|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |   Институт Информационных технологий | |
|  | |
| Кафедра Математического обеспечения и стандартизации информационных технологий | |
|  | |
|  | |

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2** | |
| **по дисциплине** | |
| **«**Процедурное программирование**»**  **Тема: «МОДУЛЬНОЕ ПРОГРАММИРОВАНИЕ. ПОДПРОГРАММЫ»** | |
|  | |
| Выполнил студент группы ИКБО-12-18 | Валяев Д.А. |
| Принял преподаватель | Миронов А.Н. |

|  |  |  |
| --- | --- | --- |
| Лабораторная работа выполнена | «25» Марта 2019 г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_201\_\_ г. | *(подпись руководителя)* |

Москва 2018

Задание 1. Работа со статическим одномерным массивом.

1. Условие задачи

Создать, отсортировать структуру, выполнить дополнительную операцию

2. Постановка задачи

Дано: дана структура из N элеметов.

Ограничения на данные: нет

Ограничения на решаемую задачу: нет

3. Математическая модель задачи

-Student – структура

-вводится количество элементов

-вводятся элементы структуры

-введенные элементы сортируются

-выполняется дополнительная операция

4. Определение входных и выходных данных

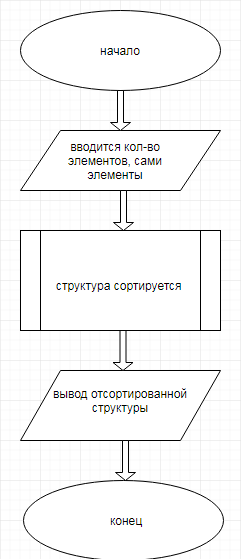
Входные данные- количество элементов ,сами элементы, вводится с клавиатуры.

Результат – отсортированная структура

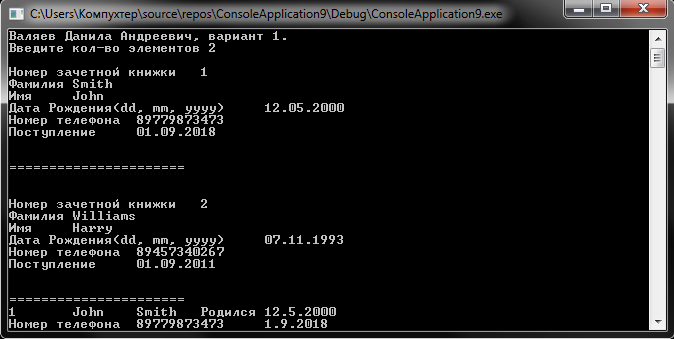
5. Таблица имен объектов программы

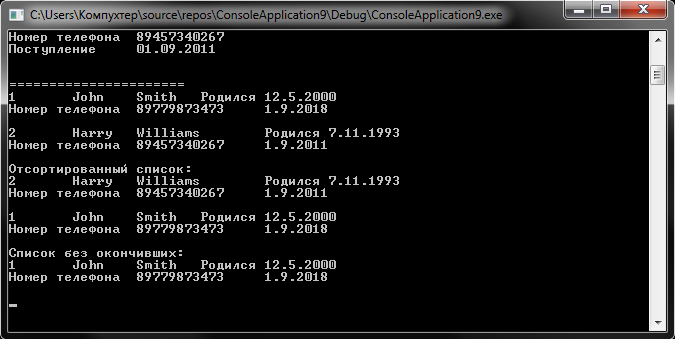
|  |  |  |  |
| --- | --- | --- | --- |
| *Имя объекта* | *Диапазон допустимых значений* | *Тип данных* | *Семантика* |
| N,I, date, id, postup | -2 147 483 648 .. 2 147 483 647 | int | Переменная, аргумент функции |
| first\_name, last\_name, phone, | 35 | char | Переменная, аргумент функции |

6. Разработка алгоритма программы



7. Разработка тестов





8. Исходный код программы

//Валяев Данила Андреевич, вариант 1.

