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: “Структуры данных”

**Вар.21**

Выполнил работу:

студент группы ИВТ-20-2Б

Галинов О.Ю.

Проверила:

Доцент кафедры ИТАС

Полякова О.А.

Пермь, 2020

**Цель задачи**

Цель – организация ввода-вывода структурированной информации.

Для достижения поставленной цели необходимо решить следующие задачи:

* Описать структуру с заданными полями.
* Организовать ввод-вывод данных в массив (из массива).
* Реализовать добавление и удаление элементов через функцию.
* Разработать программу.

**Постановка задачи**

Сформировать структуру Автомобиль с полями: марка, серийный номер, регистрационный номер, год выпуска. Удалить 3 первых элемента, добавить элемент после элемента с указанным регистрационным номером.

**Анализ задачи**

**1.** Для решения задачи используются типы данных **string, int**.

* *int regnum;*
* *int sernum;*
* *int year;*
* *string marka;*

**2.** Для хранения объектов структуры используется **vector**, для этого подключается соответствующая библиотека. Длина вектора вводится пользователем (переменная **size**).

* *#include <vector>*
* *vector <Auto> myvec(size);*

Так же описываются 2 функции типа **void**: **fill**, **print**. В функции ничего не передается, так как мы вызываем их в цикле (по кол-ву введенного значения) и они просто позволяют вводить данные для структуры.

Длинна массива не может быть меньше нуля, поэтому выполняется проверка на ввод с помощью цикла while.

**3.** Для заполнения полей структуры **Auto** используется метод **fill**, который заполняет поля структуры последовательно. Для вывода значений полей структуры используется метод **print**, который работает аналогично.

*void fill()*

*{*

*cout << "Введите год выпуска:\t";*

*cin >> year;*

*cout << "Введите регистрационный номер:\t";*

*cin >> regnum;*

*cout << "Введите серийный номер:\t";*

*cin >> sernum;*

*cout << "Введите марку машины:\t";*

*cin.ignore();*

*getline(cin, marka);*

*cout << "======================" << endl << endl;*

*}*

*void print()*

*{*

*cout << "\n Год выпуска: \t\t" << year;*

*cout << "\n Серийный номер: \t" << sernum;*

*cout << "\n Регистрационный номер:\t" << regnum;*

*cout << "\n Марка: \t\t" << marka << endl;;*

*}*

**4.** Описывается структура с 4 полями (*regnum*, *sernum*, *year*, *marka):*

*struct Auto*

*{*

*int year;*

*string marka;*

*int regnum;*

*int sernum;*

*…*

*};*

**5.** В функциях ввода-вывода данных используются операторы **cin**, **cout**.(см.п.3).

*int size;*

*cout << "Введите кол-во машин:\t";*

*cin >> size;void ShowArray(vector <State> &arr)*

*void fill()*

*{*

*cout << "Введите год выпуска:\t";*

*cin >> year;*

*…*

*}*

*void print()*

*{*

*cout << "\n Год выпуска: \t\t" << year;*

*cout << "\n Серийный номер: \t" << sernum;*

*…*

*}*

**6.**  Для функций ввода-вывода использовался цикл с автоматическим перебором переменных (см.п.5).

Чтобы удалить три первых элемента применяется метод класса **vector** (.erase).

*myvec.erase(myvec.begin(), myvec.begin() + 3); // 3 elements*

Для прохода по массиву используется цикл for. На каждой итерации цикла сравнивается поле **regnum** i-го элемента массива с введенным значением переменной **number**. В случае равенства после того элемента вводится еще одно поле структуры.

*for (int i = 0; i < myvec.size(); i++)*

*{*

*if (myvec[i].regnum == number)*

*{*

*myvec.insert(myvec.begin() + i + 1, Auto());*

*myvec[i + 1].fill();*

*}*

После цикла снова выводится структура уже с новым полем.

