زنگ سی‌شارپ – قسمت بیست و سوم

نوشته‌ی مسعود درویشیان

لینک مستقیم این مطلب در وب‌تارگت

در قسمت قبل با برخی از کاربردهای سی‌شارپ آشنا شدید و بعضی از مطالب که بیان شده بودند نیز مرور شدند. در این قسمت نیز به حل تمرین شماره ۱۳ و تمرین برخی موارد دیگر می‌پردازیم.

حل تمرین شماره ۱۳: به‌نظر نمی‌رسد برای انجام این تمرین در چالش بزرگی قرار گرفته باشید. تنها کاری که در حل این تمرین انجام دادیم "تقسیم کار" بود.

در زیر، قالبی که برای حل این تمرین در نظر گرفتیم را می‌بینید:

```
using System;
class SimplePhoneBook
{
    static void Main()
    {
        while (true)
        {
            ShowMenu();

            // Use methods to build a phonebook

            Console.ReadLine();
        }
    }

    // You can use void methods or some ret-type else
    static void ShowMenu()
    {
        Console.Clear();
        Console.WriteLine("---- Simple Phonebook ----");
        Console.WriteLine();
        Console.WriteLine("1. Add");
        Console.WriteLine("2. Search");
        Console.WriteLine("3. Show all");
        Console.WriteLine("4. Exit");
        Console.WriteLine();
        Console.Write("Choose a number: ");
    }

    static void Add()
    {
        // Add statements
    }

    static void Search()
    {
        // Search statements
    }
```

```
    static void ShowAll()
    {
        // Show statements
    }
}
```

همان‌طور که می‌بینید، برای کارهایی که قرار است در این دفترچه تلفن ساده انجام شود، متد تعریف کردیم. برای افزودن مخاطب جدید، جستجو و نمایش نیز متدهای جداگانه‌ای تعریف کردیم. این متدها را درون یک حلقه‌ی بی‌نهایت درون متد Main() اجرا می‌کنیم زیرا قصد داریم تا زمانی‌که ما اجازه می‌دهیم، برنامه اجرا شود. مسلماً این دفترچه تلفن به آرایه‌هایی برای ذخیره‌سازی اسم و شماره تلفن و همچنین چند متغیر دیگر نیاز دارد. همین‌طور بهتر است که متد ShowMenu() رشته‌ای را که کاربر انتخاب می‌کند بازگرداند:

```
using System;
class SimplePhoneBook
{
    static int Size = 5;
    static int Counter = 0;
    static bool Found = false;
    static string[] Names = new string[Size];
    static int[] PhoneNumbers = new int[Size];

    static void Main()
    {
        while (true)
        {

            Console.ReadLine();
        }
    }

    static string ShowMenu()
    {
        Console.Clear();
        Console.WriteLine("---- Simple Phonebook ----");
        Console.WriteLine();
        Console.WriteLine("1. Add");
        Console.WriteLine("2. Search");
        Console.WriteLine("3. Show all");
        Console.WriteLine("4. Exit");
        Console.WriteLine();
        Console.Write("Choose a number: ");
        return Console.ReadLine();
    }

    static void Add()
    {

    }

    static void Search()
    {

    }
```

```
    static void ShowAll()
    {

    }
}
```

