

Массивы

```
Int32 User_Health = 100;
Int32 User_Attack;
Int32 S = 0;

Random rnd = new Random();

Int32 Player1_Health = 100;
Int32 Player1_Attack;
Int32 Player2_Health = 100;
Int32 Player2_Attack;
Int32 Player3_Health = 100;
Int32 Player3_Attack;
Int32 Player4_Health = 100;
Int32 Player4_Attack;
Int32 Player5_Health = 100;
Int32 Player5_Attack;

Console.WriteLine("Начнем игру!");
Console.ReadLine();
```

```
do
{
    User_Attack = rnd.Next(15,30);
    Player1_Attack=rnd.Next(2,6);
    Player2_Attack=rnd.Next(2,6);
    Player3_Attack=rnd.Next(2,6);
    Player4_Attack=rnd.Next(2,6);
    Player5_Attack=rnd.Next(2,6);

    Console.WriteLine("Нажмите 1 для атаки!");
    Console.ReadLine();

    User_Health -= Player1_Attack;
    User_Health -= Player2_Attack;
    User_Health -= Player3_Attack;
    User_Health -= Player4_Attack;
    User_Health -= Player5_Attack;

    Console.WriteLine("У тебя осталось здоровья " + User_Health + " единиц");
    Console.ReadLine();

    Player1_Health -= User_Attack;
    Player2_Health -= User_Attack;
    Player3_Health -= User_Attack;
    Player4_Health -= User_Attack;
    Player5_Health -= User_Attack;

    S++;
    if ((S==5)&&(User_Health>0))
    {
        Console.WriteLine("Ты выжил в 5 раундах и всех победил!");
        Console.ReadLine();
        Environment.Exit(0);
    }
}
while (User_Health >= 0);

Console.WriteLine("Увы и ах( Ты проиграл 5-м противникам!");
Console.ReadLine();
```

5 ПРОТИВНИКОВ

```
Int32[] Players = new Int32[100];
```

```
Int32 User_Health = 15000;
```

```
Int32 User_Attack;
```

```
Int32 S = 0;
```

```
Random rnd = new Random();
```

```
for (int i = 0; i <= 98; i += 2)
```

```
{
```

```
    Players[i] = 100;
```

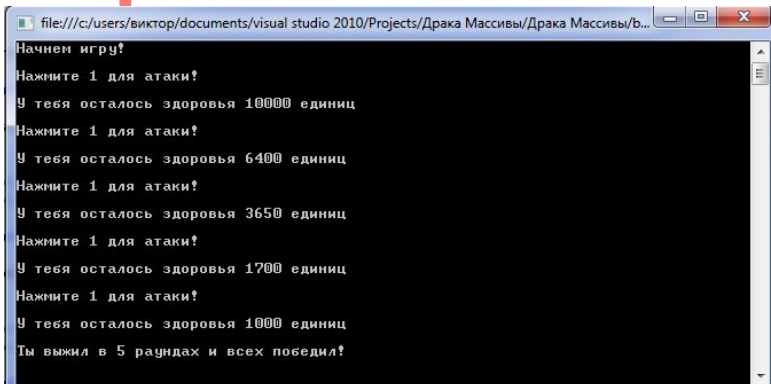
```
}
```

```
Console.WriteLine("Начнем игру!");
```

```
Console.ReadLine();
```

