**Отчет**

**Групповая лабораторная работа. Тема №3**

Хачатурян В.В.

ИИС ПИ 2-1

Исходный код на гитхабе:

Справочник крупнейших транспортных средств 3 категорий: машины, корабли и самолёты. (машины: длина в метрах, высота в метрах, год постройки, страна) (корабли: длина в метрах, осадка, название, страна постройки) (самолеты: размах крыльев, длина, максимальная скорость, максимальная высота полёта). Меню просмотра списка, отдельный вид для каждого объекта. Возможность добавлять и удалять объекты из каждого списка. Поиск по названию (по подстроке) в каждой категории.

// Тема 3

// Участники разработки групповой работы: Бибиков И., Мильдзихов С., Мильдзихов А., Понкратов К., Арвачев Н., Краюшкин Е., Хачатурян В.

// Весь код снабжен комментариями для более простого понимания структуры

#include <iostream>

#include <string>

using namespace std;

struct car { // описание структуры автомобиля

double lg; // длина

double hg; // высота

int year; // год выпуска

string country; // страна

string name; // название

};

void printItemCar(car\* arr, int n) { // вывод информации о массиве авто

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

cout << "Авто №" << i + 1 << endl;

cout << "Название: " << arr[i].name << endl;

cout << "Длина: " << arr[i].lg << endl;

cout << "Высота: " << arr[i].hg << endl;

cout << "Год постройки: " << arr[i].year << endl;

cout << "Страна: " << arr[i].country << endl;

cout << endl;

}

}

void searchItemCar(car\* arr, int n, string nm) { // поиск авто в массиве по имени

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

if (arr[i].name == nm) {

printItemCar(&arr[i], 1);

return;

}

}

cout << "Такое авто не найдено!" << endl;

return;

}

void removeItemCar(car\*& arr, int\* n) { // удаление авто из массива по имени путем создания нового массива, удаления старого и замены

string nm;

cout << "Введите название авто для удаления: ";

cin >> nm;

bool ifsearch = false; // если найдена машина

for (int i = 0; i < (\*n); i++) {

if (arr[i].name == nm)

{

arr[i].name = "NULL";

ifsearch = true;

}

};

if (!ifsearch) {

cout << "Такой элемент не найден!" << endl;

return;

}

car\* arr2 = new car[(\*n) - 1];

int j = 0;

for (int i = 0; i < (\*n) - 1; i++) {

if (arr[j].name == "NULL")

j++;

arr2[i] = arr[j];

j++;

}

(\*n)--;

delete[] arr;

arr = arr2;

return;

}

void addItemCar(car\*& arr, int\* n) { // добавление автомобиля в массив структур, суть работы аналогична удалению

car\* arr2 = new car[(\*n) + 1];

for (int i = 0; i < (\*n); i++) {

arr2[i] = arr[i];

}

car item;

cout << "Введите название: ";

cin >> item.name;

cout << "Введите длину: ";

cin >> item.lg;

cout << "Введите высоту: ";

cin >> item.hg;

cout << "Введите год постройки: ";

cin >> item.year;

cout << "Введите страну: ";

cin >> item.country;

(\*n)++;

arr2[(\*n) - 1] = item;

delete[] arr;

arr = arr2;

}

struct ship { // описание структуры корабля

double lg; // длина

double os; // осадка

string country; // страна

string name; // название

};

void printItemShip(ship\* arr, int n) { // функция полностью аналогично одноименной для авто

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

cout << "Корабль №" << i + 1 << endl;

cout << "Название: " << arr[i].name << endl;

cout << "Длина: " << arr[i].lg << endl;

cout << "Осадка: " << arr[i].os << endl;

cout << "Страна: " << arr[i].country << endl;

cout << endl;

}

}

void searchItemShip(ship\* arr, int n, string nm) { // функция полностью аналогично одноименной для авто

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

if (arr[i].name == nm) {

printItemShip(&arr[i], 1);

return;

}

}

cout << "Такой корабль не найден!" << endl;

return;