همان‌طور که می‌بینید چندین متغیر را به کلاس افزودیم و اکنون متد ShowMenu() یک رشته (که کاربر از بین گزینه‌های انتخابی، انتخاب می‌کند) را باز می‌گرداند. همان‌طور که شاید حدس زده باشید، گزینه‌ای که کاربر انتخاب می‌کند باید مورد بررسی قرار گیرد. اگر به‌یاد داشته باشید برای این منظور از دستور switch استفاده می‌کردیم. پس اکنون نیاز است که متدی تعریف کنیم تا گزینه‌ای را که کاربر وارد می‌کند، بررسی کنیم:

```
using System;
class SimplePhoneBook
{
    static int Size = 5;
    static int Counter = 0;
    static bool Found = false;
    static string[] Names = new string[Size];
    static int[] PhoneNumbers = new int[Size];

    static void Main()
    {
        while (true)
        {
            Process(ShowMenu());
            Console.ReadLine();
        }
    }

    // You can use void methods or some ret-type else
    static string ShowMenu()
    {
        Console.Clear();
        Console.WriteLine("---- Simple Phonebook ----");
        Console.WriteLine();
        Console.WriteLine("1. Add");
        Console.WriteLine("2. Search");
        Console.WriteLine("3. Show all");
        Console.WriteLine("4. Exit");
        Console.WriteLine();
        Console.Write("Choose a number: ");
        return Console.ReadLine();
    }

    static void Process(string choice)
    {
        switch (choice)
        {
            case "1":
                Console.Clear();
                Add();
                break;
            case "2":
                Console.WriteLine();
                Search();
                break;
```

```
            case “3”:
                Console.Clear();
                ShowAll();
                break;
            case “4”:
                Environment.Exit(0);
                break;
            default:
                Console.WriteLine(“Invalid Choice!”);
                break;
        }
    }

    static void Add()
    {

    }

    static void Search()
    {

    }

    static void ShowAll()
    {

    }
}
```

همان‌طور که می‌بینید متد Process() این کار را برای ما انجام می‌دهد. در متد Main()، متد ShowMenu() به‌عنوان argument به متد Process() داده می‌شود. زیرا متد ShowMenu() یک رشته را return می‌کند و متد Process() یک رشته را می‌گیرد. در مرحله‌ی بعد قصد داریم متدهای جستجو، نمایش و افزودن را تکمیل کنیم:

```
using System;
class SimplePhoneBook
{
    static int Size = 5;
    static int Counter = 0;
    static bool Found = false;
    static string[] Names = new string[Size];
    static int[] PhoneNumbers = new int[Size];

    static void Main()
    {
        while (true)
        {
            Process(ShowMenu());
            Console.ReadLine();
        }
    }

    // You can use void methods or some ret-type else
    static string ShowMenu()
    {
        Console.Clear();
        Console.WriteLine("---- Simple Phonebook ----");
        Console.WriteLine();
        Console.WriteLine("1. Add");
```

```
        Console.WriteLine("2. Search");
        Console.WriteLine("3. Show all");
        Console.WriteLine("4. Exit");
        Console.WriteLine();
        Console.Write("Choose a number: ");
        return Console.ReadLine();
    }

    static void Process(string choice)
    {
        switch (choice)
        {
            case "1":
                Console.Clear();
                Add();
                break;
            case "2":
                Console.WriteLine();
                Search();
                break;
            case "3":
                Console.Clear();
                ShowAll();
                break;
            case "4":
                Environment.Exit(0);
                break;
            default:
                Console.WriteLine("Invalid Choice!");
                break;
        }
    }

    static void Add()
    {
        Names[Counter] = GetString("Please enter your name: ");
        PhoneNumbers[Counter] = GetNumber("Please enter your phone number: ");
        Console.WriteLine("Your contact added successfully");
        Counter++;
    }

    static void Search()
    {
        string srch = GetString("Please enter name for search: ");
        Found = false;
        for (int i = 0; i < Size; i++)
        {
            if (srch == Names[i])
            {
                Found = true;
                Console.WriteLine();
                Console.WriteLine("Status: Found");
                Console.WriteLine("Contact Details: ");
                Console.WriteLine("Name: " + Names[i] + "\nNumber: " + PhoneNumbers[i]);
                break;
            }
        }
        if (!Found)
        {
            Console.WriteLine("Status: Not Found!");
        }
    }
```

```
    static void ShowAll()
    {
        for (int i = 0; i < Size; i++)
        {
            if (Names[i] == null) break;
            Console.Write(Names[i] + "\n" + PhoneNumbers[i]);
            Console.WriteLine("\n");
        }
    }

    static string GetString(string message)
    {
        Console.Write(message);
        return Console.ReadLine();
    }

    static int GetNumber(string message)
    {
        Console.Write(message);
        return Convert.ToInt32(Console.ReadLine());
    }
}
```

همان‌طور که می‌بینید حتی برای گرفتن رشته و عدد هم متدهای جداگانه‌ای تعریف کردیم. به این ترتیب، برنامه دفترچه تلفن اندکی شی‌گرا می‌شود.

