Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №8**

Дисциплина: «Информатика»

Тема: “Структуры данных”

Вариант 16

Выполнил:

Студент группы ИВТ-20-2Б Тедеев Александр Зурабович

Проверила:

Доцент кафедры ИТАС

Полякова О.А.

Пермь, 2021

**Цель задачи**

Организация ввода-вывода структурированной информации.

**Постановка задачи**

Сформировать структуру «Владелец автомобиля» с полями: фамилия, имя, отчество; номер автомобиля; телефон; номер техпаспорта. Удалить элемент с заданным номером, добавить 2 элемента перед элементом с заданной фамилией.

**Анализ задачи**

1. Объявлен динамический массив owner типа CarOwner для хранения информации о каждом введённом авто водителе; массив типа CarOwner на два элемента для последующего добавления двух авто водителей; переменная типа string imya для ввода фамилии, перед которой нужно добавить двух новый авто водителей; переменная nomer типа string для ввода номера машины, владельца которой нужно убрать из массива структур; переменная kol, которая отвечает за размер массива owner для выбора пользователем количества авто водителей в структуре.

int kol=-1;

while (kol < 1 || kol>6)

{

cout << "Количество человек от 1 до 5: ";

cin >> kol;

}

cout << endl;

CarOwner\* owner = new CarOwner[kol];

CarOwner arr[2];

string nomer, imya;

2. Объявлена структура «Владелец автомобиля» (CarOwner) c 4 полями:

* Поле name типа string для ввода фамилии.
* Поле num типа string для ввода имени и отчества.
* Поле tel типа int для ввода номера телефона.
* Поле passp типа int для ввода номера техпаспорта.

struct CarOwner

{

public:

string name, nam, num;

int tel, passp;

...

};

3. Ввод фамилии, телефона, техпаспорта обеспечивается командой cin, так как использование функции getline() для ввода, например, фамилии вызывает ошибки в консоли. Функцию getline() использовать для ввода имени и отчества, так как они разделяются пробелом, необходимо.

void Input()

{

cout << "Введите Фамилию: ";

cin >> name;

while (cin.get() != '\n'); //исправление ошибки с консолью

cout << "Имя и отчество ";

getline(cin, nam);

cout << "\nВведите номер автомобиля: ";

getline(cin, num);

cout << "Введите номер телефона: ";

cin >> tel;

cout << "Введите номер техпаспорта: ";

cin >> passp;

cout << endl;

}

void Output()

{

cout << "\nФамилия: "<< name

<< "\n Имя, отчество "<< nam

<< "\nНомер автомобиля: "<< num

<< "\nТелефон: "<< tel

<< "\nТехпаспорт: "<< passp<< endl;

}

4. Чтобы осуществить удаление элемента (авто водителя) из структуры, организовать функцию Deletion типа int, которая будет возвращать kol (размерность массива, уменьшенную на 1).

int Deletion(int kol, CarOwner\* owner, string nomer)

{

int i=0,tmp;

bool flag = false;

while ((!flag)&&(i<=kol))

{

if (nomer == owner[i].num)

{

flag = true;

for (int j = i; (j < kol - 1); j++)

{

owner[j] = owner[j+1];

}

kol--;

}

i++;

}

if (flag == false)

cout << "Такой фамилии не нашлось!" << endl;

return kol;

}

Вводится флажок, который необходим для остановки цикла, необходимого для проверки наличия номера машины и для запоминания номера этого элемента в массиве. Если пользователь не ошибся в написании фамилии, то происходит удаление того элемента и сдвиг всех остальных элементов.

Для того, чтобы осуществить добавление двух элементов (авто водителей) в структуру необходимо организовать функцию Addition типа int. В данной функции организуется цикл для сравнения введённой пользователем фамилии со фамилиями из массива структур, который передается как параметр в функцию Addition, где в переменную positon сохраняется номер фамилии в массиве структур, если находится совпадение. Флажок – признак выхода из цикла.

bool flag = false;

int tmp=0, position,count=0;

while ((!flag) && (tmp <= kol))

{

if (imya == owner[tmp].name)

{

flag = true;

position = tmp;

}

tmp++;

}

Для вставки необходимых элементов в массив структур, организуется вспомогательный массив в функции InputNew и в самой функции Addition vector, который получает элементы из поступившего в функцию массива и введённых пользователем из функции InputNew. С помощью метода insert класса vector производим вставку в массив.

vector<CarOwner>massiv(owner, owner+kol);

massiv.reserve(kol+2);

kol += 2;

for (int i = 0; i < 2; i++)

{

massiv.insert(massiv.begin() + position, 1, InputNew(arr));

}

6. Выводим изменённый список через цикл.

