Министерство образования и науки Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Алтайский государственный технический университет им. И.И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчёт защищён с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель Троицкий В.С.

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

Отчёт

по лабораторной работе № 3

«Классы и объекты языка C++»

по дисциплине «Программирование - 3»

Студент группы ПИ 92 В.М. Шульпов

Преподаватель доцент, к.т. н. Троицкий В.С.

Барнаул 2020

ЗАДАЧА

Модифицировать программу из лабораторной работы №2.

1) Заменить struct на class;

2) Функции Init, Read, Display, Add и другие, становятся методами класса;

3) Поля становятся закрытыми private, доступ к ним при необходимости осуществляется через открытые методы public;

4) В main продемонстрировать работу со статическими объектами вашего класса;

5) В main продемонстрировать работу с динамическими объектами вашего класса;

6) В main продемонстрировать работу с динамическим массивом объектов вашего класса и с массивом динамических объектов вашего класса;

7) Продемострировать применение операторов new, delete, delete[];

8) Продемострировать применение функций calloc, malloc, realloc, free;

9) При обращении к методам использовать (\*a) и ->;

!) При выполнении лабораторной работы необходимо использовать Git;

!) Язык разработки C++.

Поля структуры:

* Название
* Цена
* Цвет
* Количество оборотов двигателя в минуту
* Скорость
* Количество бензина

Функции:

* Инициализация
* Установка параметров автомобиля
* Вывод данных машины
* Запуск двигателя
* Остановка двигателя
* Добавление скорости
* Уменьшить скорость

КОД ПРОГРАММЫ

// lab work №2 Shulpov Victor PI-92

/\*

Реализовать работу автомобиля на примере динамической структуры.

Поля структуры:

• Название (строка)

• Цена (целое)

• Цвет (строка)

• Количество оборотов двигателя в минуту (целое)

• Скорость (целое)

• Количество бензина (целое)

Функции:

• Инициализация

• Установка параметров автомобиля

• Вывод данных машины

• Запуск двигателя

• Остановка двигателя

• Добавление скорости

• Уменьшить скорость

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <conio.h>

#define clear(stream) rewind((stream)) //очистка потока

#define CAR\_NUMBERS 10

class Car

{

public:

void init(char\* name, int price, char\* color, int engineRPM, int speed, int benzine);

void readCarData();

void displayDataCar();

void addBenzine(int liters);

void startEngine();

void stopEngine();

void addSpeed(int speed);

void reduceSpeed(int speed);

private:

char\* name;

int price;

char\* color;

int engineRPM;

int speed;

int benzine;

};

void Car::init(char\* name, int price, char\* color, int engineRPM, int speed, int benzine)

{

this->name = name;

this->price = price;

this->color = color;

this->engineRPM = engineRPM;

this->benzine = benzine;

this->speed = speed;

printf("Car initialized!\n");

}

void Car::readCarData() {

std::cout << "ENTER CAR DATA:" << std::endl;

std::cout << "\tname:\t";

std::cin >> this->name;

clear(stdin);

std::cout << "\tprice:\t";

std::cin >> this->price;

clear(stdin);

std::cout << "\tcolor:\t";

std::cin >> this->color;

clear(stdin);

std::cout << "\tengineRPM:\t";

std::cin >> this->engineRPM;

clear(stdin);

std::cout << "\tspeed:\t";

std::cin >> this->speed;

clear(stdin);

std::cout << "\tbenzine:\t";

std::cin >> this->benzine;

clear(stdin);

}

void Car::displayDataCar()

{

std::cout << "\t" << "Car data" << std::endl;

std::cout << "\t\tName:\t"<< this->name << std::endl;

std::cout << "\t\tPrice:\t" << this->price << std::endl;

std::cout << "\t\tColor:\t" << this->color << std::endl;

std::cout << "\t\tEngineRPM:\t" << this->engineRPM << std::endl;

std::cout << "\t\tBenzine:\t" << this->benzine << std::endl;

std::cout << "\t\tSpeed:\t" << this->speed << std::endl;

}

void Car::addBenzine(int liters)

{

std::cout << liters << "lit. benzine added!" << std::endl;

this->benzine += liters;

}

void Car::startEngine()

{

if (this->benzine > 0) {

this->engineRPM = 800;

std::cout << "Engine started!" << std::endl;

}

else {

std::cout << "No benzine. Engine didn't start!" << std::endl;

}

}

void Car::stopEngine()

{

if (this->engineRPM > 0) {

this->engineRPM = 0;

std::cout << "Engine stopped!" << std::endl;