اگر اهل بازی‌های کامپیوتری باشید حتماً با بازی‌های اتومبیل‌رانی آشنایی دارید. چندی پیش در حال بازی جدیدترین نسخه‌ی NFS (Most Wanted 2) بودم و به‌نظر رسید که تهیه‌ی مثالی شبیه‌سازی شده از این بازی می‌تواند جالب باشد. اگر توجه کرده باشید در یک بازی اتومبیل‌رانی شما گاز می‌دهید، ترمز می‌گیرید و تصادف می‌کنید. نکته‌ی جالب در بازی Most Wanted تعقیب و گریز در آن است. شما قانون را می‌شکنید، تحت تعقیب قرار می‌گیرید و امتیاز کسب می‌کنید. در این مثال شبیه‌سازی شده، قصد داریم یک ماشین داشته باشیم که می‌تواند گاز بدهد و بر سرعتش بیافزاید، ترمز کند، از سرعت قانونی فراتر رود، امتیاز کسب کند و تحت تعقیب قرار گیرد.

به کلاس زیر دقت کنید:

```
class Car
{
    // Fields
    private string Name;
    private string Color;
    private string Model;
    private int TopSpeed;
    private int DamageRate;
    private int Points;
    private int CurrentSpeed;
    private bool MostWanted;
    private int StatutorySpeed;
    private bool BreakDown;
```

```
    // Getting fields
    public string GetName() { return Name; }
    public string GetModol() { return Model; }
    public int GetTopSpeed() { return TopSpeed; }
    public string GetColor() { return Color; }
    public int GetStatutorySpeed() { return StatutorySpeed; }

    // Constructor
    public Car(string name, string color, string model, int topSpeed)
    {
        Name = name;
        Color = color;
        Model = model;
        TopSpeed = topSpeed;
        DamageRate = 0;
        Points = 0;
        CurrentSpeed = 0;
        MostWanted = false;
        StatutorySpeed = 200;
        BreakDown = false;
    }

    // Methods
    // Accelerate's method
    public void Accelerate(int accelerateValue)
    {
        if (BreakDown)
        {
            Console.WriteLine("Your car is out of control! Fix it first.");
            return;
        }

        if (CurrentSpeed + accelerateValue <= TopSpeed)
        {
            CurrentSpeed += accelerateValue;
            Console.WriteLine("Current Speed: {0} Km/h", CurrentSpeed);
        }
        else
        {
            Console.WriteLine("Current Speed: {0} Km/h", CurrentSpeed);
            Console.WriteLine("You can't broke the max speed!");
        }

        if (CurrentSpeed > StatutorySpeed)
        {
            MostWanted = true;
            Points += 30;
            Console.WriteLine();
            Console.ForegroundColor = ConsoleColor.Red;
            Console.WriteLine("You broke the statutory speed");
            Console.WriteLine("You're most wanted!");
            Console.WriteLine("You've to run away...");
            Console.WriteLine("Current Point: " + Poits);
            Console.ForegroundColor = ConsoleColor.Gray;
            Console.WriteLine();
        }
    }

    // Brake's method
    public void Brake()
    {
```

```
        CurrentSpeed = 0;
        Console.WriteLine("Brake!\nCurrent Speed: {0} Km/h", CurrentSpeed);
        if (MostWanted)
        {
            MostWanted = false;
            Console.ForegroundColor = ConsoleColor.Green;
            Console.WriteLine("You aren't most wanted anymore");
            Console.ForegroundColor = ConsoleColor.Gray;
        }
    }

    // Accident's method
    public void Accident()
    {
        DamageRate += 20;
        CurrentSpeed = 0;
        Console.WriteLine("OOPS! Be cerful, damage rate is " + DamageRate);

        if (DamageRate >= 60)
        {
            Console.ForegroundColor = ConsoleColor.Red;
            Console.WriteLine("Your car is breaking down! you've to fix it.");
            Console.ForegroundColor = ConsoleColor.Gray;
            BreakDown = true;
        }
    }

    // Fix's method
    public void Fix()
    {
        BreakDown = false;
        Console.WriteLine();
        Console.WriteLine("Well, your car is ok.");
    }
}
```

