

# Работа с классами

# ДОПОЛНИТЕЛЬНАЯ ВОЗМОЖНОСТЬ

```
static void poisk(int[] mas1, out int? num1)
{
    num1 = null;
    for(int i=0; i<mas1.Length; i++)
    {
        if(mas1[i]<0)
        {
            num1 = i;
            break;
        }
    }
}
```

# Обращение

```
static void Main(string[] args)
{
    int? num;
    int[] arr1 = {2,5,6,7,89 };
    poisk(arr1, out num);
    if (num == null) Console.WriteLine("No number");
    else
        Console.WriteLine("Num=" + num);
}
```

# Объявление класса

```
class имя_класса  
{  
// объявление данных  
// описание методов  
}
```

# Пример объявления класса

```
class kla
{
    int[] mas1;
    double sr;
```

# Конструкторы

```
public kla()  
{  
    Console.WriteLine("Count of elements ");  
    int n = Convert.ToInt32(Console.ReadLine());  
    mas1 = new int[n];  
}  
public kla(int n)  
{  
    mas1 = new int[n];  
}
```

# Деструктор

```
~kla()    // public недопустим  
{  
    Console.WriteLine("I am a destructor");  
}
```

# ФУНКЦИИ ВВОДА/ВЫВОДА

```
public void inpt()
{
    for(int i=0;i<mas1.Length;i++)
    {
        Console.Write(i + " ");
        mas1[i] = Int32.Parse(Console.ReadLine());
    }
}

public void otpt()
{
    foreach (int x in mas1)
        Console.Write(x + " ");
    Console.WriteLine();
}
```



# Функции обработки

```
public void SrAr(int c)
{
    int sum=0, kol=0;
    foreach (int x in mas1)
        if (x > c)
            {
                sum += x; kol++;
            }
    if (kol > 0) sr = (double)sum / kol;
    else sr = null;
}
public double? GetSr()
{
    return sr;
```

# Их использование

```
kla my = new kla(4);  
    double? aver;  
    my.inpt();  
    my.SrAr(5);  
    aver = my.GetSr();  
    if (aver == null) Console.WriteLine("No");  
    else  
        Console.WriteLine("Aver=" + ((double)aver).ToString("F2"));
```

# Перегрузка операторов

Общий формат перегрузки:

```
public static тип_возвращаемого_значения operator Знак_операции (
операнды )
{
// текст функции перегрузки
}
```

# Пример перегрузки

```
public static kla operator +(kla op1, kla op2)
{
    kla temp = new kla(op1.mas1.Length);
    for (int i = 0; i < temp.mas1.Length; i++)
        temp.mas1[i] = op1.mas1[i] + op2.mas1[i];
    return temp;
}
```

# Пример перегрузки

```
public static kla operator ++(kla op)
{
    for (int i = 0; i < op.mas1.Length; i++)
        op.mas1[i]++;
    return op;
}
```

# Перегрузка операторов сравнения

```
public static bool operator >(kla op1, kla op2)
{
    bool b1 = true;
    for (int i = 0; i < op1.mas1.Length; i++)
        if (op1.mas1[i] < op2.mas1[i])
            {
                b1 = false; break;
            }
    return b1;
}
```

# Перегрузка операторов сравнения

```
public static bool operator <(kla op1, kla op2)
{
    bool b1 = true;
    for (int i = 0; i < op1.mas1.Length; i++)
        if (op1.mas1[i] > op2.mas1[i])
            {
                b1 = false; break;
            }
    return b1;
}
```

# Использование перегрузок

```
static void Main(string[] args)
{
    kla my, my1, my2;
    my = new kla(4);
    my1 = new kla(4);
    bool ind;
    my.inpt(); my1.inpt();
    my2 = my1 + my;
    my2.otpt();
    my1++;
    my1.otpt();
    ind = my > my1;
    Console.WriteLine("Bool " + ind);
}
```



# Индексатор

```
Тип_данных_элементов массива  this [int индекс]
{
get
    {
//возврат значения    }
set
    {
    // присвоение значения
}
}
```

# Пример индексатора

```
public int this[int ind1]
{
    get {
        if (ind1 >= 0 && ind1 < mas1.Length) return mas1[ind1];
        else return -1000;
    }
    set
    {
        if (ind1 >= 0 && ind1 < mas1.Length) mas1[ind1] = value;
    }
}
```

# Использование индексатора

```
kla my1;  
my1 = new kla(4);  
my1.inpt();  
for (int i = 0; i < 4; i++)  
    my1[i] += 100;  
my1.otpt();
```

# Свойтва

Определение свойства:

Тип\_данных\_свойства имя\_свойства

{

  get {

//получение значения свойства

}

  set {

// использование переданного значения свойства

}

}

# Определение класса

```
class Treug
{
    double a, b, c;
    public void Assgn(double a, double b, double c)
    {
        this.a = a;this.b = b;this.c = c;
    }
    public void otpt()
    {
        Console.WriteLine("Treug " + a + " " + b + " " + c);
    }
}
```

# СВОЙСТВО 1

```
public double a1
{
    get {
        return a;
    }
    set {
        if (value >= 0 && value < b + c) a = value;
    }
}
```

# СВОЙСТВО 2

```
public double per
```

```
{
```

```
    get { return a + b + c; }
```

```
}
```

# СВОЙСТВО 3

```
double fun1()
```

```
{ return a + b + c; }
```

```
public double perfun
```

```
{
```

```
    get { return fun1(); }
```

```
}
```



# ИСПОЛЬЗОВАНИЕ СВОЙСТВ

```
static void Main(string[] args)
{
    Treug tr1 = new Treug();
    tr1.Assgn(4, 5, 6);
    tr1.otpt();
    tr1.a1 = 100;
    tr1.otpt();
    tr1.a1 = 7;
    tr1.otpt();
    Console.WriteLine("Perim " + tr1.per);
}
```