

# C#

---

# ТЕМА 3. Классы

## Задание 2. «Свойства и массивы»:

---

- 1.Свойства класса (методы аксессоры).
- 2.Использование массивов в качестве полей класса.
- 3.Пример.

# Свойства класса

Свойства класса – нечто среднее между полем и методом, представляет собой конструкцию вида:

```
<Модификатор> <Тип свойства> <Имя свойства>  
{  
get{return <значение>}  
set{<поле>=value}  
}
```

Часто свойства связываются с закрытыми полями класса и помогают осуществить доступ к этим полям из внешних (относительно класса) частей программы. Свойства вместе с модификаторами доступа реализуют механизм защиты данных от несанкционированного доступа.

## Свойства класса

```
<Модификатор> <Тип значения> <Имя свойства>  
{  
get{return <значение>}  
set{<поле>=value}  
}
```

Как мы видим, свойство имеет заголовок и тело. В заголовке указывается модификатор доступа (обычно `public`), тип возвращаемого свойством значения и имя свойства. В теле объявлено два метода **get** и **set**. Больше ничего в теле свойства объявлять нельзя. Метод `get` имеет ключевое слово `return` и возвращает какое-либо значение (обычно значение какого-либо поля, хотя не обязательно). Метод `set` имеет ключевое слово `value` и присваивает (устанавливает) это значение полю объекта.

## Пример объявление свойства в классе triangle:

```
public class triangle
{
    int ab;           // закрытое поле
    public int Ab // СВОЙСТВО
    {
        get { return ab;}
        set { ab=value;}
    }
}
```

**Имя свойства такое же как имя поля, только начинается с большой буквы.** В свойстве может отсутствовать get или set. Тогда поле будет только для чтения или только для записи.

## Пример использование описанного свойства в программе:

```
triangle x=new triangle();
```

```
// полю ab объекта x присвоится значение 6.
```

```
x.Ab=6;
```

```
//переменной b присвоится значение поля ab объекта x
```

```
int b=x.Ab;
```

Операторы присваивания возможны, не смотря на то, что поле **ab** описано с модификатором **privat**.

# Рекомендации по программированию

Класс как тип, определенный пользователем, должен содержать:

- Скрытые (`private`) поля ;
- Конструктор без параметров, присваивающий полям нулевые значения;
- Конструктор с параметрами, присваивающий полям значения параметров;
- Методы-аксессоры (так иначе называют свойства) для всех полей;
- Набор методов, реализующих поведение объектов класса. Для доступа к методам из других классов их нужно описывать со спецификатором `public`.



# Задача

Описать класс **Ar** для работы с одномерным массивом.

Закрытые поля класса:

- **n**-количество элементов в массиве;
- **a**-одномерный целочисленный массив;
- **ko**- количества отрицательных элементов массива;

Реализовать методы:

- Конструктор с двумя параметрами ( $n$  и  $x$ ), создающий массив из  $n$  элементов и заполняющий его случайными числами из интервала  $(-x; x)$ ;
- Конструктор с одним параметром типа `string` (имя файла с числами для заполнения массива), вычисляющий  $n$  - количество чисел в файле, создающий массив из  $n$  элементов и заполняющий его из файла с заданным именем;



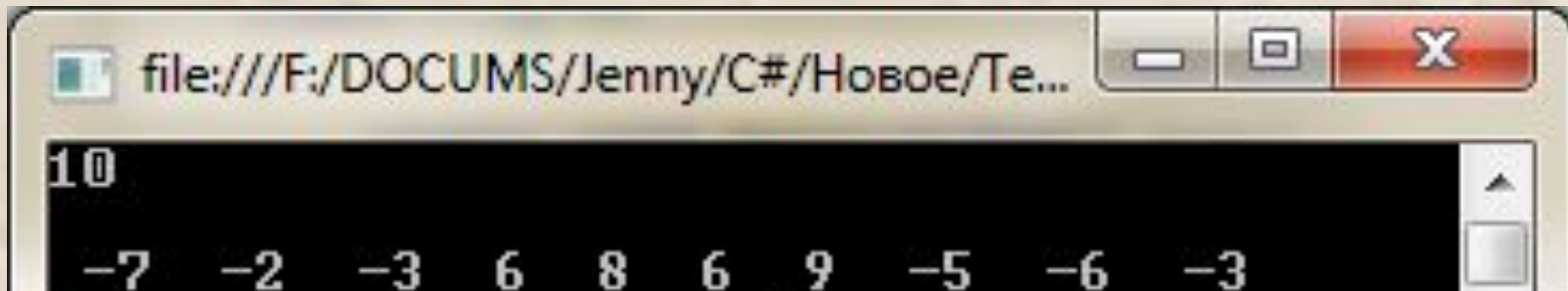
## Задача

- Метод **print** (без параметров), выводящий массив на экран;
- Свойство **Ко**, доступное только для чтения, для получения количества отрицательных элементов массива;
- Метод **Max** (без параметров), возвращающий индекс максимального элемента;
- Метод **Sum** (с одним параметром *t*), вычисляющий сумму элементов, расположенных до (левее) элемента с индексом *t*;
- Метод **Pr**(с одним параметром *p*), вычисляющий произведение элементов, расположенных после (правее) элемента с индексом *p*.