اجازه دهید هر قسمت از این کلاس را جداگانه شرح دهیم. در ابتدا یک سری متغیر می‌بینید که هر ماشین در این بازی می‌تواند این متغیرها را داشته باشد. متغیرها همه‌گی private هستند زیرا دلیلی ندارد که خارج از این کلاس قابل دستکاری باشند اما برای دسترسی به بعضی از آن‌ها چند متد public تعریف کرده‌ایم که مقدار آن‌ها را برمی‌گرداند:

```
// Fields
private string Name;
private string Color;
private string Model;
private int TopSpeed;
private int DamageRate;
private int Points;
private int CurrentSpeed;
private bool MostWanted;
private int StatutorySpeed;
private bool BreakDown;

// Getting fields
public string GetName() { return Name; }
public string GetModol() { return Model; }
public int GetTopSpeed() { return TopSpeed; }
public string GetColor() { return Color; }
public int GetStatutorySpeed() { return StatutorySpeed; }
```

در قسمت بعد constructor قرار گرفته است که موجب می‌شود در حین ساخت شیء از این کلاس، فیلدها مقداردهی شوند:

```
// Constructor
public Car(string name, string color, string model, int topSpeed)
{
    Name = name;
    Color = color;
    Model = model;
    TopSpeed = topSpeed;
    DamageRate = 0;
    Points = 0;
    CurrentSpeed = 0;
    MostWanted = false;
    StatutorySpeed = 200;
    BreakDown = false;
}
```

در این‌جا نام، رنگ و مدل اتومبیل نیز مشخص می‌شود. همچنین میزان خسارت وارد شده به ماشین (DamageRate)، امتیاز (Points) و سرعت جاری (CurrentSpeed) صفر در نظر گرفته شده است. در ابتدا اتومبیل مربوطه سالم است (BreakDown) و تحت تعقیب (MostWanted) نیست. سرعت مجاز (StatutorySpeed) ۲۰۰ کیلومتر بر ساعت در نظر گرفته شده است. اگر سرعت شما از این حد فراتر رود، تحت تعقیب قرار می‌گیرید و بر امتیاز شما افزوده می‌شود البته در صورتی‌که سرعت خود را به کمتر از حد مجاز برسانید دیگر تحت تعقیب نیستید. با هربار تصادف، میزان خسارت بالا می‌رود و اگر این میزان به یک حد مشخص برسد اتومبیل از کار می‌افتد و دیگر قادر به حرکت نیست و برای حرکت حتماً بایستی تعمیر شود.

در قسمت بعد متدها قرار دارند. همان‌طور که می‌دانید متدها کارهای خاصی را انجام می‌دهند. متدی وظیفه‌ی زیاد کردن سرعت ماشین را دارد و متد دیگر ترمز می‌گیرد و سرعت را کاهش می‌دهد. هنگامی که ماشین به جایی برخورد می‌کند متدی صدا زده می‌شود که میزان خسارت را بالا می‌برد. همچنین متدی برای تعمیر ماشینی که از کار افتاده است نیز وجود دارد.

به متد زیر دقت کنید:

```
// Accelerate's method
public void Accelerate(int accelerateValue)
{
    if (BreakDown)
    {
        Console.WriteLine("Your car is out of control! Fix it first.");
        return;
    }
```

```
    if (CurrentSpeed + accelerateValue <= TopSpeed)
    {
        CurrentSpeed += accelerateValue;
        Console.WriteLine("Current Speed: {0} Km/h", CurrentSpeed);
    }
    else
    {
        Console.WriteLine("Current Speed: {0} Km/h", CurrentSpeed);
        Console.WriteLine("You can't broke the max speed!");
    }

    if (CurrentSpeed > StatutorySpeed)
    {
        MostWanted = true;
        Points += 30;
        Console.WriteLine();
        Console.ForegroundColor = ConsoleColor.Red;
        Console.WriteLine("You broke the statutory speed");
        Console.WriteLine("You're most wanted!");
        Console.WriteLine("You've to run away...");
        Console.WriteLine("Current Point: " + Points);
        Console.ForegroundColor = ConsoleColor.Gray;
        Console.WriteLine();
    }
}
```