#include "pch.h"

#include <iostream>

#include <windows.h>

#include <cstring>

#include <time.h>

using namespace std;

int count() {

int n;

cout << "Введите кол-во элементов ";

cin>>n;

return n;

}

struct Student

{

int id[1]; //Номер зачетку

char first\_name[35]; //Имя

char last\_name[35]; //Фамилия

char phone[35]; //Номер телефона

int date[3], postup[3]; //Дата pождения и поступления в учебное заведение

void show(); //вывод

void input(); //ввод

};

void Student::input()

{

cout << "\n\n";

cout << "Номер зачетной книжки\t";

cin >> id[0];

cin.ignore();

cout << "Фамилия\t";

cin.getline(last\_name, 35);

cout << "Имя\t";

cin.getline(first\_name, 35);

cout << "Дата Рождения(dd, mm, yyyy)\t";

cin >> date[0];

cin.ignore();

cin >> date[1];

cin.ignore();

cin >> date[2];

cin.ignore();

cout << "Номер телефона\t";

cin.getline(phone, 35);

cout << "Поступление\t";

cin >> postup[0];

cin.ignore();

cin >> postup[1];

cin.ignore();

cin >> postup[2];

cin.ignore();

cout << "\n\n";

cout << "======================\n";

}

void Student::show()

{

cout << id[0] <<"\t" << first\_name << "\t" << last\_name << "\tРодился " << date[0] << "." << date[1] << "." << date[2] << "\n";

cout << "Номер телефона\t" << phone << "\t" << postup[0] << "." << postup[1] << "." << postup[2] << "\n\n";

}

int main()

{

setlocale(LC\_ALL, "ru");

cout << "Валяев Данила Андреевич, вариант 1.";

const int N = count();

Student Student[N];

for (int i = 0; i < N; i++) Student[i].input();

for (int i = 0; i < N; i++) Student[i].show();

for (int i = 1; i < N; i++) {

for (int j = 0; j < N - 1; j++) {

if (Student[i].date[2] < Student[j].date[2]) {

swap(Student[i], Student[j]);

}

else if (Student[i].date[2] == Student[j].date[2]) {

if (Student[i].date[1] < Student[j].date[1])

swap(Student[i], Student[j]);

else if (Student[i].date[1] == Student[j].date[1]) {

if (Student[i].date[0] < Student[j].date[0])

swap(Student[i], Student[j]);

}

}

}

}

cout << "Отсортированный список студентов:\n";

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

Student[i].show();

}

int times;

time\_t t;

tm \*tk;

time(&t);

tk = localtime(&t);

times = 1900 + tk->tm\_year;

cout << "Список студентов без окончивших институт:\n";

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

if (times - Student[i].postup[2] <= 6) {

Student[i].show();

}

}

cin.get();

}

Задание 2. Работа с динамическим одномерным массивом.

1. Условие задачи

Требуется из одномодульного приложения, разработанного в задании 1, создать многомодульное приложение.

2. Математическая модель задачи

-Student – структура

-вводится количество элементов

-вводятся элементы структуры

-введенные элементы сортируются

-выполняется дополнительная операция

4. Определение входных и выходных данных

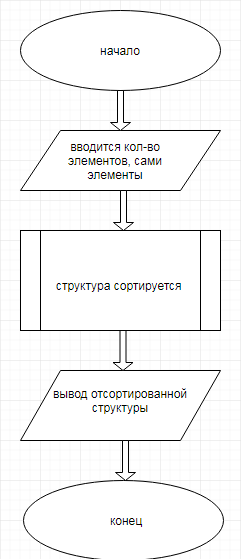
Входные данные- количество элементов ,сами элементы, вводится с клавиатуры.