# Задача

В методе **Main** класса **Program**:

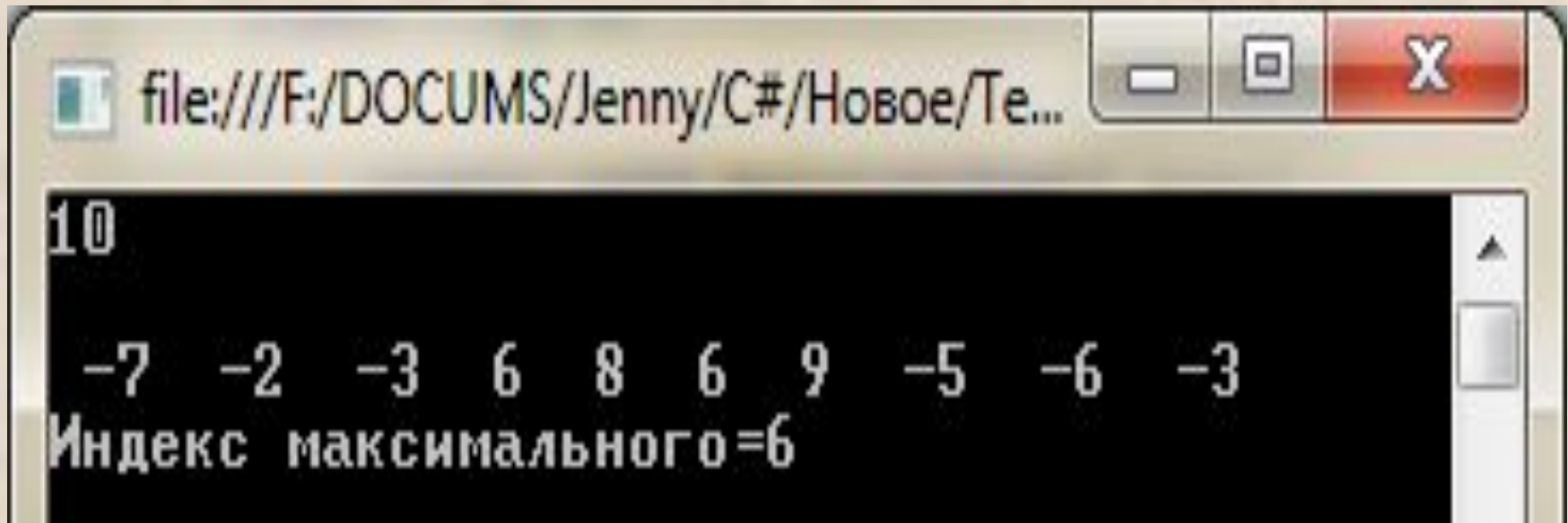
- Ввести с клавиатуры количество элементов в массиве;
- Создать объект с помощью первого конструктора;
- Применить к этому объекту метод **print**. Результат:



```
file:///F:/DOCUMS/Jenny/C#/Новое/Те...
10
-7 -2 -3 6 8 6 9 -5 -6 -3
```

# Задача

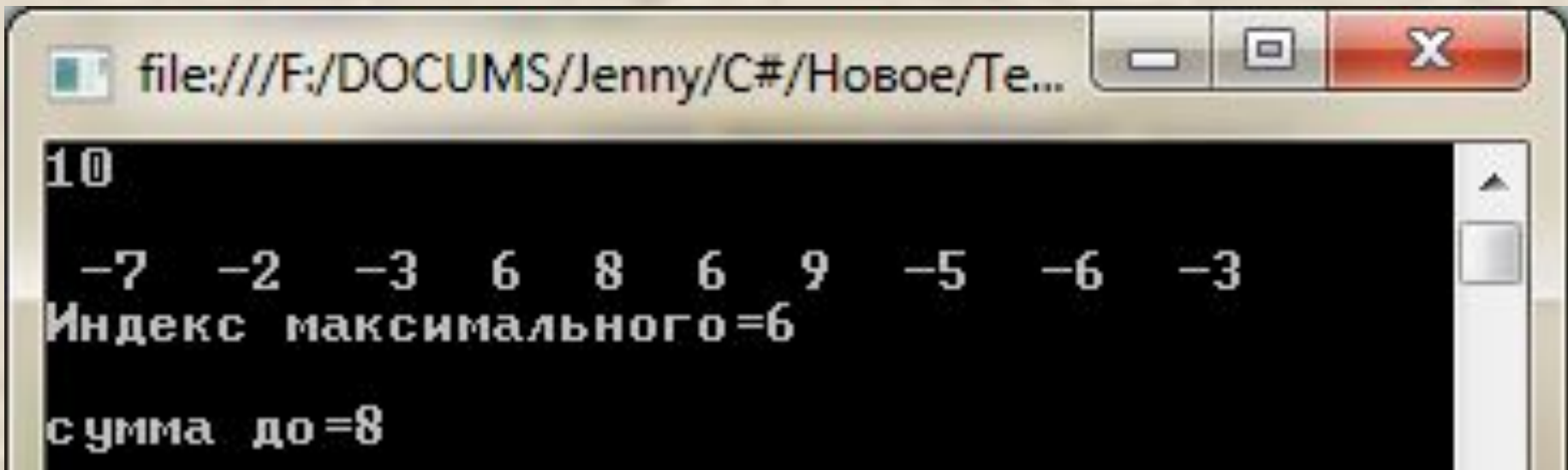
- Применить к этому объекту метод **Max** и вывести на экран полученное значение. Результат:



```
file:///F:/DOCUMENTS/Jenny/C#/Новое/Те...
10
-7 -2 -3 6 8 6 9 -5 -6 -3
Индекс максимального=6
```