این متد وظیفه‌ی شتاب دادن به ماشین را بر عهده دارد. این متد در ابتدا بررسی می‌کند که اگر اتومبیل خراب است، پیغامی در این مورد به کاربر نشان دهد و از متد خارج شود. اگر اتومبیل سالم باشد، این متد بررسی می‌کند که مجموع سرعت جاری ماشین و مقداری که برای سرعت بخشیدن به ماشین (در پارامتر) دریافت می‌کند، بیش‌تر از سرعت نهایی ماشین نشود. این متد در نهایت بررسی می‌کند که اگر سرعت ماشین از حد مجاز فراتر رفته باشد، ۳۰ امتیاز کسب شود و ماشین تحت تعقیب قرار گیرد.

به متد بعدی توجه کنید:

```
// Brake's method
public void Brake()
{
    CurrentSpeed = 0;
    Console.WriteLine("Brake!\nCurrent Speed: {0} Km/h", CurrentSpeed);
    if (MostWanted)
    {
        MostWanted = false;
        Console.ForegroundColor = ConsoleColor.Green;
        Console.WriteLine("You aren't most wanted anymore");
        Console.ForegroundColor = ConsoleColor.Gray;
    }
}
```

با این متد سرعت اتومبیل را به صفر می‌رسانید. همچنین این متد بررسی می‌کند که اگر تحت تعقیب هستید، به‌دلیل اینکه سرعت شما صفر می‌شود، دیگر تحت تعقیب نباشید.

به متد بعد توجه کنید:

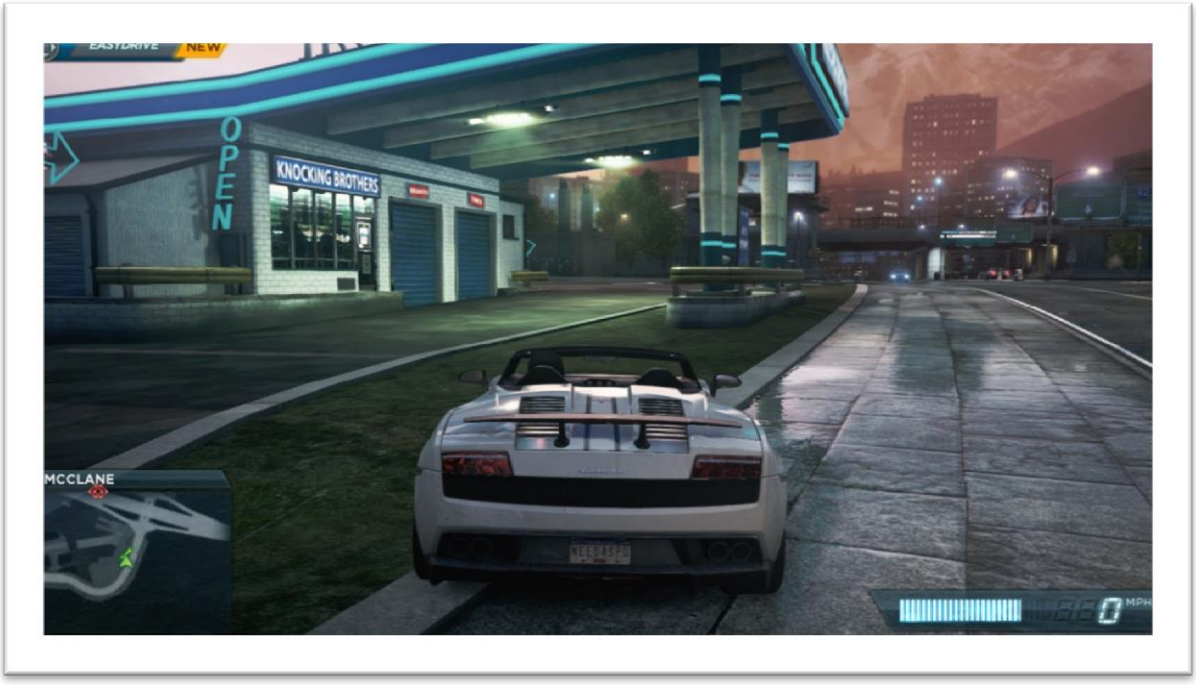
```
// Accident's method
public void Accident()
{
    DamageRate += 20;
    CurrentSpeed = 0;
    Console.WriteLine("OOPS! Be cerful, damage rate is " + DamageRate);

    if (DamageRate >= 60)
    {
        Console.ForegroundColor = ConsoleColor.Red;
        Console.WriteLine("Your car is breaking down! you've to fix it.");
        Console.ForegroundColor = ConsoleColor.Gray;
        BreakDown = true;
    }
}
```

