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 № 2

«Структуры языка C»

по дисциплине «Программирование - 3»

Студент группы ПИ 92 В.М. Шульпов

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

Барнаул 2020

ЗАДАЧА

Реализовать работу автомобиля на примере динамической структуры.

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

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

#include <malloc.h>

#define MAX\_BUFF\_SIZE 50

#define N 10

typedef struct car {

char\* name;

int price;

char\* color;

int engineRPM;

int speed;

int benzine;

};

struct car\* init(char\* name); //init

struct car\* setDataCar(struct car\* car);

void displayDataCar(struct car\* car);

struct car\* addBenzine(struct car\* car, int liters);

struct car\* startEngine(struct car\* car);

struct car\* stopEngine(struct car\* car);

struct car\* addSpeed(struct car\* car);

struct car\* reduceSpeed(struct car\* car);

int main(void)

{

int n;

struct car\* car;

car = init("BMW"); //инициализируем машину1 (выделяем под нее память, заполняем стандартными значениями)

free(car); //очищаем память, выделенную под машину1

struct car\* car1;

car1 = init("AUDI"); //инициализируем машину2

free(car1);//очищаем память, выделенную под машину2

struct car\* car3;

car3 = init("VOLVO"); //инициализируем машину3

displayDataCar(car3); //отображаем параметры машины3

setDataCar(car3); //устанавлливаем параметры: имя, цену, цвет (остальное уже 0)

displayDataCar(car3);

startEngine(car3); //пытаемся завести двигатель

displayDataCar(car3);

addBenzine(car3, 10); //добавляем бензин

displayDataCar(car3);

startEngine(car3); //снова пытаемся завести двигатель

displayDataCar(car3);

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

addSpeed(car3); //добавляем скорость по 20 км/ч 4 раза

}

displayDataCar(car3);

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

reduceSpeed(car3); //убавляем скорость по 20 км/ч 4 раза

}

displayDataCar(car3);

stopEngine(car3); //останавливаем двигатель

displayDataCar(car3);

free(car3); //очищаем память, выделенную под машину3

return 0;

}

struct car\* init(char\* name) {

struct car\* car;

car = (struct car\_stack\*)malloc(sizeof(struct car)); // выделяем память

car->name = name;

car->price = 0;

car->color = "0";

car->engineRPM = 0;

car->benzine = 0;

car->speed = 0;

printf("Car initialized!\n");

return car;

}

struct car\* setDataCar(struct car\* car) {

printf("Car parameters set!\n");

printf("\tEnter a car name: ");

char \*name = (char\*)malloc(sizeof(char));

scanf("%s", name);

car->name = name;

printf("\tEnter a car price: ");

while (getchar() != '\n');

int num = 0;

scanf("%d", &num);

car->price = num;

printf("\tEnter a car color: ");

while (getchar() != '\n');

char\* color = (char\*)malloc(sizeof(char));

scanf("%s", color);

car->color = color;

return car;

}

void displayDataCar(struct car\* car) {

puts("\tCar data");

char\* str = car->name;

printf("\t\tName:\t%s\n", str);

int num = (\*car).price;

printf("\t\tPrice:\t%d\n", num);

str = car->color;

printf("\t\tColor:\t%s\n", str);

num = (\*car).engineRPM;

printf("\t\tEngine RPM:\t%d\n", num);

num = (\*car).speed;

printf("\t\tSpeed:\t%d\n", num);

num = (\*car).benzine;

printf("\t\tBenzine:\t%d\n", num);

return car;

}

struct car\* addBenzine(struct car\* car, int liters) {

printf("%d lit. benzine added!\n", liters);

car->benzine += liters;

return car;

}

struct car\* startEngine(struct car\* car) {

if (car->benzine > 0) {

car->engineRPM = 800;

printf("Engine started!\n");

}

else {

printf("No benzine. Engine didn't start!\n");

}

return car;

}

struct car\* stopEngine(struct car\* car) {

if (car->engineRPM > 0) {

car->engineRPM = 0;

printf("Engine stopped!\n");

}

else {

printf("Engine stopped already!\n");

}

return car;

}

struct car\* addSpeed(struct car\* car) {

int speed = 20;

if (car->engineRPM > 0) {

car->speed += speed;

printf("Car speeded up!\n");

}

else {

printf("Engine isn't starting. Car didn't speed up!\n");

}

return car;

}

struct car\* reduceSpeed(struct car\* car) {

int speed = 20;

if (car->speed > 0) {

car->speed -= speed;

printf("Car speeded down!\n");

}

else {

printf("Car is parking. Car didn't speed down!\n");

}

return car;

}

ТЕСТЫ

Car initialized!

Car initialized!

Car initialized!

Car data

Name: VOLVO

Price: 0

Color: 0

Engine RPM: 0

Speed: 0

Benzine: 0

Car parameters set!

Enter a car name: volvo\_3

Enter a car price: 1200000

Enter a car color: red

Car data

Name: volvo\_3

Price: 1200000

Color: red

Engine RPM: 0

Speed: 0

Benzine: 0

No benzine. Engine didn't start!

Car data

Name: volvo\_3

Price: 1200000

Color: red

Engine RPM: 0

Speed: 0

Benzine: 0

10 lit. benzine added!

Car data

Name: volvo\_3

Price: 1200000

Color: red

Engine RPM: 0

Speed: 0

Benzine: 10

Engine started!

Car data

Name: volvo\_3

Price: 1200000

Color: red

Engine RPM: 800

Speed: 0

Benzine: 10

Car speeded up!

Car speeded up!

Car speeded up!

Car speeded up!

Car data

Name: volvo\_3

Price: 1200000

Color: red

Engine RPM: 800

Speed: 80

Benzine: 10

Car speeded down!

Car speeded down!

Car speeded down!

Car speeded down!

Car data

Name: volvo\_3

Price: 1200000

Color: red

Engine RPM: 800

Speed: 0

Benzine: 10

Engine stopped!

Car data

Name: volvo\_3

Price: 1200000

Color: red

Engine RPM: 0

Speed: 0

Benzine: 10