# Курс



Домашнее задание №29

**«Объектно-ориентированное программирование на C++»**

**Встреча №29**

***Тема:*** *STL часть 2*

# Задание.

Создайте приложение для работы автосалона. Необходимо хранить информацию о продаваемых автомобилях (на- звание, год выпуска, объем двигателя, цену). Реализуйте интерфейс для добавления данных, удаления данных, отображения данных, сортировке данных по различным параметрам, поиску данных по различным параметрам. При реализации используйте контейнеры, функторы и алгоритмы.

Завдання

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include<Windows.h>

#include<string>

#include<fstream>

#include<algorithm>

#include<numeric>

#include<vector>

using namespace std;

class Car

{

string name;

int year;

float V;

float cost;

public:

string Get\_name()

{

return name;

}

int Get\_year()

{

return year;

}

float Get\_V()

{

return V;

}

float Get\_cost()

{

return cost;

}

Car(string name, int year, float V, float cost)

{

this->name = name;

this->year = year;

this->V = V;

this->cost = cost;

}

void Show()

{

cout << name << endl;

cout << year << " р.\n";

cout << "Об'єм двигуна: " << V << endl;

cout << "Ціна: " << cost << "$\n";

}

};

class SortByNameCar

{

public:

bool operator()(Car& left, Car& right)

{

return left.Get\_name() < right.Get\_name();

}

};

class SortByYearCar

{

public:

bool operator()(Car& left, Car& right)

{

return left.Get\_year() < right.Get\_year();

}

};

class SortByVCar

{

public:

bool operator()(Car& left, Car& right)

{

return left.Get\_V() < right.Get\_V();

}

};

class SortByCostCar

{

public:

bool operator()(Car& left, Car& right)

{

return left.Get\_cost() < right.Get\_cost();

}

};

class CarSalon

{

vector<Car>salon;

public:

CarSalon(){}

int Get\_Size()

{

return salon.size();

}

void Add\_Car(Car car)

{

salon.push\_back(car);

}

void Delete\_Car(string name)

{

bool b = true;

for (auto it = salon.begin(); it != salon.end(); it++)

{

if (it->Get\_name() == name)

{

b = false;

salon.erase(it);

cout << "Видалено!\n";

break;

}

}

if (b)

cout << "Не знайдено!\n";

}

void Show()

{

cout << "----------------------------------------------\n";

for (auto it = salon.begin(); it != salon.end(); it++)

{

it->Show();

cout << "----------------------------------------------\n";

}

}

void SortByName()

{

sort(salon.begin(), salon.end(), SortByNameCar());

}

void SortByYear()

{

sort(salon.begin(), salon.end(), SortByYearCar());

}

void SortByV()

{

sort(salon.begin(), salon.end(), SortByVCar());

}

void SortByCost()

{

sort(salon.begin(), salon.end(), SortByCostCar());

}

void FindByName(string name)

{

bool b = true;

cout << "----------------------------------------------\n";

for (auto it = salon.begin(); it != salon.end(); it++)

{

if (it->Get\_name() == name)

{

b = false;

it->Show();

cout << "----------------------------------------------\n";

}

}

if (b)

cout << "Не знайдено!\n";

}

void FindByYear(int year)

{

bool b = true;

cout << "----------------------------------------------\n";

for (auto it = salon.begin(); it != salon.end(); it++)

{

if (it->Get\_year() == year)

{

b = false;

it->Show();

cout << "----------------------------------------------\n";

}

}

if (b)

cout << "Не знайдено!\n";

}

void FindByV(float V)

{

bool b = true;

cout << "----------------------------------------------\n";

for (auto it = salon.begin(); it != salon.end(); it++)

{

if (it->Get\_V() == V)

{

b = false;

it->Show();

cout << "----------------------------------------------\n";

}

}

if (b)

cout << "Не знайдено!\n";

}

void FindByCost(float cost)

{

bool b = true;

cout << "----------------------------------------------\n";

for (auto it = salon.begin(); it != salon.end(); it++)

{

if (it->Get\_cost() == cost)

{

b = false;

it->Show();

cout << "----------------------------------------------\n";

}

}

if (b)

cout << "Не знайдено!\n";

}

};

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

CarSalon salon;

string name;

int year, k1,k2;

float V;

float cost;

cout << "Додати машину до салону->1\nВидалити машину із салону ->2\nПоказати весь салон ->3\nСортування ->4\nПошук ->5\nВихід ->0\n";

cout << "->";

cin >> k1;

while (k1 < 0 || k1>5)

{

cout << "Некоректний ввід\n";

cout << "->";

cin >> k1;

}

cout << "----------------------------------------------\n";

while (k1)

{

switch (k1)

{

case 1:

cout << "Введіть назву машини ->";

cin.ignore();

getline(cin, name);

cout << "Введіть рік випуску машини ->";

cin >> year;

cout << "Введіть об'єм двигуна машини ->";

cin >> V;

cout << "Введіть ціну машини ->";

cin >> cost;

salon.Add\_Car(Car(name, year, V, cost));

break;

case 2:

cout << "Введіть назву машини, яку треба забрати з салону->";

cin.ignore();

getline(cin, name);

salon.Delete\_Car(name);

break;

case 3:

if (salon.Get\_Size() > 0)

salon.Show();

else

cout << "Салон пустий!\n";

break;

case 4:

cout << "Сортувати за(назвою ->1, роком ->2, об'ємом двигуна ->3, ціною ->4, вихід ->0)\n";

cout << "->";

cin >> k2;

while (k2 < 0 || k2>4)

{

cout << "Некоректний ввід\n";

cout << "->";

cin >> k2;

}

switch (k2)

{

case 1:

salon.SortByName();

cout << "Відсортовано!\n";

break;

case 2:

salon.SortByYear();

cout << "Відсортовано!\n";

break;

case 3:

salon.SortByV();

cout << "Відсортовано!\n";

break;

case 4:

salon.SortByCost();

cout << "Відсортовано!\n";

break;

}

break;

case 5:

cout << "Знайти за (назвою ->1, роком ->2, об'ємом двигуна ->3, ціною ->4, вихід ->5)\n";

cout << "->";

cin >> k2;

while (k2 < 0 || k2>4)

{

cout << "Некоректний ввід\n";

cout << "->";

cin >> k2;

}

switch (k2)

{

case 1:

cout << "Введіть назву машини ->";

cin.ignore();

getline(cin, name);

salon.FindByName(name);

break;

case 2:

cout << "Введіть рік машини ->";

cin >> year;

salon.FindByYear(year);

break;

case 3:

cout << "Введіть об'єм двигуна машини ->";

cin >> V;

salon.FindByV(V);

break;

case 4:

cout << "Введіть ціну машини ->";

cin >> cost;

salon.FindByCost(cost);

break;

}

break;

}

cout << "----------------------------------------------\n";

cout << "Додати машину до салону->1\nВидалити машину із салону ->2\nПоказати весь салон ->3\nСортування ->4\nПошук ->5\nВихід ->0\n";

cout << "->";

cin >> k1;

while (k1 < 0 || k1>5)

{

cout << "Некоректний ввід\n";

cout << "->";

cin >> k1;

}

}

system("pause");

return 0;

}