هنگامی‌که اتومبیل مربوطه تصادف می‌کند این متد فراخوانی می‌شود. با فراخوانی این متد میزان خسارت ماشین افزایش می‌یابد و سرعت ماشین به صفر می‌رسد.



همچنین بررسی می‌شود که اگر میزان خسارت از ۶۰ بیشتر شده باشد، متغیر BreakDown برابر با true شود که این باعث می‌شود ماشین قادر به حرکت کردن و شتاب گرفتن نباشد و قبل از هر چیز، ابتدا بایستی تعمیر شود. در بازی نام برده شده، ایستگاه‌هایی در بازی هستند که در هنگام خرابی ماشین با مراجعه به آن ایستگاه‌ها، ماشین به همان حالت اول بر می‌گردد و سالم می‌شود.



متد زیر این کار را انجام می‌دهد:

```
// Fix's method
public void Fix()
{
    BreakDown = false;
    Console.WriteLine();
    Console.WriteLine("Well, your car is ok.");
}
```

در مرحله‌ی بعد قصد داریم از این کلاس و متدهایش استفاده کنیم:

```
using System;
class MyClass
{
    static void Main()
    {
        Car Lamborghini = new Car("Lamborghini", "Orange", "AVENTADOR LP 700-4", 350);

        Console.WriteLine("Car Details:");
        Console.WriteLine("  Name: " + Lamborghini.GetName());
        Console.WriteLine("  Model: " + Lamborghini.GetModol());
        Console.WriteLine("  Top Speed: " + Lamborghini.GetTopSpeed());
        Console.WriteLine("  Color: " + Lamborghini.GetColor());
        Console.WriteLine();
        Console.WriteLine("Statutory Speed: {0} Km/h", Lamborghini.GetStatutorySpeed());
        Console.WriteLine();

        Lamborghini.Accelerate(20);
        Lamborghini.Accelerate(30);
        Lamborghini.Accelerate(50);
        Lamborghini.Accelerate(130);
        Lamborghini.Accelerate(10);
        Lamborghini.Brake();
```

```
        Lamborghini.Accelerate(20);
        Lamborghini.Accelerate(150);
        Lamborghini.Accelerate(60);
        Lamborghini.Brake();
        Lamborghini.Accelerate(30);
        Lamborghini.Accident();
        Lamborghini.Accelerate(40);
        Lamborghini.Accident();
        Lamborghini.Accelerate(40);
        Lamborghini.Accident();
        Lamborghini.Accelerate(20);
        Lamborghini.Fix();
        Lamborghini.Accelerate(80);
    }
}
class Car
{
    // Fields
    private string Name;
    private string Color;
    private string Model;
    private int TopSpeed;
    private int DamageRate;
    private int Points;
    private int CurrentSpeed;
    private bool MostWanted;
    private int StatutorySpeed;
    private bool BreakDown;

    // Getting fields
    public string GetName() { return Name; }
    public string GetModol() { return Model; }
    public int GetTopSpeed() { return TopSpeed; }
    public string GetColor() { return Color; }
    public int GetStatutorySpeed() { return StatutorySpeed; }

    // Constructor
    public Car(string name, string color, string model, int topSpeed)
    {
        Name = name;
        Color = color;
        Model = model;
        TopSpeed = topSpeed;
        DamageRate = 0;
        Points = 0;
        CurrentSpeed = 0;
        MostWanted = false;
        StatutorySpeed = 200;
        BreakDown = false;
    }

    // Methods
    // Accelerate's method
    public void Accelerate(int accelerateValue)
    {
        if (BreakDown)
        {
            Console.WriteLine("Your car is out of control! Fix it first.");