Результат – отсортированная структура

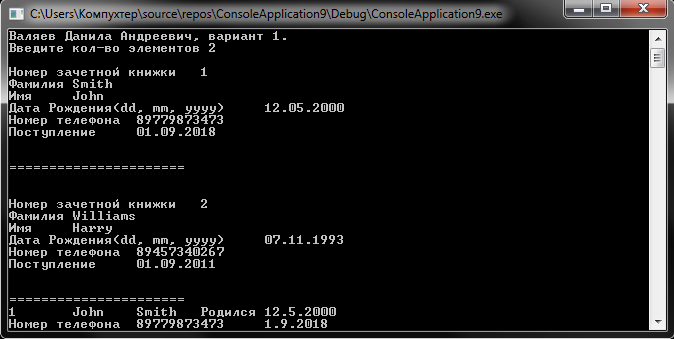
5. Таблица имен объектов программы

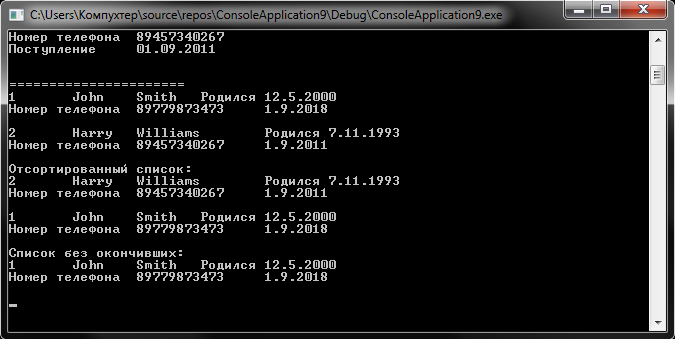
|  |  |  |  |
| --- | --- | --- | --- |
| *Имя объекта* | *Диапазон допустимых значений* | *Тип данных* | *Семантика* |
| N,I, date, id, postup | -2 147 483 648 .. 2 147 483 647 | int | Переменная, аргумент функции |
| first\_name, last\_name, phone, | 35 | char | Переменная, аргумент функции |

6. Разработка алгоритма программы



7. Разработка тестов





8. Исходный код программы

//Валяев Данила Андреевич, вариант 1.

#include "pch.h"

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <windows.h>

#include <cstring>

#include <time.h>

#include " Unit\_Def "

#include " Unit\_Input\_Output:"

#include " Unit\_Operation:

using namespace std;

void sort() {

}

int main()

{

setlocale(LC\_ALL, "ru");

cout << "Валяев Данила Андреевич, вариант 1.";

const int N = count();

Student Student[N];

for (int i = 0; i < N; i++) Student[i].input();

for (int i = 0; i < N; i++) Student[i].show();

for (int i = 1; i < N; i++) {

for (int j = 0; j < N - 1; j++) {

if (Student[i].date[2] < Student[j].date[2]) {

swap(Student[i], Student[j]);

}

else if (Student[i].date[2] == Student[j].date[2]) {

if (Student[i].date[1] < Student[j].date[1])

swap(Student[i], Student[j]);

else if (Student[i].date[1] == Student[j].date[1]) {

if (Student[i].date[0] < Student[j].date[0])

swap(Student[i], Student[j]);

}

}

}

}

cout << "Отсортированный список студентов:\n";

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

Student[i].show();

}

int times;

time\_t t;

tm \*tk;

time(&t);

tk = localtime(&t);

times = 1900 + tk->tm\_year;

cout << "Список студентов без окончивших институт:\n";

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

if (times - Student[i].postup[2] <= 6) {

Student[i].show();

}

}

cin.get();

}

Unit\_Def:

#pragma once

struct Student

{

int id[1]; //Номер зачетку

char first\_name[35]; //Имя

char last\_name[35]; //Фамилия

char phone[35]; //Номер телефона

int date[3], postup[3]; //Дата pождения и поступления в учебное заведение

void show(); //вывод

void input(); //ввод

void sort();

};

Unit\_Input\_Output:

#pragma once

#include " Unit\_Def "

using namespace std;

int count() {

int n;

cout << "Введите кол-во элементов ";

cin >> n;

return n;

}

void Student::show()

{

cout << id[0] << "\t" << first\_name << "\t" << last\_name << "\tРодился " << date[0] << "." << date[1] << "." << date[2] << "\n";

cout << "Номер телефона\t" << phone << "\t" << postup[0] << "." << postup[1] << "." << postup[2] << "\n\n";

}

Unit\_Operation:

#pragma once

#include " Unit\_Def "

using namespace std;

void Student::input()

{

cout << "\n\n";

cout << "Номер зачетной книжки\t";

cin >> id[0];

cin.ignore();

cout << "Фамилия\t";

cin.getline(last\_name, 35);

cout << "Имя\t";

cin.getline(first\_name, 35);

cout << "Дата Рождения(dd, mm, yyyy)\t";

cin >> date[0];

cin.ignore();

cin >> date[1];

cin.ignore();

cin >> date[2];

cin.ignore();

cout << "Номер телефона\t";

cin.getline(phone, 35);

cout << "Поступление\t";

cin >> postup[0];

cin.ignore();

cin >> postup[1];

cin.ignore();

cin >> postup[2];

cin.ignore();

cout << "\n\n";

cout << "======================\n";

}