}

else {

std::cout << "Engine stopped already!" << std::endl;

}

}

void Car::addSpeed(int speed)

{

if (this->engineRPM > 0) {

this->speed += speed;

std::cout << "Car speeded up!" << std::endl;

}

else {

std::cout << "Engine isn't starting. Car didn't speed up!" << std::endl;

}

}

void Car::reduceSpeed(int speed)

{

if (this->speed > 0) {

this->speed -= speed;

std::cout << "Car speeded down!" << std::endl;

}

else {

std::cout << "Car is parking. Car didn't speed down!" << std::endl;

}

}

int main()

{

//////////////////////////////////////STATIC OBJECT////////

std::cout << "\n\nSTATIC OBJECT\n\n";

Car static\_obj\_car;

char name[100] = "", color[100] = "";

static\_obj\_car.init(name, 0, color, 0, 0, 0); //инициализируем поля объекта

static\_obj\_car.displayDataCar();

static\_obj\_car.readCarData();

static\_obj\_car.displayDataCar();

static\_obj\_car.startEngine(); //пытаемся завести двигатель

static\_obj\_car.displayDataCar();

static\_obj\_car.addBenzine(10); //добавляем бензин

static\_obj\_car.displayDataCar();

static\_obj\_car.startEngine(); //снова пытаемся завести двигатель

static\_obj\_car.displayDataCar();

for (int i = 0; i < 4; i++) {

static\_obj\_car.addSpeed(i\*5); //добавляем скорость

static\_obj\_car.displayDataCar();

}

for (int i = 0; i < 4; i++) {

static\_obj\_car.reduceSpeed(i\*5); //убавляем скорость

static\_obj\_car.displayDataCar();

}

static\_obj\_car.stopEngine(); //останавливаем двигатель

static\_obj\_car.displayDataCar();

std::cout << "\n\nPress key to continue!\n\n";

\_getch();

std::system("cls");

////////////////////////////////////DYNAMIC OBJECT/////

std::cout << "\n\nDYNAMIC OBJECT\n\n";

Car \*dynamic\_obj\_car = new Car;

char name2[100] = "", color2[100] = "";

dynamic\_obj\_car->init(name2, 0, color2, 0, 0, 0); //инициализируем поля объекта

dynamic\_obj\_car->displayDataCar();

dynamic\_obj\_car->readCarData();

dynamic\_obj\_car->displayDataCar();

dynamic\_obj\_car->startEngine(); //пытаемся завести двигатель

dynamic\_obj\_car->displayDataCar();

dynamic\_obj\_car->addBenzine(10); //добавляем бензин

dynamic\_obj\_car->displayDataCar();

dynamic\_obj\_car->startEngine(); //снова пытаемся завести двигатель

dynamic\_obj\_car->displayDataCar();

for (int i = 0; i < 4; i++) {

dynamic\_obj\_car->addSpeed(i\*5); //добавляем скорость

dynamic\_obj\_car->displayDataCar();

}

for (int i = 0; i < 4; i++) {

dynamic\_obj\_car->reduceSpeed(i\*5); //убавляем скорость

dynamic\_obj\_car->displayDataCar();

}

dynamic\_obj\_car->stopEngine(); //останавливаем двигатель

dynamic\_obj\_car->displayDataCar();

delete dynamic\_obj\_car;

std::cout << "\n\nPress key to continue!\n\n";

\_getch();

std::system("cls");

///////////////////////////////////DYNAMIC OBJECTS ARRAY///

Car \*dynamic\_obj\_cars;

dynamic\_obj\_cars = new Car[5];

delete[] dynamic\_obj\_cars;

///////////////////////////////////MALLOC ///

Car \*dynamic\_obj\_car2;

dynamic\_obj\_car2 = (Car\*)malloc(CAR\_NUMBERS\*sizeof(Car));

free(dynamic\_obj\_car2);

//////////////////////////////////CALLOC /////

Car\* dynamic\_obj\_car3;

dynamic\_obj\_car3 = (Car\*)calloc(CAR\_NUMBERS, sizeof(Car));

free(dynamic\_obj\_car3);

//////////////////////////////////REALOC - change current allocated memory

dynamic\_obj\_car3 = (Car\*)realloc(dynamic\_obj\_car3, 2\*CAR\_NUMBERS \*sizeof(Car));

free(dynamic\_obj\_car3);

return 0;