}

void removeItemShip(ship\*& arr, int\* n) { // функция полностью аналогично одноименной для авто

string nm;

cout << "Введите название корабля для удаления: ";

cin >> nm;

bool ifsearch = false; // если найден

for (int i = 0; i < (\*n); i++) {

if (arr[i].name == nm)

{

arr[i].name = "NULL";

ifsearch = true;

}

};

if (!ifsearch) {

cout << "Такой элемент не найден!" << endl;

return;

}

ship\* arr2 = new ship[(\*n) - 1];

int j = 0;

for (int i = 0; i < (\*n) - 1; i++) {

if (arr[j].name == "NULL")

j++;

arr2[i] = arr[j];

j++;

}

(\*n)--;

delete[] arr;

arr = arr2;

return;

}

void addItemShip(ship\*& arr, int\* n) { // функция полностью аналогично одноименной для авто

ship\* arr2 = new ship[(\*n) + 1];

for (int i = 0; i < (\*n); i++) {

arr2[i] = arr[i];

}

ship item;

cout << "Введите название: ";

cin >> item.name;

cout << "Введите длину: ";

cin >> item.lg;

cout << "Введите осадку: ";

cin >> item.os;

cout << "Введите страну: ";

cin >> item.country;

(\*n)++;

arr2[(\*n) - 1] = item;

delete[] arr;

arr = arr2;

}

struct air {

double rzm; // размах крыльев

double lg; // длина

double max\_s; // максимальная скорость

double max\_h; // максимальная высота полета

string name; // название

};

void printItemAir(air\* arr, int n) { // функция полностью аналогично одноименной для авто

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

cout << "Самолет №" << i + 1 << endl;

cout << "Название: " << arr[i].name << endl;

cout << "Длина: " << arr[i].lg << endl;

cout << "Размах крыльев: " << arr[i].rzm << endl;

cout << "Максимальная скорость: " << arr[i].max\_s << endl;

cout << "Максимальная высота: " << arr[i].max\_h << endl;

cout << endl;

}

}

void searchItemAir(air\* arr, int n, string nm) { // функция полностью аналогично одноименной для авто

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

if (arr[i].name == nm) {

printItemAir(&arr[i], 1);

return;

}

}

cout << "Такой самолет не найден!" << endl;

return;

}

void removeItemAir(air\*& arr, int\* n) { // функция полностью аналогично одноименной для авто

string nm;

cout << "Введите название самолета для удаления: ";

cin >> nm;

bool ifsearch = false; // если найден

for (int i = 0; i < (\*n); i++) {

if (arr[i].name == nm)

{

arr[i].name = "NULL";

ifsearch = true;

}

};

if (!ifsearch) {

cout << "Такой элемент не найден!" << endl;

return;

}

air\* arr2 = new air[(\*n) - 1];

int j = 0;

for (int i = 0; i < (\*n) - 1; i++) {

if (arr[j].name == "NULL")

j++;

arr2[i] = arr[j];

j++;

}

(\*n)--;

delete[] arr;

arr = arr2;

return;

}

void addItemAir(air\*& arr, int\* n) { // функция полностью аналогично одноименной для авто

air\* arr2 = new air[(\*n) + 1];

for (int i = 0; i < (\*n); i++) {

arr2[i] = arr[i];

}

air item;

cout << "Введите название: ";

cin >> item.name;

cout << "Введите длину: ";

cin >> item.lg;

cout << "Введите размах: ";

cin >> item.rzm;

cout << "Введите максимальную скорость: ";

cin >> item.max\_s;

cout << "Введите максимальную высоту: ";

cin >> item.max\_h;

(\*n)++;

arr2[(\*n) - 1] = item;

delete[] arr;

arr = arr2;

}

int main() {

setlocale(LC\_ALL, "ru");

int count\_cars = 2;

int count\_ships = 2;

int count\_airs = 2;

car \*cars = new car[count\_cars]; // создание массивов структур для каждого элемента и добавление двух тестовых

cars[0].name = "FirstCar";

cars[0].lg = 2;

cars[0].hg = 1.5;

cars[0].year = 2000;

cars[0].country = "Russia";

