

Властивість – це ідентифікатор визначеного типу, при звертанні до якого викликаються пов'язані **методи доступу**. Фактично властивість – це природне розширення поля класу. Синтаксис використанні властивості нічим не відрізняється від поля, але при звертанні до властивості **не опрацюються дані, а викликаються пов'язані методи**.

Синтаксис опису властивостей в С# [Ошибка! Источник ссылки не найден., с. 24-27] ...:

Загальний синтаксис опису властивості:

```
[специфікатор доступу] <ідентифікатор>
{ get { // тіло функції зчитування значення return ; }
  set { // тіло функції встановлення значення } }
```

Як правило, властивість оголошують відкритою (зі специфікатором доступу public). Якщо властивість використовується для доступу до деякого закритого поля, то тип властивості повинен співпадати із типом цього поля. Тип властивості не може бути void. При описі властивості може бути відсутньою або частина get або частина set, але не обидві одночасно. Якщо у описі властивості присутня тільки частина get, то така властивість називається властивістю **тільки для читання**. Якщо у описі властивості присутня тільки частина set, то така властивість називається властивістю **тільки для запису**.

Метод get повинен містити оператор return. В методі set для доступу до значення, яке встановлюється, використовується неявний параметр value.

```
class Parallelogram
{private double a, b, alfa;
 public double A
   {get { return a; }
   set { if (value >= 0)
         a = value; }}
 public double B
   {get { return b; }
   set { if (value >= 0)
         b = value; }}
...}

class Parallelepiped : Parallelogram
{private double h;
 public double H
   {get { return h; }
   set { if (value >= 0)
         h = value; }}
 public Parallelepiped(double a,double b,double h) : base(a, b, 90)
   { this.h = h; }
 new public double area()
   { return 2*(A * B + A * H + B * H); }
 public double volume()
   { return A * B * H; }
 public override void Info()
   {MessageBox.Show("Дані паралелепіеда:\nсторони по " + A.ToString() +
", "+B.ToString()+", "+H.ToString()+" од.;\nплоща: " + area().ToString() +
", кв. од.\nob'єм: " + volume().ToString() + " куб. од.", "Інформація",
MessageBoxButtons.OK, MessageBoxIcon.Information); }}}
```

Характерною рисою властивостей є те, що із властивостями можна працювати як із звичайними полями об'єкта. В той же час, при наданні властивості деякого значення насправді викликається функція встановлення значення (блок set), а при звертанні до значення властивості викликається функція зчитування значення (блок get).

Індексатори є різновидом властивостей і дозволяють звертатися до полів класу за деяким індексом. Загальний синтаксис опису індексатора:

```
[специфікатор доступу] this[<список індексів>]
{ get { //тіло функції одержання значень за індексами return ; }
  set { //тіло функції встановлення значень за індексами } }
```

Як і у випадку властивостей, як правило, індексатор оголошують відкритим (зі специфікатором доступу public). Тип індексатора не може бути void. При описі індексатора може бути відсутньою або частина get або частина set, але не обидві одночасно. Метод get повинен містити оператор return. В методі set для доступу до значення, яке встановлюється,

використовується неявний параметр value. У списку індексів вказують через кому опис індексів (тип індексу та його ім'я). Тип індексу та кількість індексів у списку може бути довільною, але найчастіше використовують один індекс цілого типу. Індексатори в основному використовуються у класах для доступу до закритого поля-масиву (одновимірний або багатовимірний), хоча можуть надавати доступ і до інших членів.

Задача. Довжини сторін початкового прямокутного паралелепіпеда задані користувачем, а кожного наступного – вдвічі менші від попереднього. Визначити номер першого паралелепіпеда, об'єм якого менший від заданого числа.

```
class EmbeddedParallelepiped : Parallelepiped
{
    public EmbeddedParallelepiped(double a, double b, double h) : base(a, b, h)
    { }

    public Parallelepiped this[int i]
    {
        get {if (i <= 0) return new Parallelepiped(A, B, H);
            else return new Parallelepiped(A / Math.Pow(2, i), B / Math.Pow(2, i),
                H / Math.Pow(2, i)); }
    }
}
```

Тоді використання об'єкту цього класу при натисненні відповідної кнопки буде таким:

```
private void button4_Click(object sender, EventArgs e)
{
    var vp=new EmbeddedParallelepiped(5,7,6);
    int i=1;
    double VMax = 17;
    if (!inputDouble(ref VMax, "Введіть обмеження об'єму")) return;
    while (vp[i].volume() > VMax) i++;
    MessageBox.Show("Вкладений паралелепіпед № " + i.ToString() + " має об'єм " +
        vp[i].volume() + " куб. од.", "Інформація", MessageBoxButtons.OK,
        MessageBoxIcon.Information); }
}
```

Приклад. Визначення вільних пар протягом тижня.

```
private void button1_Click(object sender, EventArgs e)
{
    CRozkladWeek rozkladWeek = new CRozkladWeek(4);
    string[] nameDay = new string[] { "понеділ.", "вівторок", "середа", "четвер",
        "п'ятниця", "субота", "неділя" };
    rozkladWeek[0, 1] = "Програмування";
    rozkladWeek[0, 2] = "Дискретна математика";
    string rez="";
    for (int day=0; day<=6; day++)
    {
        bool nayavno=false;
        for (int para=0; para<=5; para++)
            if (rozkladWeek[day,para]==null)
                {if (!nayavno)
                    {rez+=nameDay[day]+":\t"+(para+1).ToString(); nayavno=true;}
                    else rez+=", "+(para+1).ToString(); }
                if (nayavno) rez+="\n"; }
    if (rez == "") rez = "Вільні пари відсутні";
    else rez = "Вільні пари:\n" + rez;
    MessageBox.Show(rez); }
}
```

```
class CRozkladWeek
{
    private string[,] rozklad = new string[7,6];
    int nomerWeek;
    public CRozkladWeek(int nomerWeek)
    {
        this.nomerWeek = nomerWeek; }
    public string this[int weekDay, int para]
    {
        get {if (weekDay >= 0 && weekDay <= 6 &&
            para >= 0 && para <= 5)
                return rozklad[weekDay, para];
            else return null; }
        set {if (weekDay >= 0 && weekDay <= 6 &&
            para >= 0 && para <= 5)
                rozklad[weekDay, para]=value;}}
}
```