**«Платформа Microsoft .NET и язык программирования C#» №5**

**Тема: Делегаты и события.**

**Цель:**

Закрепить у слушателей практические навыки и теоретические знания для работы классами и объектами, свойствами. Научиться использовать структуры и перечисления.

**Задание 1. Игра** «**Автомобильные гонки**».

Разработать игру "**Автомобильные гонки**" с использованием делегатов.

1. В игре использовать несколько типов автомобилей: спортивные, легковые, грузовые и автобусы.

2. Реализовать игру «Гонки». Принцип игры: Автомобили двигаются от старта к финишу со скоростями, которые изменяются в установленных пределах случайным образом. Победителем считается автомобиль, пришедший к финишу первым

**Рекомендации по выполнению работы**

1. Разработать абстрактный класс «**автомобиль**» (класс Car). Собрать в нем все общие поля, свойства (например, скорость) методы (например, ехать).

2. Разработать классы автомобилей с конкретной реализацией конструкторов и методов, свойств. В классы автомобилей добавить необходимые события (например, финиш).

3. Класс игры должен производить запуск соревнований автомобилей, выводить сообщения о текущем положении автомобилей, выводить сообщение об автомобиле, победившем в гонках. Создать **делегаты**, обеспечивающие вызов методов из классов автомобилей (например, выйти на старт, начать гонку).

4. Игра заканчивается, когда какой-то из автомобилей проехал определенное расстояние (старт в положении 0, финиш в положении 100). Уведомление об окончании гонки (прибытии какого-либо автомобиля на финиш) реализовать с помощью событий.

**Результат решения:**

using System;

namespace test11

{

abstract class Car

{

string name;

int distance\_done;

int speed;

int maxspeed;

int P;//У кінських силах

int m;//Маса кузова у кг

public string Name

{

get { return name; }

}

public int Distance\_Done

{

get { return distance\_done; }

}

public Car(string name, int P, int m)

{

this.distance\_done = 0;

this.speed = 0;

this.name = name;

this.P = P;

this.m = m;

this.maxspeed = (int)(P / 2 \* (float)1700 / m);

}

public void Move()

{

int plus = P / 40;

if (speed \* 3.6 >= maxspeed)

{

plus = 0;

}

speed += plus;

plus = (new Random()).Next(P) / 40;

while (speed < plus)

{

plus = (new Random()).Next(P) / 40;

}

speed -= plus;

distance\_done += speed;

}

public void PrintInfo()

{

Console.WriteLine(name);

Console.WriteLine(string.Format("{0:0.00}",(speed\*3.6))+"км/год");

Console.WriteLine(((float)distance\_done/1000) + "км");

}

}

class SportCar: Car

{

public SportCar(string name, int P, int m):base(name, P, m) { }

public void PrintInfo()

{

Console.Write("Спортивна машина ");

base.PrintInfo();

}

}

class LightWeightCar : Car

{

public LightWeightCar(string name, int P, int m) : base(name, P, m) { }

public void PrintInfo()

{

Console.Write("Легкова машина ");

base.PrintInfo();

}

}

class BusCar : Car

{

public BusCar(string name, int P, int m) : base(name, P, m) { }

public void PrintInfo()

{

Console.Write("Автобус ");

base.PrintInfo();

}

}

class TruckCar : Car

{

public TruckCar(string name, int P, int m) : base(name, P, m) { }

public void PrintInfo()

{

Console.Write("Вантажна машина ");

base.PrintInfo();

}

}

delegate void Start(Car[] cars, float distance);

class GameRaceOn

{

Car[] cars;

float distance;

event Start game;

static void Start(Car[] cars, float distance)

{

Console.WriteLine("На старті:");

for (int i = 0; i < cars.Length; i++)

{

Console.WriteLine(cars[i].Name);

}

System.Threading.Thread.Sleep(2000);

Console.Clear();

}

static void Attention(Car[] cars, float distance)

{

Console.WriteLine("Увага!");

for (int i = 0; i < cars.Length; i++)

{

Console.WriteLine(cars[i].Name);

}

System.Threading.Thread.Sleep(2000);

Console.Clear();

}

static void LetsGo(Car[] cars, float distance)

{

Console.WriteLine("Руш!");

for (int i = 0; i < cars.Length; i++)

{

Console.WriteLine(cars[i].Name);

}

System.Threading.Thread.Sleep(1000);

Console.Clear();

}

static void Winner(Car[] cars, float distance)

{

int index=0;

for (int i = 1; i < cars.Length; i++)

{

if(cars[index].Distance\_Done < cars[i].Distance\_Done)

{

index = i;

}

}

Console.WriteLine("Виграв:");

Console.WriteLine(cars[index].Name);

Console.WriteLine(((float)cars[index].Distance\_Done / 1000) + "км");

}

public GameRaceOn(Car[] cars, float distance)

{

this.cars = cars;

this.distance = distance\*1000;

game += Start;

game += Attention;

game += LetsGo;

}

void ShowAll()

{

for (int i = 0; i < cars.Length; i++)

{

cars[i].PrintInfo();

Console.WriteLine("-------------------------------");

}

}

public void StartGame()

{

Console.WriteLine("Ціль: " + (float)(distance / 1000)+"км");

System.Threading.Thread.Sleep(1000);

Console.Clear();

bool b = true;

game(cars,distance);

while (b)

{

ShowAll();

for (int i = 0; i < cars.Length; i++)

{

if (cars[i].Distance\_Done >= distance)

{

b = false;

break;

}

cars[i].Move();

}

if (b)

{

System.Threading.Thread.Sleep(1000);

Console.Clear();

}

}

game = null;

Console.Clear();

game += Winner;

game(cars, distance);

}

}

class Program

{

public static void Main()

{

System.Console.OutputEncoding = System.Text.Encoding.UTF8;

Car[] cars = new Car[]

{

new SportCar("Lamborghini",700,1575),

new LightWeightCar("Reno",200,1785),

new BusCar("Ankai GK15",296,10500),

new TruckCar("MAN",179,2000)

};

GameRaceOn gameRaceOn = new GameRaceOn(cars,3.5F);

gameRaceOn.StartGame();

}

}

}