50

ПРОТИВНИКОВ!!



```
file:///c:/users/виктор/documents/visual studio 2010/Projects/Драка Массивы/Драка Массивы/b...
Начнем игру!
Нажмите 1 для атаки!
У тебя осталось здоровья 10000 единиц
Нажмите 1 для атаки!
У тебя осталось здоровья 6400 единиц
Нажмите 1 для атаки!
У тебя осталось здоровья 3650 единиц
Нажмите 1 для атаки!
У тебя осталось здоровья 1700 единиц
Нажмите 1 для атаки!
У тебя осталось здоровья 1000 единиц
Ты выжил в 5 раундах и всех победил!
```

```
do
```

```
{
```

```
    User_Attack = rnd.Next(15,30);
```

```
    for (int i = 1; i <= 99; i += 2)
```

```
    {
```

```
        Players[i] = rnd.Next(1, 3);
```

```
    }
```

```
    Console.WriteLine("Нажмите 1 для атаки!");
```

```
    Console.ReadLine();
```

```
    for (int i = 0; i <= 98; i += 2)
```

```
    {
```

```
        User_Health -= Players[i];
```

```
    }
```

```
    Console.WriteLine("У тебя осталось здоровья " + User_Health + " единиц");
```

```
    Console.ReadLine();
```

```
    for (int i = 0; i <= 98; i += 2)
```

```
    {
```

```
        Players[i] -= User_Attack;
```

```
    }
```

```
    S++;
```

```
    if ((S==5)&&(User_Health>0))
```

```
    {
```

```
        Console.WriteLine("Ты выжил в 5 раундах и всех победил!");
```

```
        Console.ReadLine();
```

```
        Environment.Exit(0);
```

```
    }
```

```
}
```

```
while (User_Health >= 0);
```

```
Console.WriteLine("Увы и ах( Ты проиграл 5-м противникам!");
```

```
Console.ReadLine();
```

Массив – набор однотипных элементов
(чисел, букв, слов и т.д.)

Скобки, указывающие, что это массив



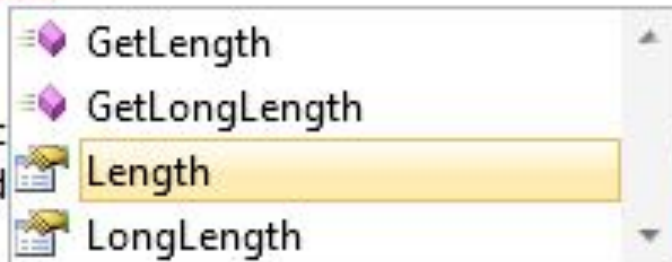
Для массивов можно применять следующие команды:

- Clear – очистка массива от имеющихся данных
- Copy – копирование массива
- Sort – сортировка массива
- Length – получение количества элементов в массиве (его длины)

```
for (int i = 0; i <= 98; i += 2)
{
    Players[i] = 100;
    Players.Length
}
```

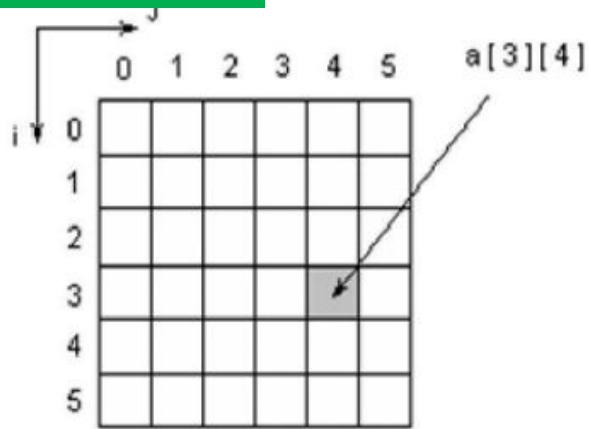
```
Console.WriteLine
Console.ReadLine
```

```
do
```



int Array.Length
Получает 32-разрядное целое ч
System.Array.