*for (int i = 0; i < myvec.size(); i++)*

*{*

*myvec[i].print();*

*}*

**Код**

#include <iostream>

#include <string>

#include <vector>

using namespace std;

struct Auto

{

int year;

string marka;

int regnum;

int sernum;

void fill()

{

cout << "Введите год выпуска:\t";

cin >> year;

cout << "Введите регистрационный номер:\t";

cin >> regnum;

cout << "Введите серийный номер:\t";

cin >> sernum;

cout << "Введите марку машины:\t";

cin.ignore();

getline(cin, marka);

cout << "======================" << endl << endl;;

}

void print()

{

cout << "\n Год выпуска: \t\t" << year;

cout << "\n Серийный номер: \t" << sernum;

cout << "\n Регистрационный номер:\t" << regnum;

cout << "\n Марка: \t\t" << marka << endl;;

}

};

int main()

}

system("chcp 1251 > null");

setlocale(LC\_ALL, "rus");

vector <Auto> myvec;

int size;

cout << "Введите кол-во машин:\t";

cin >> size;

while (size < 4)

{

cout << "Введите кол-во корректно:\t";

cin >> size;

}

cout << endl;

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

{

myvec.push\_back(Auto());

myvec[i].fill();

}

cout << "Ваша структура:" << endl;

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

{

myvec[i].print();

} cout << "======================" << endl;

cout << "Структура без первых 3 элементов: " << endl; //remove first

myvec.erase(myvec.begin(), myvec.begin() + 3); // 3 elements

for (int i = 0; i < myvec.size(); i++)

{

myvec[i].print();

}

int number;

cout << "Введите какой-нибудь регистрационный номер:\t";

cin >> number;

for (int i = 0; i < myvec.size(); i++)

{

if (myvec[i].regnum == number)

{

myvec.insert(myvec.begin() + i + 1, Auto());

myvec[i + 1].fill();

}

}

cout << "======================" << endl;

cout << "Структура с добавленным элементом после введенного регистрационного номера: " << endl;

for (int i = 0; i < myvec.size(); i++)

{

myvec[i].print();

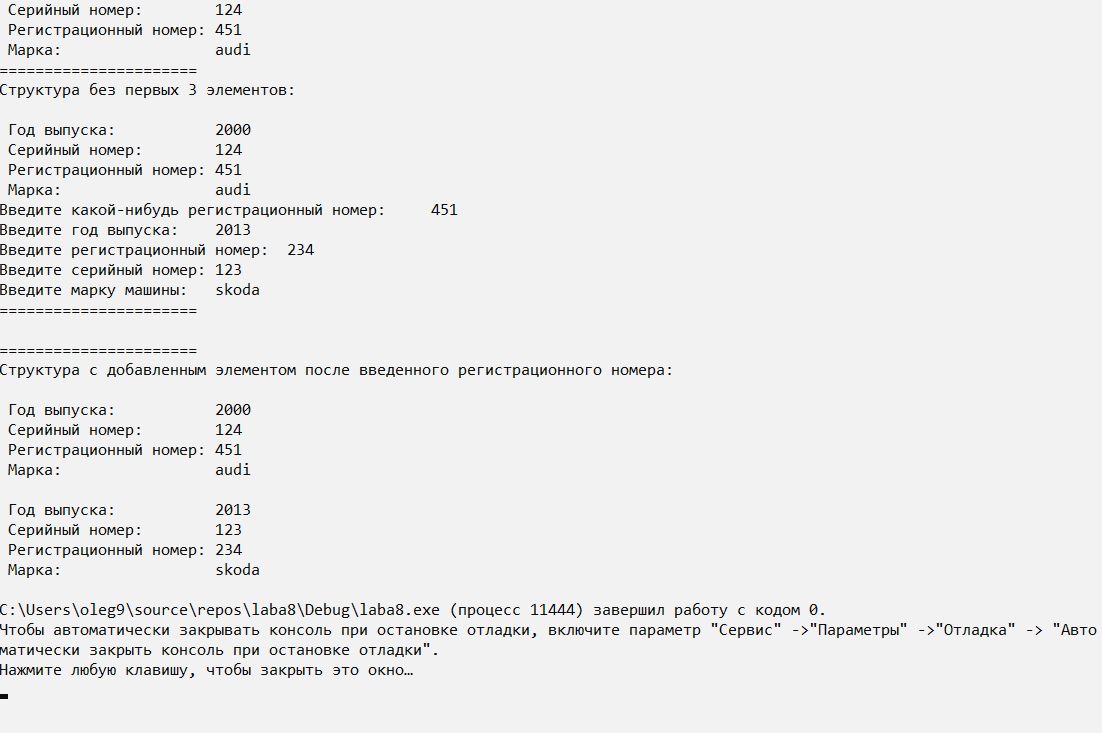
}

return 0;

}

**Работа кода**





**Блок-схема**

