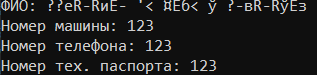
flag = true;

break;

}

}

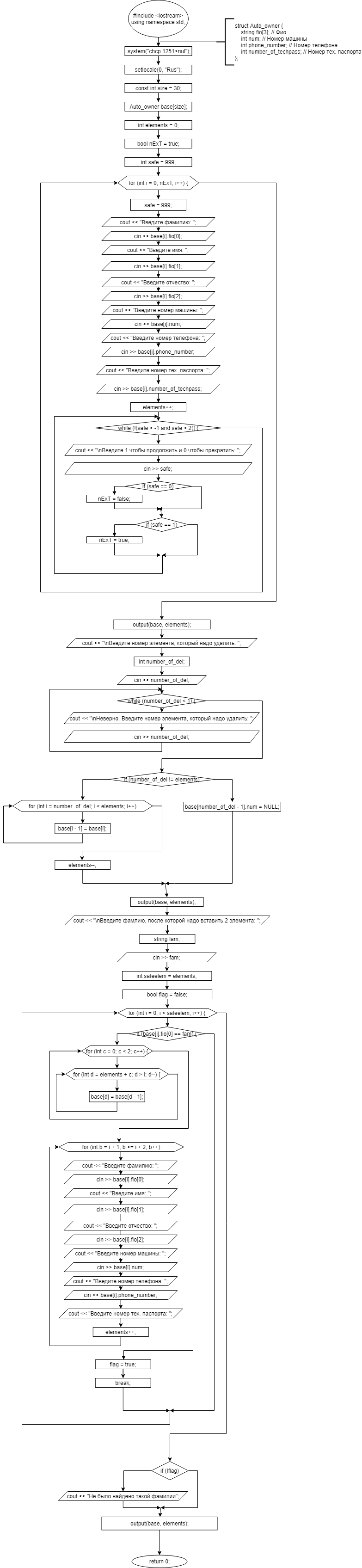
7. Проблемы возникшие при выполнении работы

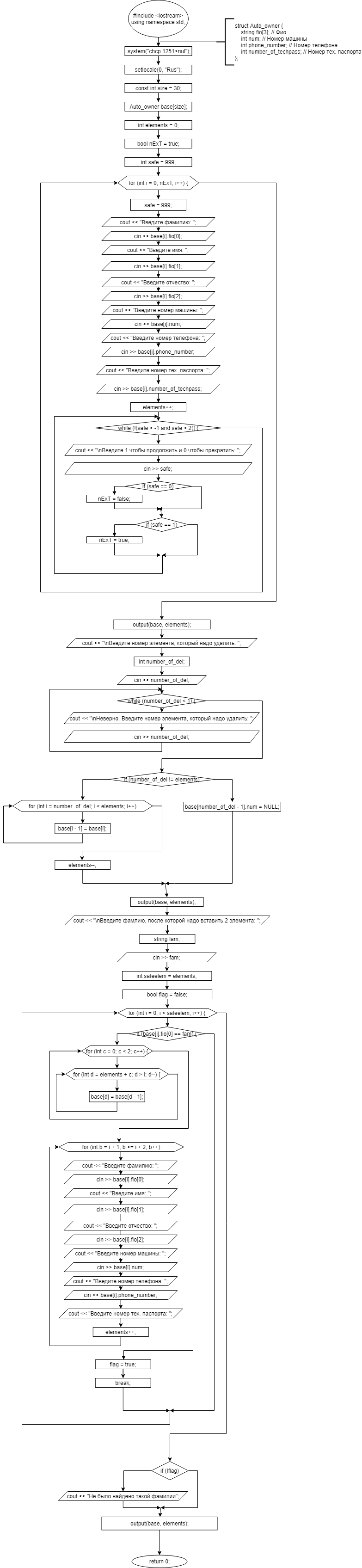
Проблемы с кодировкой при вводе кириллицы.

