Лабораторная работа 7 Делегаты и события

Делегаты, события

using мой\_вариант\_лабораторная\_7;

internal class Program

{

private static void Main(string[] args)

{

Auto auto = new Auto("Audi", 60);

Console.WriteLine(auto);

Console.WriteLine();

auto.addDelAuto(m1);

auto.addDelAuto(m2);

Program program = new Program();

auto.addDelAuto(program.m3);

auto.Move();

Console.WriteLine();

Auto auto1 = new Auto("BMW", 80);

auto1.addDelAuto(m1);

auto1.addDelAuto(m2);

auto1.addDelAuto(delegate () { Console.WriteLine("Плывет на встречу неизведанному"); });

auto1.Move();

Console.WriteLine();

if (auto.EquelDel(auto1))

{

Console.WriteLine("Делегаты одинаковы");

}

else

{

Console.WriteLine("Делегаты разные");

}

Console.WriteLine();

class1 class1 = new class1();

auto.OnPrevishenie += class1.Message;

class2 class2 = new class2();

auto1.OnPrevishenie += (() => Console.WriteLine("Верхняя граница. Рекомендовано понизить скорость"));

auto.addDelSpeed(UpSpeed1);

auto.UpSpeed(40);

auto1.addDelSpeed(UpSpeed1);

auto1.UpSpeed(40);

Console.WriteLine();

Console.ReadKey();

void m1()

{

Console.WriteLine("Едет");

}

}

public static int UpSpeed1(int speed1)

{

return speed1;

}

public static void m2()

{

Console.WriteLine("Перемещается");

}

void m3()

{

Console.WriteLine("Двигается");

}

}

namespace мой\_вариант\_лабораторная\_7

{

//public delegate void DELAUTO();

//public delegate int DELSPEED(int s);

internal class Auto

{

private string name;

private int speed;

public Action deLAUTO;

public Func<int, int> DELSPEED;

event Action onPrevishenie;

public void addDelSpeed(Func<int, int> dELSPEED)

{

this.DELSPEED += dELSPEED;

}

public void addDelAuto(Action deLAUTO)

{

this.deLAUTO += deLAUTO;

}

public void Move()

{

Console.Write($"{Name} ");

deLAUTO?.Invoke();

}

public bool EquelDel(Auto other)

{

return deLAUTO == other.deLAUTO;

}

public void UpSpeed(int speed1)

{

if (DELSPEED != null)

{

speed += speed1;

Console.WriteLine(this);

}

if(speed>=95 | speed < 0)

{

onPrevishenie?.Invoke();

}

}

public Auto(string name, int speed)

{

Speed = speed;

Name = name;

}

public int Speed {

get => speed;

set

{

if (value >= 0) speed = value;

}

}

public string Name { get => name; set => name = value; }

public event Action OnPrevishenie

{

add {

onPrevishenie += value;

}

remove

{

onPrevishenie -= value;

}

}

public override string? ToString()

{

string str = "Автомобиль ";

if (name != null)

{

str += $"{Name} имеет скорость {Speed}";

}

return str;

}

}

class class1

{

public void Message()

{

Console.WriteLine("Ничего страшного. Автомобиль едет по трассе");

}

}

class class2

{

public void Message() {

Console.WriteLine("Зарегестрировано превышение. Рекомендовано понижение скорости");

}

}

}