            return;
        }

        if (CurrentSpeed + accelerateValue <= TopSpeed)
```

```
        {
            CurrentSpeed += accelerateValue;
            Console.WriteLine("Current Speed: {0} Km/h", CurrentSpeed);
        }
        else
        {
            Console.WriteLine("Current Speed: {0} Km/h", CurrentSpeed);
            Console.WriteLine("You can't broke the max speed!");
        }

        if (CurrentSpeed > StatutorySpeed)
        {
            MostWanted = true;
            Points += 30;
            Console.WriteLine();
            Console.ForegroundColor = ConsoleColor.Red;
            Console.WriteLine("You broke the statutory speed");
            Console.WriteLine("You're most wanted!");
            Console.WriteLine("You've to run away...");
            Console.WriteLine("Current Point: " + Points);
            Console.ForegroundColor = ConsoleColor.Gray;
            Console.WriteLine();
        }
    }

    // Brake's method
    public void Brake()
    {
        CurrentSpeed = 0;
        Console.WriteLine("Brake!\nCurrent Speed: {0} Km/h", CurrentSpeed);
        if (MostWanted)
        {
            MostWanted = false;
            Console.ForegroundColor = ConsoleColor.Green;
            Console.WriteLine("You aren't most wanted anymore");
            Console.ForegroundColor = ConsoleColor.Gray;
        }
    }

    // Accident's method
    public void Accident()
    {
        DamageRate += 20;
        CurrentSpeed = 0;
        Console.WriteLine("OOPS! Be cerful, damage rate is " + DamageRate);

        if (DamageRate >= 60)
        {
            Console.ForegroundColor = ConsoleColor.Red;
            Console.WriteLine("Your car is breaking down! you've to fix it.");
            Console.ForegroundColor = ConsoleColor.Gray;
            BreakDown = true;
        }
    }

    // Fix's method
    public void Fix()
    {
        BreakDown = false;
        Console.WriteLine();
        Console.WriteLine("Well, your car is ok.");
    }
```

```
}
```

خروجی:

```
Car Details:
  Name: Lamborghini
  Model: AVENTADOR LP 700-4
  Top Speed: 350
  Color: Orange

Statutory Speed: 200 Km/h

Current Speed: 20 Km/h
Current Speed: 50 Km/h
Current Speed: 100 Km/h
Current Speed: 230 Km/h

You broke the statutory speed
You're most wanted!
You've to run away...
Current Point: 30

 Current Speed: 240 Km/h

You broke the statutory speed
You're most wanted!
You've to run away...
Current Point: 60

Brake!
Current Speed: 0 Km/h
You aren't most wanted anymore
Current Speed: 20 Km/h
Current Speed: 170 Km/h
Current Speed: 230 Km/h

You broke the statutory speed
You're most wanted!
You've to run away...
Current Point: 90

Brake!
Current Speed: 0 Km/h
You aren't most wanted anymore
Current Speed: 30 Km/h
OOPS! Be cerful, damage rate is 20
Current Speed: 40 Km/h
OOPS! Be cerful, damage rate is 40
Current Speed: 40 Km/h
OOPS! Be cerful, damage rate is 60
Your car is breaking down! you've to fix it.
Your car is out of control! Fix it first.
Well, your car is ok.
Current Speed: 80 Km/h
```

البته همان‌طور که ذکر شد، این یک مثال شبیه‌سازی شده بود و با دنیای واقعی برنامه‌نویسی بازی خیلی فاصله دارد.