Двумерный массив



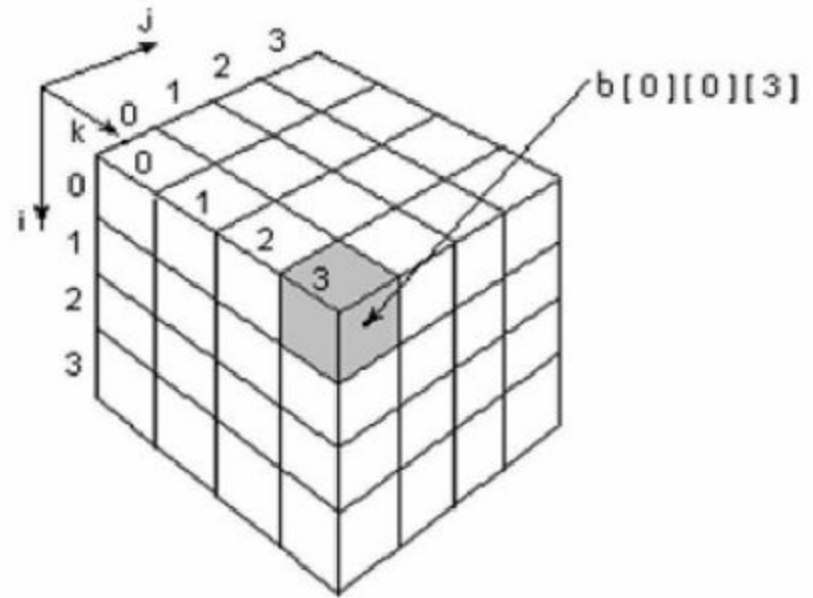
`Int32[,] Players_Attack = new Int32 [5,5];`

T
Разделитель, показывающий
что массив двумерный

T
Число строк
и столбцов
в массиве
(всего 5*5
элементов)