# Задача

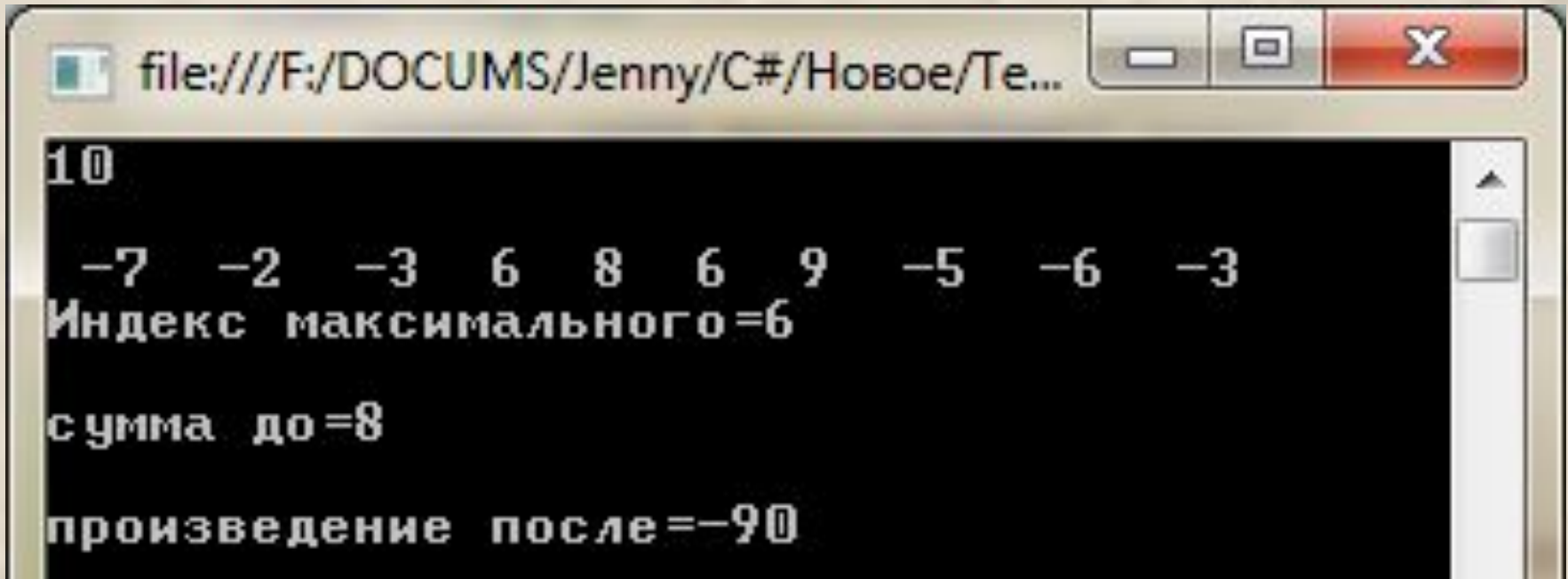
- Применить к этому объекту метод **Sum** и вывести на экран полученное значение.  
Результат:



```
file:///F:/DOCUMENTS/Jenny/C#/Новое/Те...
10
-7 -2 -3 6 8 6 9 -5 -6 -3
Индекс максимального=6
сумма до=8
```

# Задача

- Применить к этому объекту метод **Pr** и вывести на экран полученное значение. Результат:

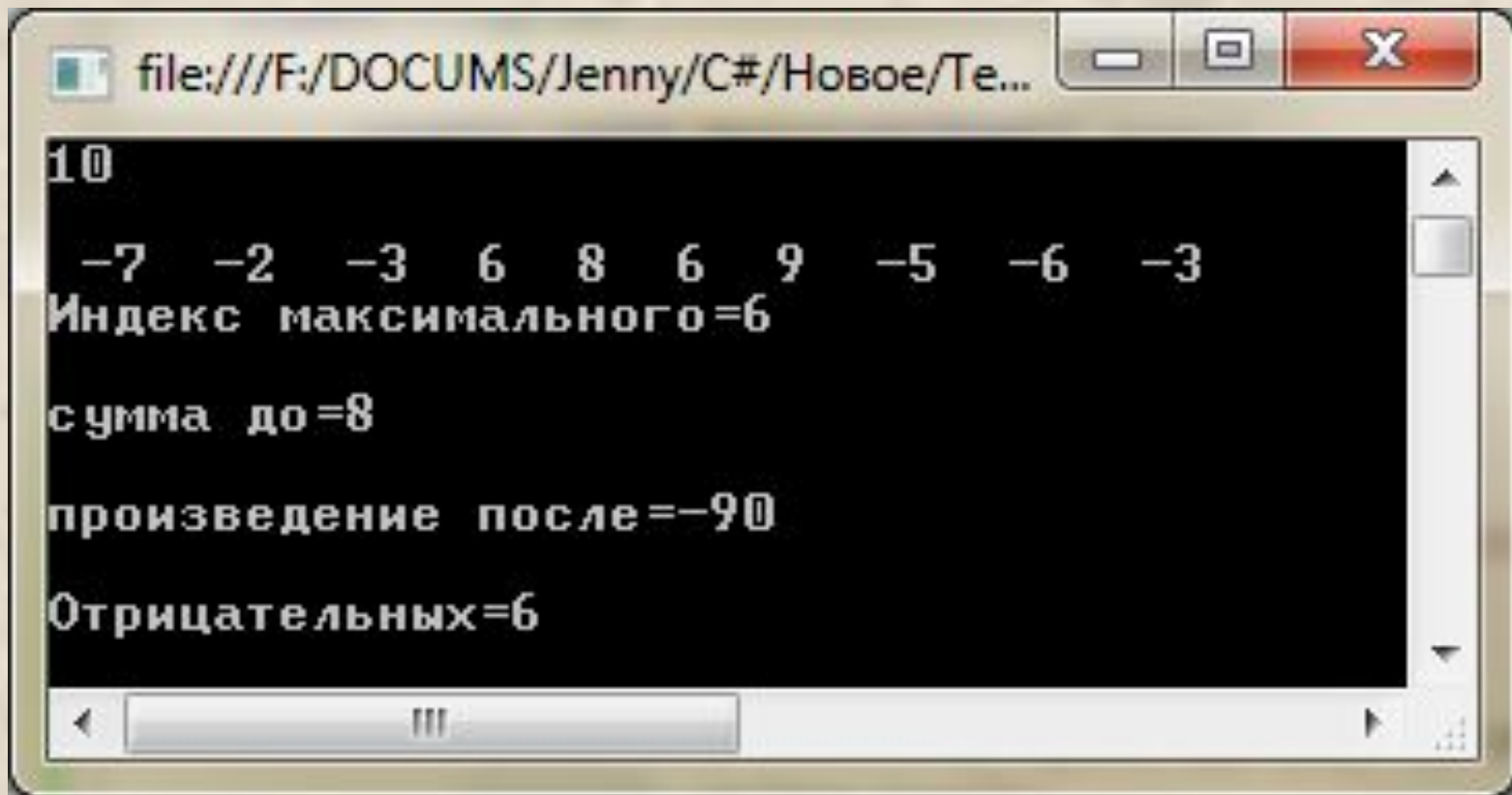


```
file:///F:/DOCUMS/Jenny/C#/Новое/Те...
10
-7 -2 -3 6 8 6 9 -5 -6 -3
Индекс максимального=6
сумма до=8
произведение после=-90
```