}

ТЕСТЫ

|  |  |
| --- | --- |
| **STATIC OBJECT**  Car initialized!  Car data  Name:  Price: 0  Color:  EngineRPM: 0  Benzine: 0  Speed: 0  ENTER CAR DATA:  name: audi\_3  price: 1200000  color: blue  engineRPM: 0  speed: 0  benzine: 5  Car data  Name: audi\_3  Price: 1200000  Color: blue  EngineRPM: 0  Benzine: 5  Speed: 0  Engine started!  Car data  Name: audi\_3  Price: 1200000  Color: blue  EngineRPM: 800  Benzine: 5  Speed: 0  10lit. benzine added!  Car data  Name: audi\_3  Price: 1200000  Color: blue  EngineRPM: 800  Benzine: 15  Speed: 0  Engine started!  Car data  Name: audi\_3  Price: 1200000  Color: blue  EngineRPM: 800  Benzine: 15  Speed: 0  Car speeded up!  Car data  Name: audi\_3  Price: 1200000  Color: blue  EngineRPM: 800  Benzine: 15  Speed: 0  Car speeded up!  Car data  Name: audi\_3  Price: 1200000  Color: blue  EngineRPM: 800  Benzine: 15  Speed: 5  Car speeded up!  Car data  Name: audi\_3  Price: 1200000  Color: blue  EngineRPM: 800  Benzine: 15  Speed: 15  Car speeded up!  Car data  Name: audi\_3  Price: 1200000  Color: blue  EngineRPM: 800  Benzine: 15  Speed: 30  Car speeded down!  Car data  Name: audi\_3  Price: 1200000  Color: blue  EngineRPM: 800  Benzine: 15  Speed: 30  Car speeded down!  Car data  Name: audi\_3  Price: 1200000  Color: blue  EngineRPM: 800  Benzine: 15  Speed: 25  Car speeded down!  Car data  Name: audi\_3  Price: 1200000  Color: blue  EngineRPM: 800  Benzine: 15  Speed: 15  Car speeded down!  Car data  Name: audi\_3  Price: 1200000  Color: blue  EngineRPM: 800  Benzine: 15  Speed: 0  Engine stopped!  Car data  Name: audi\_3  Price: 1200000  Color: blue  EngineRPM: 0  Benzine: 15  Speed: 0  Press key to continue! | **DYNAMIC OBJECT**  Car initialized!  Car data  Name:  Price: 0  Color:  EngineRPM: 0  Benzine: 0  Speed: 0  ENTER CAR DATA:  name: bmw\_x6  price: 2000000  color: black  engineRPM: 0  speed: 0  benzine: 0  Car data  Name: bmw\_x6  Price: 2000000  Color: black  EngineRPM: 0  Benzine: 0  Speed: 0  No benzine. Engine didn't start!  Car data  Name: bmw\_x6  Price: 2000000  Color: black  EngineRPM: 0  Benzine: 0  Speed: 0  10lit. benzine added!  Car data  Name: bmw\_x6  Price: 2000000  Color: black  EngineRPM: 0  Benzine: 10  Speed: 0  Engine started!  Car data  Name: bmw\_x6  Price: 2000000  Color: black  EngineRPM: 800  Benzine: 10  Speed: 0  Car speeded up!  Car data  Name: bmw\_x6  Price: 2000000  Color: black  EngineRPM: 800  Benzine: 10  Speed: 0  Car speeded up!  Car data  Name: bmw\_x6  Price: 2000000  Color: black  EngineRPM: 800  Benzine: 10  Speed: 5  Car speeded up!  Car data  Name: bmw\_x6  Price: 2000000  Color: black  EngineRPM: 800  Benzine: 10  Speed: 15  Car speeded up!  Car data  Name: bmw\_x6  Price: 2000000  Color: black  EngineRPM: 800  Benzine: 10  Speed: 30  Car speeded down!  Car data  Name: bmw\_x6  Price: 2000000  Color: black  EngineRPM: 800  Benzine: 10  Speed: 30  Car speeded down!  Car data  Name: bmw\_x6  Price: 2000000  Color: black  EngineRPM: 800  Benzine: 10  Speed: 25  Car speeded down!  Car data  Name: bmw\_x6  Price: 2000000  Color: black  EngineRPM: 800  Benzine: 10  Speed: 15  Car speeded down!  Car data  Name: bmw\_x6  Price: 2000000  Color: black  EngineRPM: 800  Benzine: 10  Speed: 0  Engine stopped!  Car data  Name: bmw\_x6  Price: 2000000  Color: black  EngineRPM: 0  Benzine: 10  Speed: 0  Press key to continue! |