`Players_Options[1, 2] = 100;`

Трёхмерный

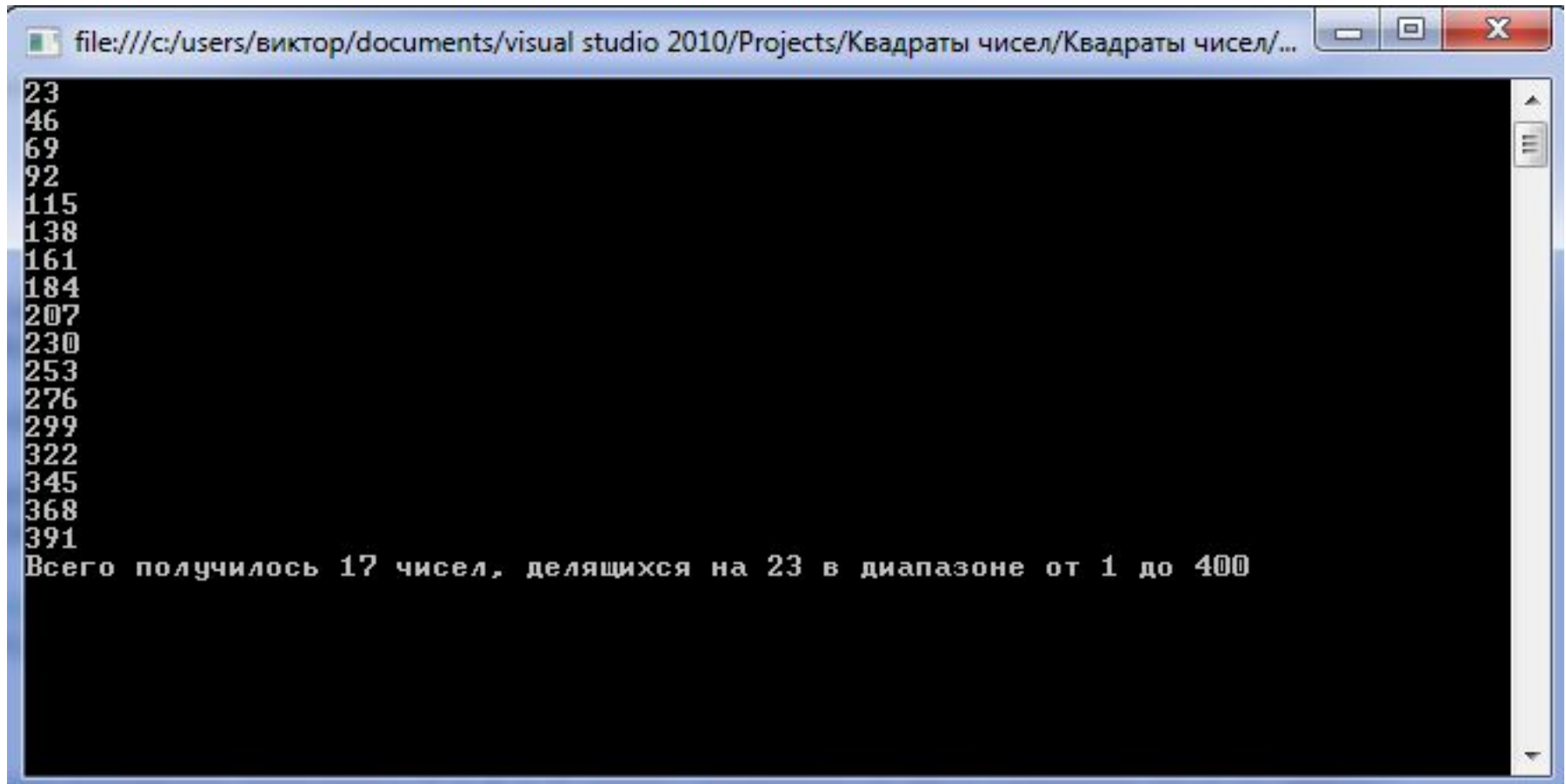


`Int32[,,] XYZ = new Int32 [10,10,10];`

`XYZ[4,1,10] = 55;`

Практика

Найти среди чисел от 1 до 400 те, которые делятся на 23, подсчитать их число вывести их на экран



```
file:///c:/users/виктор/documents/visual studio 2010/Projects/Квадраты чисел/Квадраты чисел/...  
23  
46  
69  
92  
115  
138  
161  
184  
207  
230  
253  
276  
299  
322  
345  
368  
391  
Всего получилось 17 чисел, делящихся на 23 в диапазоне от 1 до 400
```

```
Int32[] Delenie23 = new Int32[401];  
Int32 S = 0;
```

```
for (int i=1; i<=400; i++)  
{  
    Delenie23[i] = i;  
    if (Delenie23[i] % 23 == 0)  
    {  
        Console.WriteLine(Delenie23[i]);  
        S++;  
    }  
}
```

```
    Console.WriteLine("Всего получилось " + S + " чисел, делящихся на 23 в  
диапазоне от 1 до 400");  
    Console.ReadLine();  
}
```

Функции (методы)

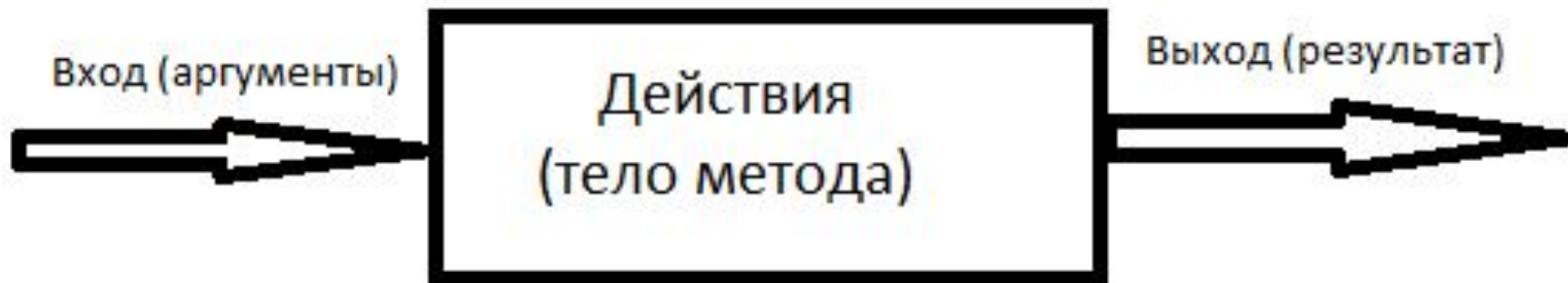
Программисты нуждаются в возможности разбить программу на несколько отдельных частей, чтобы она напоминала составной механизм, в котором одну деталь, можно заменить на другую, какие-то части временно вывести из строя и прикрутить новые элементы, добавляющие функциональность. В программировании эти составные части называются функциями.

Функция – часть кода, выполняющая определенные действия.

Функции используются либо чтобы не писать один и тот же код много раз, либо для решения специфических задач (например, блок кода, который выводит сообщения в определенной последовательности).