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

{

massiv[i].Output();

}

**Код**

#include <iostream>

#include <string>

#include <stdlib.h>

#include <vector>

using namespace std;

struct CarOwner

{

public:

string name, nam, num;

int tel, passp;

void Input()

{

cout << "Введите Фамилию: ";

cin >> name;

while (cin.get() != '\n'); //исправление ошибки с консолью

cout << "Имя и отчество ";

getline(cin, nam);

cout << "\nВведите номер автомобиля: ";

getline(cin, num);

cout << "Введите номер телефона: ";

cin >> tel;

cout << "Введите номер техпаспорта: ";

cin >> passp;

cout << endl;

}

void Output()

{

cout << "\nФамилия: "<< name

<< "\n Имя, отчество "<< nam

<< "\nНомер автомобиля: "<< num

<< "\nТелефон: "<< tel

<< "\nТехпаспорт: "<< passp<< endl;

}

};

void InputList(int kol, CarOwner \*owner);

CarOwner InputNew(CarOwner\* arr);

int Deletion(int kol, CarOwner\* owner, string nomer);

void Printing(int kol, CarOwner\* owner);

int Addition(int kol, CarOwner\* owner, string imya, CarOwner \*arr);

int main()

{

setlocale(LC\_ALL, "Ru");

system("chcp 1251>nul");

int kol=-1;

while (kol < 1 || kol>6)

{

cout << "Количество человек от 1 до 5: ";

cin >> kol;

}

cout << endl;

CarOwner\* owner = new CarOwner[kol];

CarOwner arr[2];

InputList(kol, owner);

cout << "Введите номер элемента, который необходимо удалить: "<< endl;

string nomer, imya;

while (cin.get() != '\n');

getline(cin, nomer);

kol=Deletion(kol, owner, nomer);

cout << "------------------------------------";

Printing(kol, owner);

cout << "------------------------------------";

cout << "\nПеред кем добавить 2 элемента? (введите фамилию): ";

getline(cin, imya);

kol = Addition(kol, owner, imya, arr);

cout << "------------------------------------";

delete[] owner;

return 0;

}

void InputList(int kol, CarOwner\* owner)

{

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

{

owner[i].Input();

}

}

int Deletion(int kol, CarOwner\* owner, string nomer)

{

int i=0,tmp;

bool flag = false;

while ((!flag)&&(i<=kol))

{

if (nomer == owner[i].num)

{

flag = true;

for (int j = i; (j < kol - 1); j++)

{

owner[j] = owner[j+1];

}

kol--;

}

i++;

}

if (flag == false)

cout << "Такой номера не нашлось!" << endl;

return kol;

}

void Printing(int kol, CarOwner\* owner)

{

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

{

owner[i].Output();

}

}

int Addition(int kol, CarOwner\* owner, string imya, CarOwner\*arr)

{

bool flag = false;

int tmp=0, position,count=0;

while ((!flag) && (tmp <= kol))

{

if (imya == owner[tmp].name)

{

flag = true;

position = tmp;

}

tmp++;

}

if (flag)

{

vector<CarOwner>massiv(owner, owner+kol);

massiv.reserve(kol+2);

kol += 2;

for (int i = 0; i < 2; i++)

{

massiv.insert(massiv.begin() + position, 1, InputNew(arr));

}

cout << "------------------------------------";

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

{

massiv[i].Output();

}

}

else

cout << "Такой фамилии нет! " << endl;

return kol;

}

CarOwner InputNew(CarOwner \*arr)

{

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

{

arr[i].Input();

}

return \*arr;