# Задача

- Определить значение свойства **Ко** этого объекта и вывести на экран полученное значение.

Результат:



```
file:///F:/DOCUMS/Jenny/C#/Новое/Те...
10
-7 -2 -3 6 8 6 9 -5 -6 -3
Индекс максимального=6
сумма до=8
произведение после=-90
Отрицательных=6
```



# Задача

- Создать объект с помощью второго конструктора и применить к этому объекту те же методы.



## Задача 2. Класс Ar

```
class Ar
{
    //поля
    int n; //количество элементов в массиве
    int[] a; //сам массив
    int ko; //количество отрицательных элементов
}
```

## Задача 2. Класс Ar

```
//конструктор, заполняющий массив
//случайными числами из (-x; x)
public Ar(int n,int x)
{
    this.n = n;
    a = new int[n];
    Random o = new Random();
    for (int i = 0; i < n; ++i)
        a[i] = o.Next(-x, x);
}
```

## Задача 2. Класс Ar

```
//конструктор, заполняющий массив
//из текстового файла с именем в s
public Ar(string s)
{
    //считаем количество чисел в файле
    StreamReader f = new StreamReader(s);
    string ss = "";int i=0;
    while (ss!=null)
    {
        ss = f.ReadLine();
        if (ss != null) i = i + 1;
    }
    //создаем массив для данного количества чисел
    this.n = i; a = new int[n];
    f.Close();
    //заново читаем файл и записываем числа в массив
    StreamReader f1 = new StreamReader(s);
    ss = ""; i = 0;
    while (ss != null)
    {
        ss = f1.ReadLine();
        if (ss != null)
        {
            a[i] = Convert.ToInt32(ss); i = i + 1;
        }
    }
    f1.Close();
}
```

## Задача 2. Класс Ar

```
//свойство только для чтения
public int Ko
{
    get
    {
        ko=0;
        for (int i = 0; i < n; ++i)
            if (a[i] < 0) ko = ko + 1;
        return ko;
    }
}
```

## Задача 2. Класс Ar

```
//метод вывода массива на экран
public void print()
{
    Console.WriteLine();
    for (int i = 0; i < n; ++i)
        Console.Write(" {0} ", a[i]);
    Console.WriteLine();
}
```



## Задача 2. Класс Ar

```
//метод находит наибольшее число и его индекс
public int Max()
{
    int m = a[0]; int t = 0;
    for (int i = 1; i < n; ++i)
        if (a[i] > m) { m = a[i]; t = i; };
    return t;
}
```

## Задача 2. Класс Ar

```
//складывает все числа от начала массива
//до числа с номером imax
public int Sum(int imax)
{

    int s = 0;
    for (int i = 0; i < imax; ++i)
        s = s + a[i];
    return s;
}
```



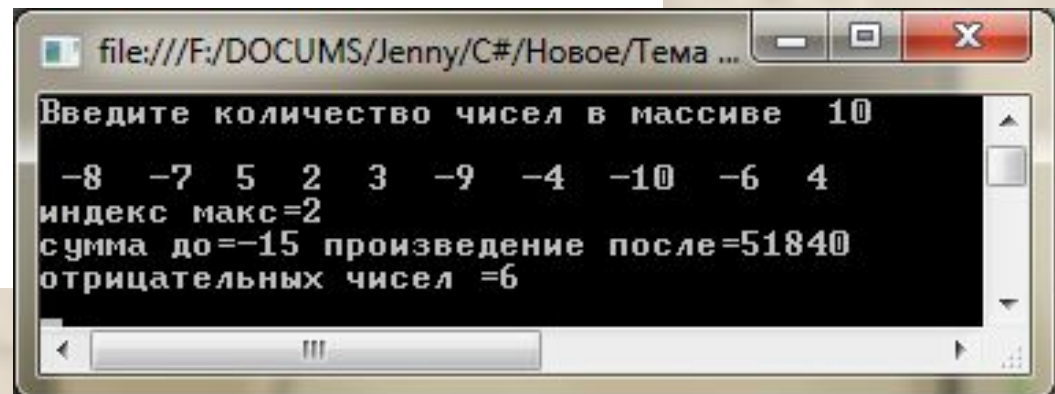
## Задача 2. Класс Ar

```
//перемножает все числа от числа с номером imax
// до конца массива
public int Pr(int imax)
{

    int s = 1;
    for (int i = imax+1; i < n; ++i)
        s = s * a[i];
    return s;
}
```