//////

cars[1].name = "SecondCar";

cars[1].lg = 3;

cars[1].hg = 2.5;

cars[1].year = 1970;

cars[1].country = "USA";

//////////// корабли //////////////

ship\* ships = new ship[count\_ships];

ships[0].name = "1Ship";

ships[0].lg = 200;

ships[0].os = 50;

ships[0].country = "England";

//////

ships[1].name = "2Ship";

ships[1].lg = 750;

ships[1].os = 150;

ships[1].country = "France";

//////////// самолеты //////////////

air\* airs = new air[count\_airs];

airs[0].name = "FirstAir";

airs[0].lg = 200;

airs[0].rzm = 30;

airs[0].max\_s = 2000;

airs[0].max\_h = 7000;

//////

airs[1].name = "SecondAir";

airs[1].lg = 50;

airs[1].rzm = 5;

airs[1].max\_s = 400;

airs[1].max\_h = 2000;

// основное меню с выбором справочника

int number;

mainMenu: // отметка goto для возврата в основное меню

cout << "Выберите необходимый справочник: " << endl;

cout << "1. Справочник автомобилей" << endl;

cout << "2. Справочник кораблей" << endl;

cout << "3. Справочник самолетов" << endl;

cout << "4.Выход из приложения" << endl;

cin >> number;

// ниже представлена конструкция swith меню для каждой сущности и использованием описанных выше функций

switch (number) {

case 1:

system("cls");

carMenu:

cout << endl;

int number1;

cout << "Меню справочника автомобилей: " << endl;

cout << "1. Показать список авто" << endl;

cout << "2. Добавить новое авто в список" << endl;

cout << "3. Удалить из списка по названию" << endl;

cout << "4. Поиск по названию" << endl;

cout << "5. В основное меню" << endl;

cin >> number1;

if (number1 == 1) printItemCar(cars, count\_cars);

else if (number1 == 2) addItemCar(cars, &count\_cars);

else if (number1 == 3) removeItemCar(cars, &count\_cars);

else if (number1 == 4) {

cout << "Введите название авто для поиска по списку: ";

string nm;

cin >> nm;

searchItemCar(cars, count\_cars, nm);

}

else if (number1 == 5) goto mainMenu;

goto carMenu;

break;

case 2:

system("cls");

shipMenu:

cout << endl;

int number2;

cout << "Меню справочника кораблей: " << endl;

cout << "1. Показать список кораблей" << endl;

cout << "2. Добавить новый корабль в список" << endl;

cout << "3. Удалить из списка по названию" << endl;

cout << "4. Поиск по названию" << endl;

cout << "5. В основное меню" << endl;

cin >> number2;

if (number2 == 1) printItemShip(ships, count\_ships);

else if (number2 == 2) addItemShip(ships, &count\_ships);

else if (number2 == 3) removeItemShip(ships, &count\_ships);

else if (number2 == 4) {

cout << "Введите название корабля для поиска по списку: ";

string nm;

cin >> nm;

searchItemShip(ships, count\_ships, nm);

}

else if (number2 == 5) goto mainMenu;

goto shipMenu;

break;

case 3:

system("cls");

airMenu:

cout << endl;

int number3;

cout << "Меню справочника самолетов: " << endl;

cout << "1. Показать список самолетов" << endl;

cout << "2. Добавить новый самолет в список" << endl;

cout << "3. Удалить из списка по названию" << endl;

cout << "4. Поиск по названию" << endl;

cout << "5. В основное меню" << endl;

cin >> number3;

if (number3 == 1) printItemAir(airs, count\_airs);

else if (number3 == 2) addItemAir(airs, &count\_airs);

else if (number3 == 3) removeItemAir(airs, &count\_airs);

else if (number3 == 4) {

cout << "Введите название самолета для поиска по списку: ";

string nm;

cin >> nm;

searchItemAir(airs, count\_airs, nm);

}

else if (number3 == 5) goto mainMenu;

goto airMenu;

break;

case 4:

return 0;

break;

default:

cout << "Ошибка ввода, повторите";

goto mainMenu;

}

return 0;

}