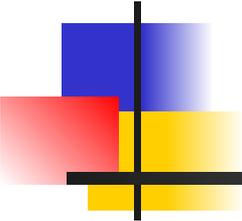


# Модуль 5

---

## Массивы

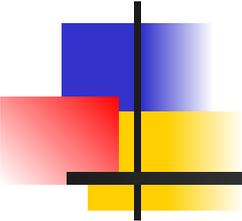


# Обзор модуля

---

В этом модуле вы изучите

- Введение в массивы
- Типы массивов
- Класс Array



# Занятие 1 - Введение в массивы

---

На первом занятии, **Введение в массивы**, вы изучите:

- Определение массива и его назначение.
- Синтаксис объявления массивов.

# Назначение

- Массивы  
размещают  
значения  
массива



**Array of 100 Names**

**Efficient Memory Utilization**

ЭНИЙ,  
И, И ЭТИ  
) ИМЕНИ

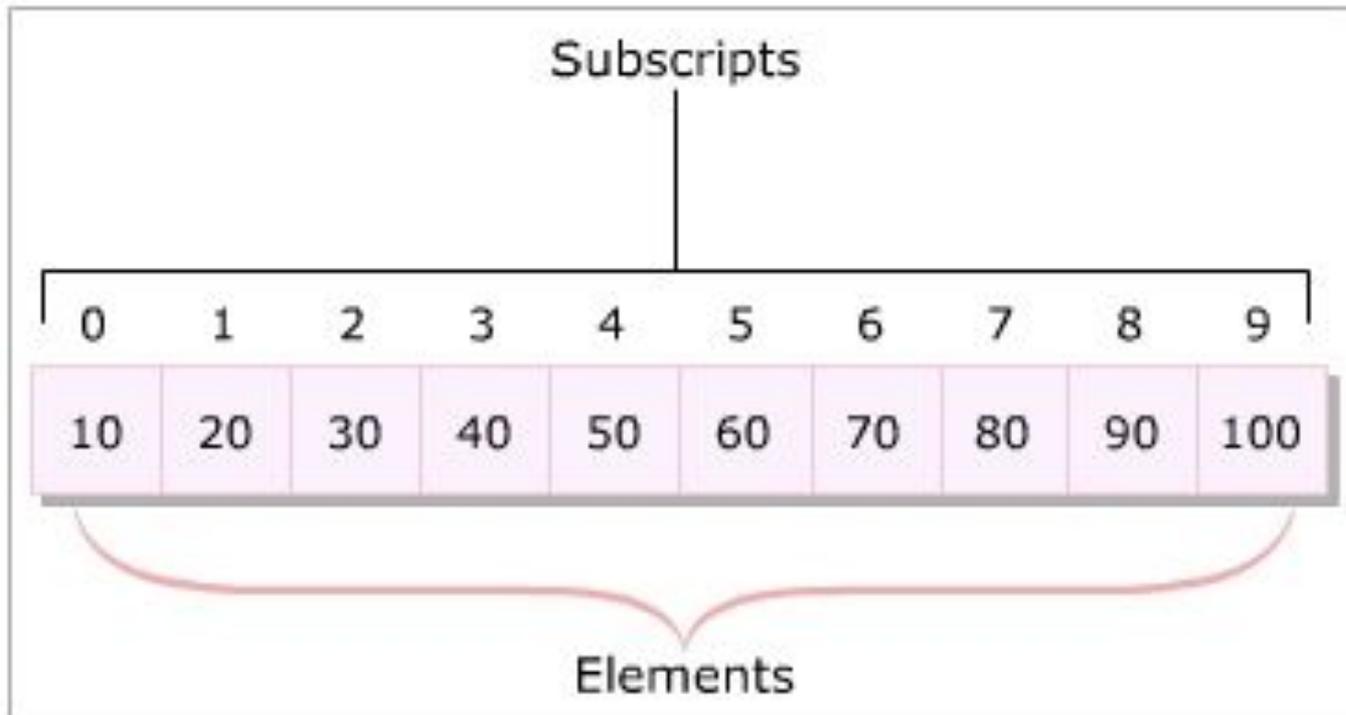
- Это упорядоченный набор значений.

```
//Program to store 100 names of students
string studentOne = "Jack Anderson";
string studentTwo = "Kate Jones";
string studentThree = "Francis Diaz";
string studentFour = "Glen Daniel";
string studentFive = "Frank James";
...
...
... Till 100 variables
```

**100 Variables Storing Names**

**Inefficient Memory Utilization**

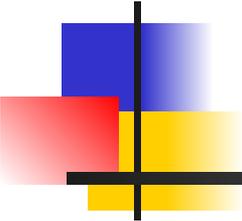
# Определение



ый

## Arrays

эффективном хранении, легкой сортировке и простому определению длины данных.



# Объявление массивов

## Синтаксис

- это переменные ссылочного типа,

```
type[] arrayName;
```

где,

type: указывает тип данных элементов массива ( например, int и char).

- arrayName: определяет имя массива.

который он будет хранить и идентификатор.

- Объявление массива не выделяет для него памяти.

# Инициализация массивов

Вывод

Значения по

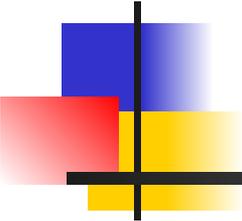
ключевого

```
class Numbers
{
    static void Main(string[] args)
    {
        int[] count = new int[10]; //создание массива
        int counter = 0;
        for(int i = 0; i < 10; i++)
        {
            count[i] = counter++; //присваивание значений
            элементам
            Console.WriteLine("The count value is: " + count[i]);
            //вывод значений элементов
        }
    }
}
```

val1: значение первого элемента.

valN: значение n-го элемента.