## Задача 2. Класс Program. Отработка 1

```
static void Main(string[] args)
{
    Console.Write("Введите количество чисел в массиве ");
    int k = Convert.ToInt32(Console.ReadLine());
    //вызывается конструктор, заполняющий массив
    //случайными числами из (-10,10)
    Ar mas1 = new Ar(k, 10);
    //вызывается метод print
    mas1.print();
}
```

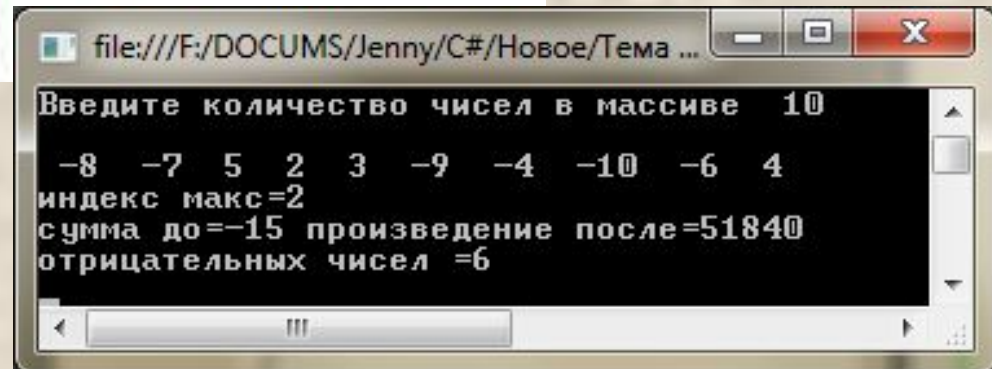


The screenshot shows a console window titled "file:///F:/DOCUMS/Jenny/C#/Новое/Тема ...". The output text is as follows:

```
Введите количество чисел в массиве 10
-8 -7 5 2 3 -9 -4 -10 -6 4
индекс макс=2
сумма до=-15 произведение после=51840
отрицательных чисел =6
```

## Задача 2. Класс Program. Отработка 1

```
//вызывается метод Max  
//находящий индекс макс числа  
int nmax1 = mas1.Max();  
Console.WriteLine("индекс макс={0}", nmax1);  
//вызывается метод Sum находящий сумму  
//чисел до числа с номером nmax  
int sm1 = mas1.Sum(nmax1);
```

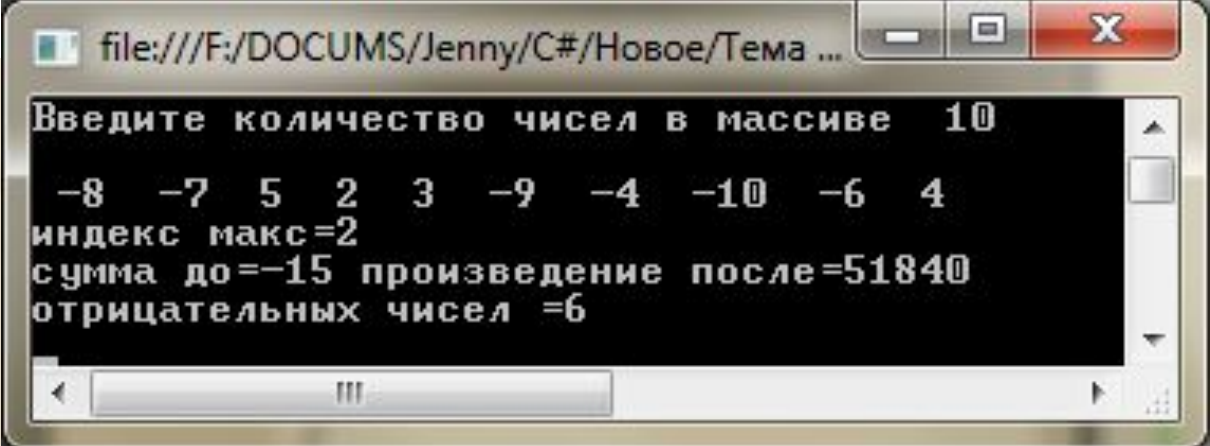


```
file:///F:/DOCUMS/Jenny/C#/Новое/Тема ...  
Введите количество чисел в массиве 10  
-8 -7 5 2 3 -9 -4 -10 -6 4  
индекс макс=2  
сумма до=-15 произведение после=51840  
отрицательных чисел =6
```

# Задача 2. Класс Program. Отработка 1

```
//вызывается метод Pr находящий произведение
//чисел после числа с номером nmax
int p1 = mas1.Pr(nmax1);
Console.WriteLine("сумма до={0} произведение после={1} ", sm1,p1);
//вызывается свойство определяющее
//количество отрицательных во всем массиве
Console.WriteLine("отрицательных чисел ={0}",mas1.Ko);
Console.ReadKey();
```

Использование свойства



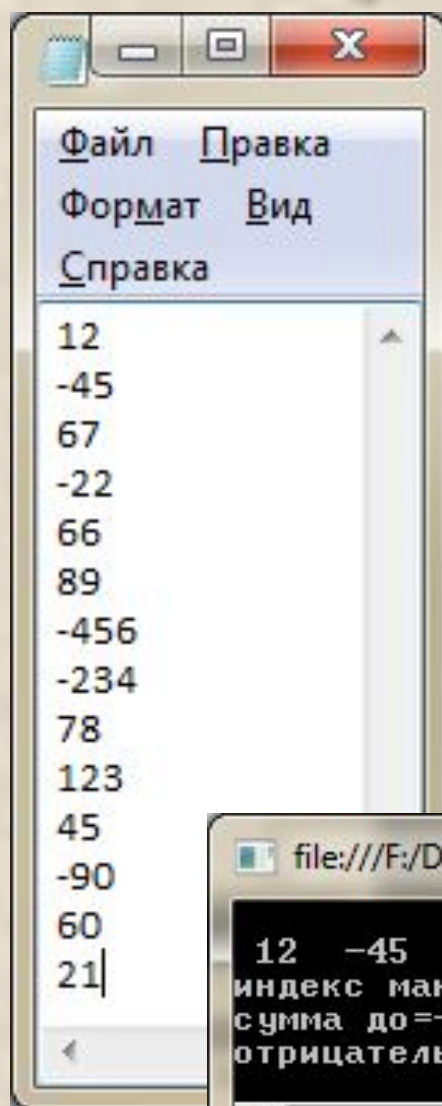
```
file:///F:/DOCUMENTS/Jenny/C#/Новое/Тема ...
Введите количество чисел в массиве 10
-8 -7 5 2 3 -9 -4 -10 -6 4
индекс макс=2
сумма до=-15 произведение после=51840
отрицательных чисел =6
```