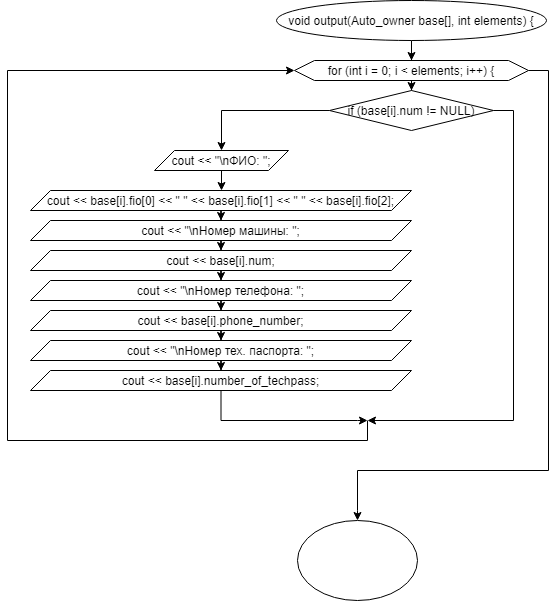
Решением было добавление строчки:

system("chcp 1251>nul");

**4. Блок-схемы программы**

****

****

****

**5. Полный код программы**

**#include <iostream>**

using namespace std;

struct Auto\_owner {

string fio[3]; // Фио

int num; // Номер машины

int phone\_number; // Номер телефона

int number\_of\_techpass; // Номер тех. паспорта

};

void output(Auto\_owner base[], int elements) {

for (int i = 0; i < elements; i++) { // Вывод

if (base[i].num != NULL) {

cout << "\nФИО: ";

cout << base[i].fio[0] << " " << base[i].fio[1] << " " << base[i].fio[2];

cout << "\nНомер машины: ";

cout << base[i].num;

cout << "\nНомер телефона: ";

cout << base[i].phone\_number;

cout << "\nНомер тех. паспорта: ";

cout << base[i].number\_of\_techpass;

}

}

}

int main()

{

system("chcp 1251>nul");

setlocale(0, "Rus");

const int size = 30;

Auto\_owner base[size];

int elements = 0;

bool nExT = true;

int safe = 999;

for (int i = 0; nExT; i++) { //Ввод

safe = 999;

cout << "Введите фамилию: ";

cin >> base[i].fio[0];

cout << "Введите имя: ";

cin >> base[i].fio[1];

cout << "Введите отчество: ";

cin >> base[i].fio[2];

cout << "Введите номер машины: ";

cin >> base[i].num;

cout << "Введите номер телефона: ";

cin >> base[i].phone\_number;

cout << "Введите номер тех. паспорта: ";

cin >> base[i].number\_of\_techpass;

elements++;

while (!(safe > -1 and safe < 2)) {

cout << "\nВведите 1 чтобы продолжить и 0 чтобы прекратить: ";

cin >> safe;

if (safe == 0) {

nExT = false;

}

if (safe == 1) {

nExT = true;

}

}

}

output(base, elements);

cout << "\nВведите номер элемента, который надо удалить: ";

int number\_of\_del;

cin >> number\_of\_del; // C единицы

while (number\_of\_del < 1) {

cout << "\nНеверно. Введите номер элемента, который надо удалить: ";

cin >> number\_of\_del;

}

if (number\_of\_del != elements) {

for (int i = number\_of\_del; i < elements; i++) {

base[i - 1] = base[i];

}

elements--;

}

else{

base[number\_of\_del - 1].num = NULL;

}

output(base, elements);

cout << "\nВведите фамлию, после которой надо вставить 2 элемента: ";

string fam;

cin >> fam;

int safeelem = elements;

bool flag = false;

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

if (base[i].fio[0] == fam) {

for (int c = 0; c < 2; c++) { //Двойной сдвиг

for (int d = elements + c; d > i; d--) {

base[d] = base[d - 1];