}

**Скриншоты**

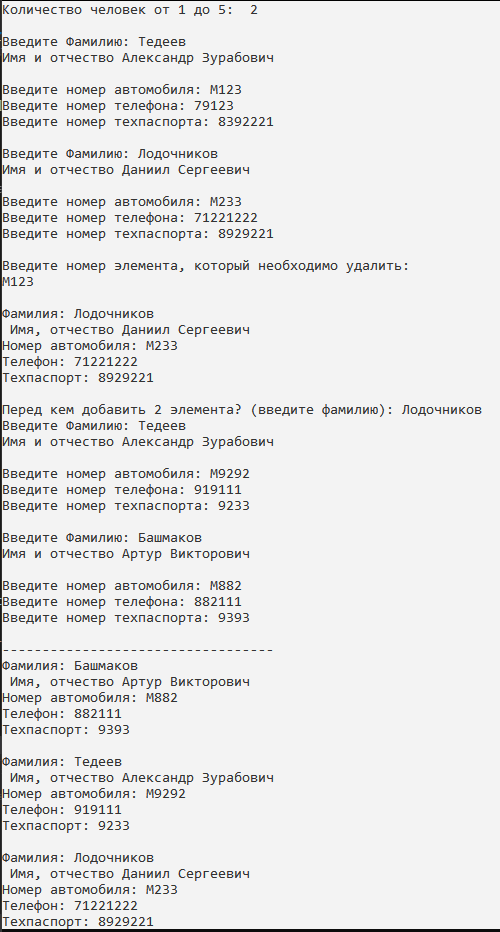


Рис. 1

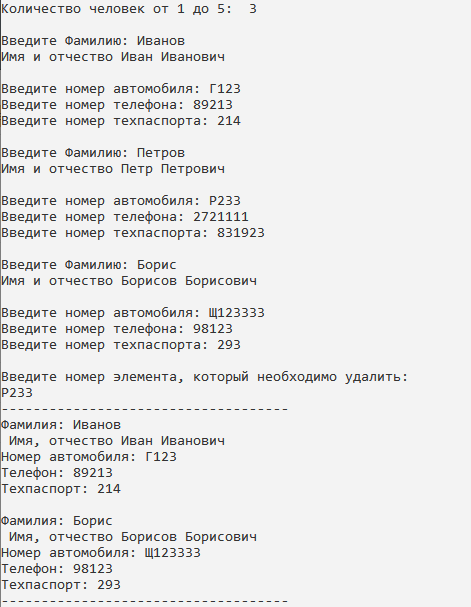


Рис. 2

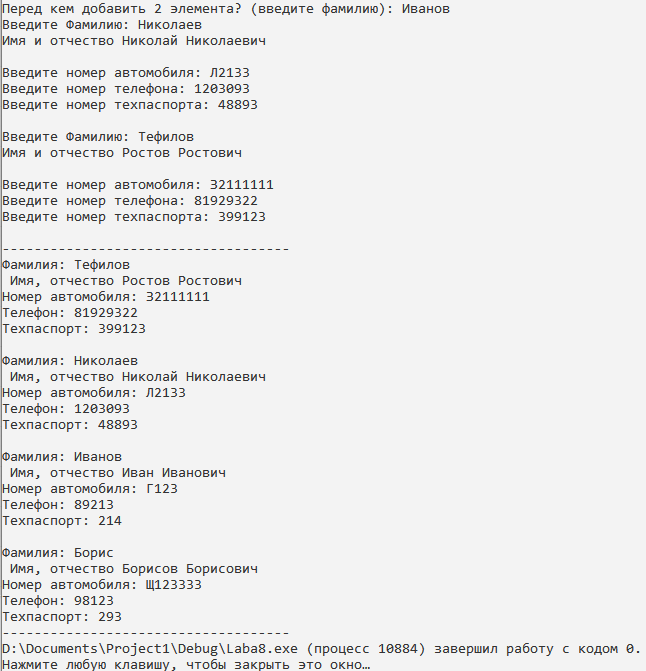
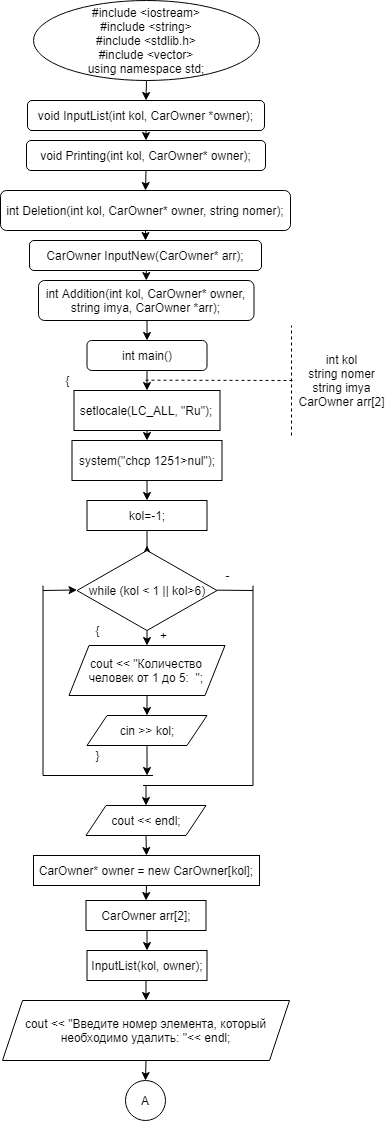
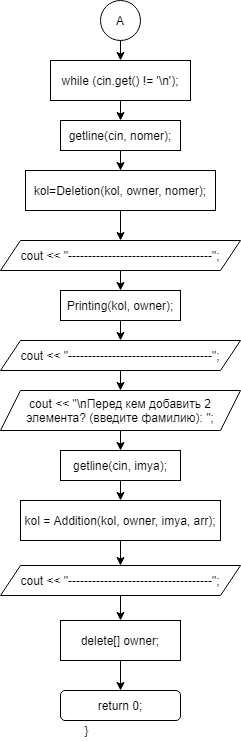


Рис. 3

**Блок-схема**

****

****