## Задача 2. Класс Program. Отработка 2

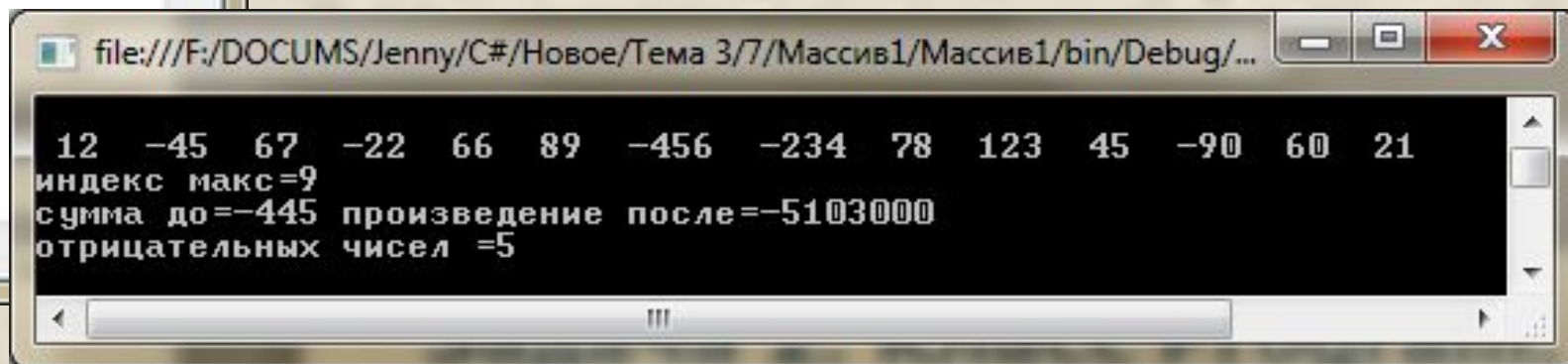
```
static void Main(string[] args)
{
    //вызывается конструктор, заполняющий
    //массив из файла 1.txt
    Ar mas2 = new Ar("1.txt");
    mas2.print();
    int nmax2 = mas2.Max();
    Console.WriteLine("индекс макс={0}", nmax2);
    int sm2 = mas2.Sum(nmax2);
    int p2 = mas2.Pr(nmax2);
    Console.WriteLine("сумма до={0} произведение после={1} ", sm2, p2);
    Console.WriteLine("отрицательных чисел ={0}", mas2.Ko);
    Console.ReadKey();
}
```

## Задача 2. Класс Program. Отработка 2



Файл П\_равка  
Формат В\_ид  
С\_правка

```
12  
-45  
67  
-22  
66  
89  
-456  
-234  
78  
123  
45  
-90  
60  
21|
```



```
file:///F:/DOCUMS/Jenny/C#/Новое/Тема 3/7/Массив1/Массив1/bin/Debug/...  
12 -45 67 -22 66 89 -456 -234 78 123 45 -90 60 21  
индекс макс=9  
сумма до=-445 произведение после=-5103000  
отрицательных чисел =5
```

# ЗАДАНИЕ

В методе **Main** класса **Program**:

- Создать объект с помощью первого конструктора;
- Вывести массив на экран с помощью метода `print`;
- Применить к этому объекту методы `a` а также получить значение свойства;
- Вывести на экран полученные значения.
- Создать объект с помощью второго конструктора;
- Вывести массив на экран с помощью метода `print`;
- Применить к этому объекту методы `a`, а также получить значение свойства;
- Вывести на экран полученные значения.



## Вариант 1

Описать класс для работы с одномерным массивом:

- Конструктор с одним параметром ( $n$ ), создающий массив из  $n$  элементов и заполняющий его однозначными случайными числами;
- Конструктор с двумя параметрами ( $n, k$ ), создающий массив из  $n$  элементов и заполняющий его натуральными числами кратными  $k$ ;
- свойство, доступное только для чтения, для получения количества элементов массива, равных нулю;
- Метод, вычисляющий  $t$  индекс первого элемента равного 0.
- Метод (с одним параметром  $t$ ), вычисляющий произведение не нулевых элементов, расположенных после элемента с индексом  $t$

## Вариант 2

Описать класс для работы с одномерным массивом:

- Конструктор с одним параметром ( $n$ ), создающий массив из  $n$  элементов и заполняющий его двухзначными случайными числами;
- Конструктор с двумя параметрами ( $n, k$ ), создающий массив из  $n$  элементов и заполняющий его натуральным рядом чисел, умноженных на заданный коэффициент ( $k$ );
- свойство, доступное только для чтения, для получения количества четных элементов массива;
- Метод, вычисляющий  $t$  индекс последнего элемента, который заканчивается на 3.
- Метод (с одним параметром  $t$ ), вычисляющий произведение элементов, расположенных до элемента с индексом  $t$

## Вариант 3

Описать класс для работы с одномерным массивом:

- Конструктор с одним параметром ( $n$ ), создающий массив из  $n$  элементов и заполняющий его натуральным рядом чисел  $(1, 2, \dots, n)$ ;
- Конструктор с двумя параметрами ( $n, k$ ), создающий массив из  $n$  элементов и заполняющий его двухзначными числами кратными  $k$ ;
- свойство, доступное только для чтения, для получения суммы нечетных элементов массива;
- Метод, вычисляющий  $t$  индекс первого элемента кратного 5.
- Метод (с одним параметром  $t$ ), вычисляющий сумму элементов, расположенных после элемента с индексом  $t$

## Вариант 4

Описать класс для работы с одномерным массивом:

- Конструктор с одним параметром ( $n$ ), создающий массив из  $n$  элементов и заполняющий его квадратами натурального ряда чисел ( $1, 4, 9 \dots n^2$ );
- Конструктор с двумя параметрами ( $n, k$ ), создающий массив из  $n$  элементов и заполняющий его натуральным рядом чисел, умноженных на заданный коэффициент ( $k$ ). Знак числа задается случайным образом;
- свойство, доступное только для чтения, для получения количества положительных элементов массива;
- Метод, вычисляющий  $t$  индекс последнего положительного элемента.
- Метод (с одним параметром  $t$ ), вычисляющий сумму модулей элементов, расположенных до элемента с индексом  $t$

## Вариант 5

Описать класс для работы с одномерным массивом:

- Конструктор с двумя параметрами ( $n, k$ ), создающий массив из  $n$  элементов и заполняющий его натуральным рядом чисел, умноженных на заданный коэффициент ( $k$ );
- Конструктор с тремя параметрами ( $n, x_1, d$ ), создающий массив из  $n$  элементов и заполняющий его членами арифметической прогрессии ( $x_1$ -первый член,  $d$ -разность). Знак числа задается случайным образом;
- свойство, доступное только для чтения, для получения суммы нечетных элементов массива;
- Метод, вычисляющий  $t$  индекс первого положительного элемента.
- Метод (с одним параметром  $t$ ), вычисляющий произведение элементов, расположенных после элемента с индексом  $t$

## Вариант 6

Описать класс для работы с одномерным массивом:

- Конструктор с двумя параметрами ( $n, k$ ), создающий массив из  $n$  элементов и заполняющий его натуральными числами кратными  $k$ ;
- Конструктор с тремя параметрами ( $n, x_1, q$ ), создающий массив из  $n$  элементов и заполняющий его членами геометрической прогрессии ( $x_1$ -первый член,  $q$ - знаменатель). Знак числа задается случайным образом;
- свойство, доступное только для чтения, для получения суммы четных элементов массива;
- Метод, вычисляющий  $t$  индекс последнего отрицательного элемента.
- Метод (с одним параметром  $t$ ), вычисляющий произведение элементов, расположенных до элемента с индексом  $t$



## Вариант 7

Описать класс для работы с одномерным массивом:

- Конструктор с двумя параметрами ( $n, k$ ), создающий массив из  $n$  элементов и заполняющий его натуральными числами кратными  $k$ . Знак числа задается случайным образом;
- Конструктор с одним параметром (имя файла с числами для заполнения массива), вычисляющий  $n$  - количество чисел в файле, создающий массив из  $n$  элементов и заполняющий его из файла с заданным именем;
- свойство, доступное только для чтения, для получения количества отрицательных элементов массива;
- Метод, вычисляющий  $t$  индекс первого положительного элемента.
- Метод (с одним параметром  $t$ ), вычисляющий сумму модулей элементов, расположенных после элемента с индексом  $t$

## Вариант 8

Описать класс для работы с одномерным массивом:

- Конструктор с одним параметром ( $n$ ), создающий массив из  $n$  элементов и заполняющий его натуральным рядом чисел  $(1, 2, 3, \dots, n)$ , знак числа задается случайным образом;
- Конструктор с тремя параметрами ( $n, x_1, d$ ), создающий массив из  $n$  элементов и заполняющий его членами арифметической прогрессии ( $x_1$ -первый член,  $d$ -разность);
- свойство, доступное только для чтения, для получения количества положительных элементов массива;
- Метод, вычисляющий  $t$  индекс последнего положительного элемента.
- Метод (с одним параметром  $t$ ), вычисляющий сумму модулей элементов, расположенных до элемента с индексом  $t$

## Вариант 9

Описать класс для работы с одномерным массивом:

- Конструктор с одним параметром ( $n$ ), создающий массив из  $n$  элементов и заполняющий его кубами натурального ряда чисел ( $1, 8, 27 \dots n^3$ ), знак числа задается случайным образом;
- Конструктор с тремя параметрами ( $n, x_1, q$ ), создающий массив из  $n$  элементов и заполняющий его членами геометрической прогрессии ( $x_1$ -первый член,  $q$ - знаменатель).
- свойство, доступное только для чтения, для получения количества элементов массива больших 100;
- Метод, вычисляющий  $t$  индекс первого отрицательного элемента.
- Метод (с одним параметром  $t$ ), вычисляющий произведение модулей элементов, расположенных после элемента с индексом  $t$

## Вариант 10

Описать класс для работы с одномерным массивом:

- Конструктор с тремя параметрами ( $n, x_1, d$ ), создающий массив из  $n$  элементов и заполняющий его членами арифметической прогрессии ( $x_1$ -первый член,  $d$ -разность);
- Конструктор с одним параметром ( $n$ ), создающий массив из  $n$  элементов и заполняющий его кубами натурального ряда чисел ( $1, 8, 27, \dots, n^3$ ), знак числа задается случайным образом;
- свойство, доступное только для чтения, для получения количества элементов массива меньших 50;
- Метод, вычисляющий  $t$  индекс последнего положительного элемента.
- Метод (с одним параметром  $t$ ), вычисляющий произведение модулей элементов, расположенных до элемента с индексом  $t$

