

Лабораторно упражнение № 7

Свойства на класовете

I. Теоретична част

Свойствата са членове на класа, които не отговарят директно на място за съхраняване на информация, а играят посредническа роля за достъп до полета на класа, който са със статут **private**. Самото свойство се дефинира с модификатор за достъп **public**. Достъпът до данните е посредством т.нар. елементи за достъп или аксесоари (accessors).

Елементите за достъп определят операторите и изразите, които се изпълняват при четене или запис на стойност в свойство. Елемент за достът, предназначен за четене на стойност на свойство се обозначава с ключова дума **get**. Елемент, предназначен за модифициране на стойност на свойството се обозначава с ключова дума **set**. Този елемент на достъп предава новата стойност посредством параметър, определен с ключовата дума **value**. Наличието на **get** и **set** методите определя дали свойството е за четене или за запис. Възможни са три варианта:

- Ако в тялото на свойството има само метод **get**, свойството е само за четене.
- Ако в тялото на свойството има само метод **set**, свойството е само за запис.
- Ако в тялото на свойството има както метод **get**, така и метод **set**, свойството е за четене и запис.

1. Дефиниране на свойство

Общият вид на дефиницията на свойство е следният:

тип ИмеСвойство

```
{  
    get  
    { тяло на get-метод }  
  
    set  
    { тяло на set-метод }  
}
```

Подобно на полетата и методите, свойствата също имат тип. Той определя какъв е резултатът, който ще бъде върнат от метод **get** и каква е стойността, която може да получи метод **set**. След името на свойството не се задават параметри, но свойството има тяло, в което може да има методи за достъп **get** и **set**.

Пример:

```
class Time  
{  
    private int hour, minute, second;  
  
    public int Hour  
    {
```

```

        get { return hour;}
        set { hour = value;}

    }

}

```

В случая, полето `hour` на класа е свързано със свойството `hour`. Често срещана практика е имената на свойствата да съвпадат с имената на полетата, като името на свойството е с главна буква, което го отличава от името на полето. Това е улеснение за програмистите, за да е ясно кое поле с кое свойство е свързано. Свойството `hour` е за четене и запис и осъществява връзка с поле `hour`. Аксесоар за достъп `get` връща стойността на полето, а `set` задава стойност на полето. Ключова дума `value` се свързва със зададената стойност. Достъпът до поле `hour` посредством това свойство може да се илюстрира чрез следния програмен код:

```

class Program
{
    static void Main()
    {
        Time currentTime = new Time();
        currentTime.Hour = 10;

    }
}

```

Тъй като полетата `minute` и `second` не са свързани със свойства, няма как стойностите за минута или секунда на текущото време да бъдат променени или достъпени в метод `Main`.

Не е задължително свойство на класа да бъде свързано с конкретно поле. В методите `get` и `set` може да има програмен код, както във всеки един метод. Пример:

```

class Circle
{
    private double radius;

    public Circle(double r)
    {
        radius=r;
    }

    public double Area
    {
        get
        {
            return radius*radius*Math.PI;
        }
    }

    public double Radius
    {
        get
        {
            return radius;
        }
    }
}

```

```

        set
        {
            radius = value;
        }
    }
}

```

В класа `Circle` е дефинирано свойство `Area`, което е само за четене и връща като резултат площта на кръга. Освен това, в класа е дефинирано и свойство `Radius`, което е както за четене, така и за запис и служи за достъп и промяна на стойността на полето `radius`. Класът има само един конструктор, който изисква един параметър от тип `double`. Следващият примерен код илюстрира работата със свойствата на клас `Circle` в метод `Main`:

```

class Program
{
    static void Main()
    {
        Circle cir = new Circle(4);
        Console.WriteLine("A circle with a radius {0}.", cir.Radius);
        Console.WriteLine("The area is {0}.", cir.Area);
        Console.Write("Enter a value for the radius:");
        cir.Radius = Convert.ToDouble(Console.ReadLine());
        Console.WriteLine("A circle with a radius {0}.", cir.Radius);
        Console.WriteLine("The area is {0}.", cir.Area);
    }
}

```

II. Задачи за изпълнение:

1. Да се декларира C# клас, описващ правоъгълник с височина и ширина. Класът да съдържа конструктор с 2 параметъра, които дават стойности на полетата; конструктор с параметри, който нулира стойностите на полетата; свойства само за запис, които променят стойностите на полетата и свойство за четене `Area`, което връща площта на правоъгълника. Да се създаде конзолно приложение, чрез което се дефинират три обекта от клас `Rectangle`, задават се размери на всеки един от тях и се намира кой от правоъгълниците е с най-голяма площ.
2. Да се декларира C# клас `Car`, описващ автомобил, с полета марка, цвят и регистрационен номер. Класът да съдържа два конструктора без параметри и с три параметъра, които инициализират стойностите на полетата. Класът да съдържа свойства за четене и запис на всяко едно от полетата и метод, който извежда всички данни за автомобил. Да се създаде конзолно приложение за демонстрация на работата с клас `Car`.
3. Да се декларира клас `Person`, описващ персона с име, фамилия и телефонен номер. Класът да съдържа два конструктора без параметри и с три параметъра, които инициализират стойностите на полетата. Класът да съдържа свойства за четене и запис на всяко едно от полетата и метод, който извежда всички данни за персоната. Да се създаде конзолно приложение за работата с класа.

Относно въпроси по темата на адрес: ln_zh_st@yahoo.com