некоторых широко используемых типов данных.



# Занятие 2 - Типы массивов

---

На втором занятии, **Типы массивов**, вы изучите:

- Объяснение одномерных массивов.
- Объяснение многомерных массивов.
- Объяснение неровных массивов и их использование.
- Объяснение, как проходить циклом по массивам, используя foreach- цикл.

# Одномерные массивы

## Вывод

Элементы одномерного массива сохраняются  
одной строкой в выделенной памяти

```
James  
Alex  
Fernando
```

где,

`type`: Тип переменной, сопровождаемый квадратными скобками  
(`[]`).

`arrayName`: определяет имя переменной

`length`: Указывает число элементов в объявляемом массиве.

`new`: Создает экземпляр массива.

```
}
```

```
}
```

# Многомерные массивы

## Вывод

```
0 1 2 3 4
5 6 7 8 9
10 11 12 13 14
```

где,

type: Тип данных, сопровождаемый [].  
arrayName: определяет имя массива.  
value1: Указывает число строк.  
value2: Указывает число столбцов.

- Это многомерный массив, в котором одно из указанных измерений может иметь различающиеся значения.
- Он может иметь неравное число столбцов для каждой строки.

# Неровные массивы

## Вывод

массив - это многомерный массив,

```
List of companies in group 1: Intel AMD
List of companies in group 2: IBM Microsoft Sun
List of companies in group 3: HP Canon Lexmark Epson
```

```
string[][] companies = new string[3][];
companies[0] = new string[] {"Intel", "AMD"};
companies[1] = new string[] {"IBM", "Microsoft", "Sun"};
companies[2] = new string[] {"HP", "Canon", "Lexmark", "Epson"};
for (int i=0; i<companies.GetLength (0); i++)
{
    Console.Write("List of companies in group " + (i+1) + ":\t");
    for (int j=0; j<companies[i].GetLength (0); j++)
    {
        Console.Write(companies [i][j] + " ");
    }
    Console.WriteLine();
}
}
```

**Неровный массив**

# Использование цикла "foreach" для массивов

## Вывод

foreach в C# - это расширение цикла for.

```
Congratulations!! Ashley you have been granted an extra  
leave
```

```
Congratulations!! Joe you have been granted an extra leave  
Congratulations!! Mikel you have been granted an extra  
leave
```

где,

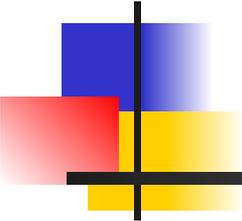
type: Тип переменной.

identifier: Имя переменной.

list: Имя переменной массива.

```
studentname + " you have been granted an extra leave ",  
    }  
    }  
}
```

элемента массива.

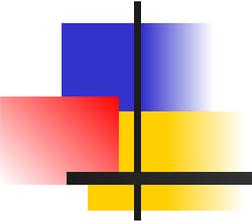


# Занятие 3 - Класс Array

---

На последнем занятии, **Array класс**, вы изучите:

- Описание класса `Array` и его назначение.
- Список обычно используемых свойств и методов класса `Array`.
- Объяснение, как конструировать массив с помощью класса `Array`.



# Класс "Array"

---

- Это встроенный класс в пространстве имен System и базовый класс для всех массивов в C#.
- Он обеспечивает методы для таких различных задач, как:
  - создание
  - поиск
  - копирование
  - сортировка массивов

# Свойства и методы

Методы	Описания
Clear	Удаляет все элементы в массиве и устанавливает его размер заданный размер или нет. Значение по умолчанию true.
CopyTo IsReadOnly	Копирует все элементы текущего одномерного массива в другой одномерный массив, начиная с указанной позиции. Значение по умолчанию false.
GetLength	Возвращает число элементов в массиве.
IsSynchronized	Возвращает true, если массив синхронизирован, иначе false.
GetLowerBound	Возвращает нижнюю границу массива.
GetUpperBound	Возвращает верхнюю границу массива.
Length	Возвращает количество элементов в массиве.
Length	Возвращает общее число элементов массива.
SetValue	Устанавливает указанное значение в заданной индексом позиции.
Rank	Возвращает число измерений массива.
GetValue	Возвращает значение в заданной индексом позиции.
SyncRoot	Возвращает объект, использующийся для синхронизации доступа к массиву.

# Использование класса "Array"

- **Пример** `Array` позволяет создавать массивы с помощью метода `CreateInstance()`

```
public static Array CreateInstance(Type elementType, int length1, int length2)
```

- где,

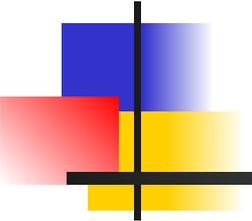
`length1`: Указывает длину строки.

`length2`: Указывает длину столбца.

`elementType`: Результирующий тип после приведения.

`Length`: Указывает длину массива.

```
objArray.SetValue("Business Administration", 4);  
for (int i = 0; i <= objArray.GetUpperBound(0); i++)  
{  
    Console.WriteLine(objArray.GetValue(i));  
}  
}
```



# Резюме

---

- **Массивы**

- Массивы - коллекции значений одного типа данных.

- **Типы массивов**

- Есть два типа массивов в C# - одномерные и многомерные.

- **Класс `Array`**

- Класс `Array` обеспечивает методы и свойства для создания, поиска и сортировки массивов.