## Вариант 11

Описать класс для работы с одномерным массивом:

- Конструктор с тремя параметрами ( $n, x_1, q$ ), создающий массив из  $n$  элементов и заполняющий его членами геометрической прогрессии ( $x_1$ -первый член,  $q$ - знаменатель).
- Конструктор с одним параметром (строка `string` с числами, числа в строке разделяются ровно одним пробелом), вычисляющий  $n$  - количество чисел в строке, создающий массив из  $n$  элементов и заполняющий его числами из строки. Использовать метод `Split` для строк.
- свойство, доступное только для чтения, для получения количества элементов массива, которые заканчиваются на 0;
- Метод, вычисляющий  $t$  индекс первого элемента кратного 3.
- Метод (с одним параметром  $t$ ), вычисляющий сумму элементов, расположенных после элемента с индексом  $t$



## Вариант 12

Описать класс для работы с одномерным массивом:

- Конструктор с тремя параметрами ( $n, x_1, d$ ), создающий массив из  $n$  элементов и заполняющий его членами арифметической прогрессии ( $x_1$ -первый член,  $d$ -разность). Знак числа задается случайным образом;
- Конструктор с одним параметром (строка `string` с числами, числа в строке разделяются точками), вычисляющий  $n$  - количество чисел в строке, создающий массив из  $n$  элементов и заполняющий его числами из строки. Использовать метод `Split` для строк.
- свойство, доступное только для чтения, для получения суммы элементов массива, которые заканчиваются на 7;
- Метод, вычисляющий  $t$  индекс последнего элемента кратного 3.
- Метод (с одним параметром  $t$ ), вычисляющий сумму элементов, расположенных до элемента с индексом  $t$

## Вариант 13

Описать класс для работы с одномерным массивом:

- Конструктор с тремя параметрами ( $n, x_1, q$ ), создающий массив из  $n$  элементов и заполняющий его членами геометрической прогрессии ( $x_1$ -первый член,  $q$ - знаменатель). Знак числа задается случайным образом;
- Конструктор с одним параметром (строка `string` с числами, числа в строке разделяются двоеточиями), вычисляющий  $n$  - количество чисел в строке, создающий массив из  $n$  элементов и заполняющий его числами из строки. Использовать метод `Split` для строк.
- свойство, доступное только для чтения, для получения количества положительных элементов массива;
- Метод, вычисляющий  $t$  индекс первого отрицательного элемента.
- Метод (с одним параметром  $t$ ), вычисляющий сумму модулей элементов, расположенных после элемента с индексом  $t$

## Вариант 14

Описать класс для работы с одномерным массивом:

- Конструктор с одним параметром ( $n$ ), создающий массив из  $n$  элементов и заполняющий его простыми числами, большими числа 1000;
- Конструктор с одним параметром (строка `string` с числами, числа в строке разделяются точкой с запятой), вычисляющий  $n$  - количество чисел в строке, создающий массив из  $n$  элементов и заполняющий его числами из строки. Использовать метод `Split` для строк.
- свойство, доступное только для чтения, для получения первого нечетного элемента;
- Метод, вычисляющий  $t$  индекс максимального нечетного элемента.
- Метод (с одним параметром  $t$ ), вычисляющий произведение элементов, расположенных до элемента с индексом  $t$

## Вариант 15

Описать класс для работы с одномерным массивом:

- Конструктор с одним параметром ( $n$ ), создающий массив из  $n$  элементов и заполняющий его числами Фибоначчи, большими числа 100;
- Конструктор с одним параметром (строка `string` с числами, числа в строке разделяются запятыми), вычисляющий  $n$  - количество чисел в строке, создающий массив из  $n$  элементов и заполняющий его числами из строки. Использовать метод `Split` для строк.
- свойство, доступное только для чтения, для получения последнего четного элемента;
- Метод, вычисляющий  $t$  индекс минимального нечетного элемента.
- Метод (с одним параметром  $t$ ), вычисляющий произведение элементов, расположенных после элемента с индексом  $t$

## Вариант 16

Описать класс для работы с одномерным массивом:

- Конструктор с одним параметром ( $n$ ), создающий массив из  $n$  элементов и заполняющий его простыми числами, большими числа 500. Знак числа задается случайным образом;
- Конструктор с одним параметром (строка `string` с числами, числа в строке разделяются одним или несколькими пробелами), вычисляющий  $n$  - количество чисел в строке, создающий массив из  $n$  элементов и заполняющий его числами из строки.
- свойство, доступное только для чтения, для получения первого отрицательного элемента;
- Метод, вычисляющий  $t1$  индекс максимального элемента.
- Метод, вычисляющий  $t2$  индекс минимального элемента.
- Метод (с двумя параметрами  $t1$  и  $t2$ ), вычисляющий сумму элементов, расположенных между элементами с индексами  $t1$  и  $t2$



## Вариант 17

Описать класс для работы с одномерным массивом:

- Конструктор с одним параметром ( $n$ ), создающий массив из  $n$  элементов и заполняющий его числами Фибоначчи, большими числа 500. Знак числа задается случайным образом;
- Конструктор с одним параметром (строка `string` с числами, числа в строке разделяются запятыми и возможны пробелы), вычисляющий  $n$  - количество чисел в строке, создающий массив из  $n$  элементов и заполняющий его числами из строки.
- свойство, доступное только для чтения, для получения последнего положительного элемента;
- Метод, вычисляющий  $t1$  индекс максимального по модулю элемента.
- Метод, вычисляющий  $t2$  индекс минимального по модулю элемента.
- Метод (с двумя параметрами  $t1$  и  $t2$ ), вычисляющий произведение элементов, расположенных между элементами с индексами  $t1$  и  $t2$