}

}

for (int b = i + 1; b <= i + 2; b++) { //Ввод

cout << "Введите фамилию: ";

cin >> base[b].fio[0];

cout << "Введите имя: ";

cin >> base[b].fio[1];

cout << "Введите отчество: ";

cin >> base[b].fio[2];

cout << "Введите номер машины: ";

cin >> base[b].num;

cout << "Введите номер телефона: ";

cin >> base[b].phone\_number;

cout << "Введите номер тех. паспорта: ";

cin >> base[b].number\_of\_techpass;

elements++;

}

flag = true;

break;

}

}

if (!flag) {

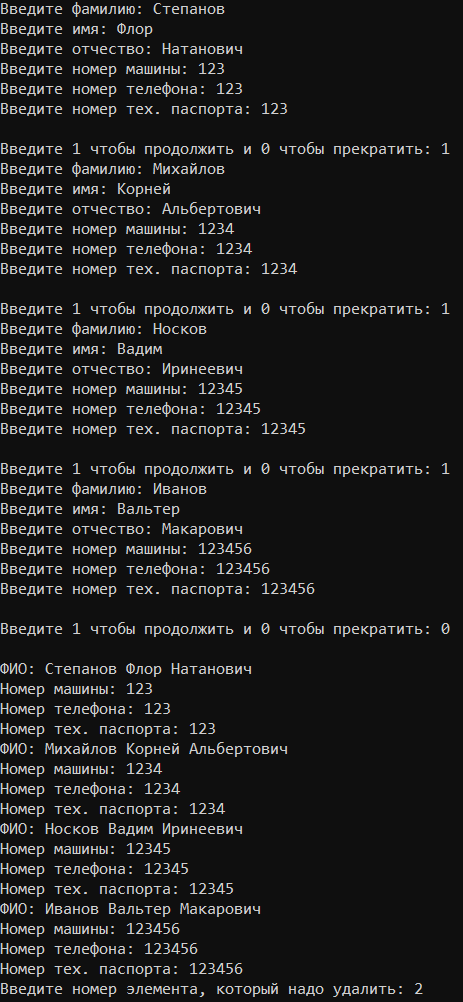
cout << "Не было найдено такой фамилии";

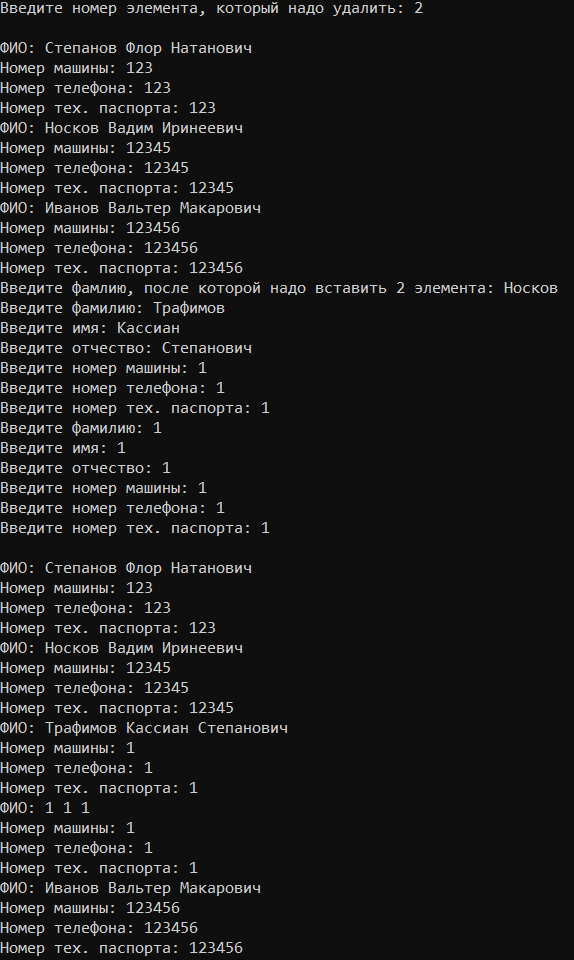
}

output(base, elements);

**}**

**6. Скриншоты**

****

****