В общем виде:

```
тип модификаторы название_функции (параметры функции)
{
    // тело функции
}
```

Если работаем в консоли, то используем `static`

Что возвращает функция
если ничего, то пишем `void`

Название функции

Тип входного параметра

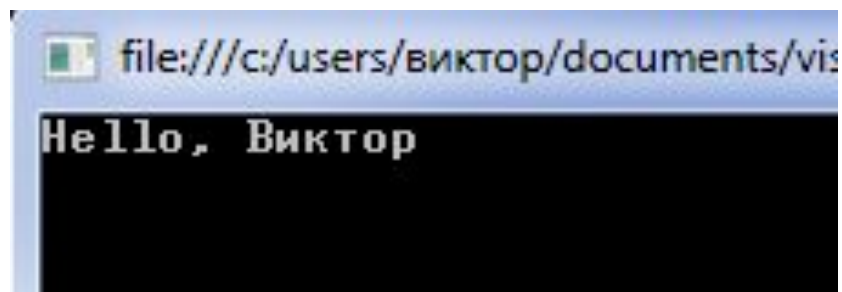
Имя параметра

```
static void SayHello(string name)  
{  
    Console.WriteLine("Hello, " + name);  
}
```

Тело функции (метода)

```
namespace ConsoleApplication50
{
    class Program
    {
        Описание функции
        static void SayHello(string name)
        {
            Console.WriteLine("Hello, " + name);
            Console.ReadLine();
        }

        static void Main(string[] args)
        {
            Вызов функции
            SayHello("Виктор");
            параметр
        }
    }
}
```



A screenshot of a Windows console window. The title bar shows the file path: file:///c:/users/виктор/documents/vis... The console output displays the text "Hello, Виктор" in a white monospace font on a black background.

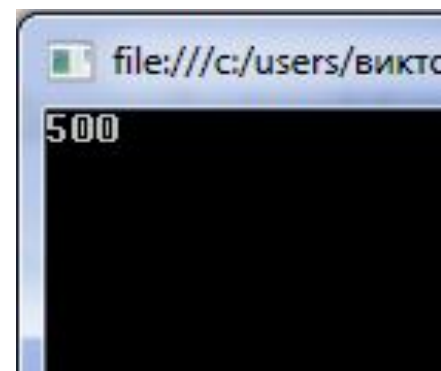
```
namespace ConsoleApplication50
{
    class Program
    {
        static Int64 Umnozhenie (Int32 a, Int32 b)
        {
            Int32 c;
            c = a * b;
            return c;
        }

        static void Main(string[] args)
        {
            Console.WriteLine(Umnozhenie(25, 20));
            Console.ReadLine();
        }
    }
}
```

Описание функции

Команда для возврата результат

Вызов функции

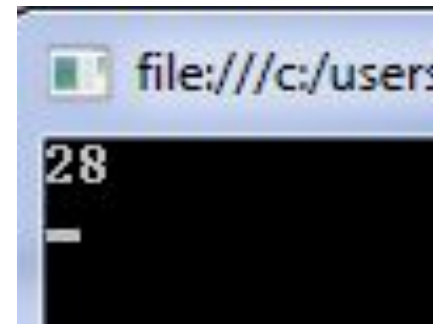


Практика

Написать функцию, которая по введенному году рождения вычисляет сколько лет пользователю

```
static Int32 Age (Int32 Year_of_Birth)
{
    Int32 c;
    c = 2016 - Year_of_Birth;
    return c;
}
```

```
Console.WriteLine(Age(1988));
Console.ReadLine();
```



Домашнее задание

Вариант 1 - Написать генератор паролей. Пользователь вводит желаемую длину пароля и отмечает, нужны ли цифры в пароле.

Вариант 2 - Написать тест. Для задания вопросов написать функцию. Создать один массив для вопросов, ответов, правильных ответов.

Вариант 3 - Написать текстовый квест. Использовать функции для упрощения кода программы, массивы для хранения характеристик игроков и врагов, а также состояния комнат или других локаций.