

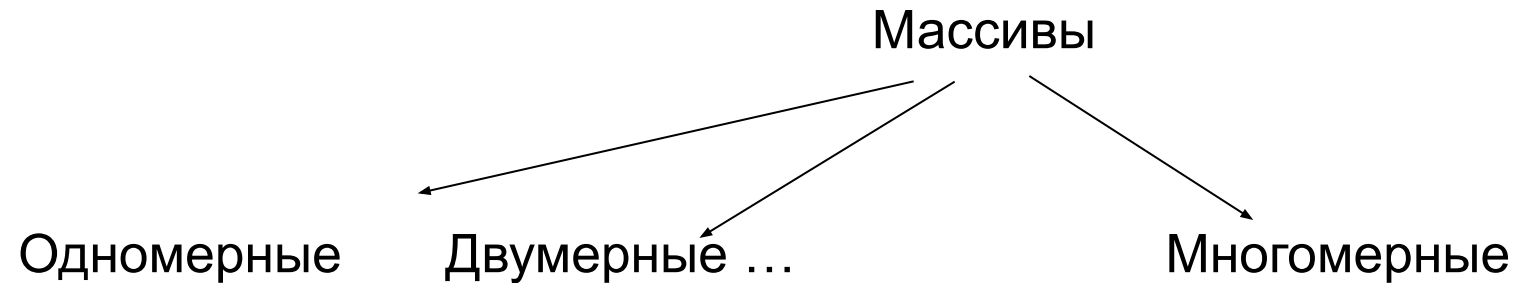
# МАССИВЫ

**Массивы** – это набор переменных, имеющих одно имя , один тип данных и расположенных в памяти последовательно друг за другом.

Каждая переменная массива называется **элементом массива**.

- Массивы используются для хранения ряда значений в одной переменной.

**Нумерация элементов массива** начинается с **0**



Чаще всего используются одномерные и двумерные массивы

# Массивы



```
graph TD; A[Массивы] --> B[Статические]; A --> C[Динамические];
```

Статические

Динамические

Статические массивы содержат **фиксированное количество** элементов, которое задается при описании массива

В динамических массивах **число элементов может меняться** при выполнении одной программы

# Описание массивов

При описании массивов используются те же ключевые слова **Dim**, **Static**, **Private**, **Public**, как при описании обычных переменных

**Dim Имя(N) As <тип данных >** , где N - размерность

**Dim Имя(N,M) As <тип данных >** , где N - количество строк, M - количество столбцов

Нумерация элементов массивов начинается с 0

<b>Dim A(12) As Integer</b>	определяет элементы A(0), A(1), A(2).....A(12)
<b>Static A(12) As Integer</b>	
<b>Dim B(3,3) As Integer</b>	определяет элементы B(0,0), B(0,1), B(0,2), B(0,3),B(1,0), B(1,1) .... B(3,3)
<b>Dim C(1 To 10)</b>	определяет элементы C (1), C (2),..... ,C (10)
<b>Dim D(1 To 3, 1 To 3)</b>	определяет элементы D(1,1), B(1,2), B(1,3), B(2,1),B(2,2), .... B(3,3)

# Заполнение массива

## 1. Оператором присваивания

**Dim Фрукты (3) As String**

‘ Здесь 3 - максимальный номер элемента

**Фрукты(0)="Яблоки"**

**Фрукты(1)="Груши"**

**Фрукты(2)="Ананасы"**

**Фрукты(3)="Персики"**

## 2. В цикле

**Dim I As Integer**

**Dim N(10) As Integer**

**Randomize**

**For I= 0 To 10**

**N(I)=Int(Rnd\*10+1)**

**Next I**

# Пример заполнение двумерного массива

```
Private Sub CommandButton1_Click()  
Dim multiarray(1, 1) As String  
multiarray(0, 0) = "Здание1, Помещение1"  
multiarray(0, 1) = "Здание1, Помещение2"  
multiarray(1, 0) = "Здание2, Помещение1"  
multiarray(1, 1) = "Здание2, Помещение2"  
End Sub
```

Представление массива в памяти

Строка	Столбец 0	Столбец 1
0	Здание1, Помещение1	Здание1, Помещение2
1	Здание2, Помещение1	Здание2, Помещение2

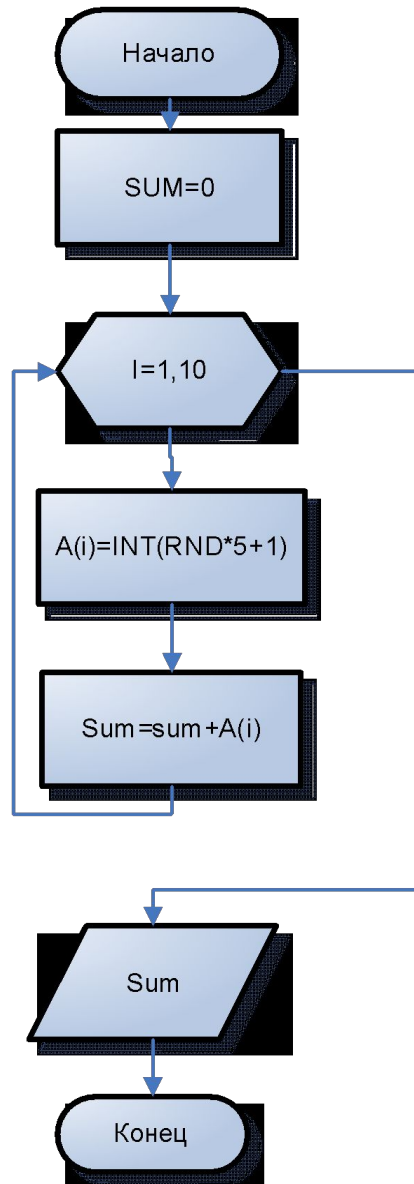
# Пример заполнения и вывода массива

```
Public a(10) As Integer
```

```
Private Sub CommandButton1_Click()  
Dim i As Integer  
For i = 0 To 10  
a(i) = Int(rnd*20)+1  
Next i  
End Sub
```

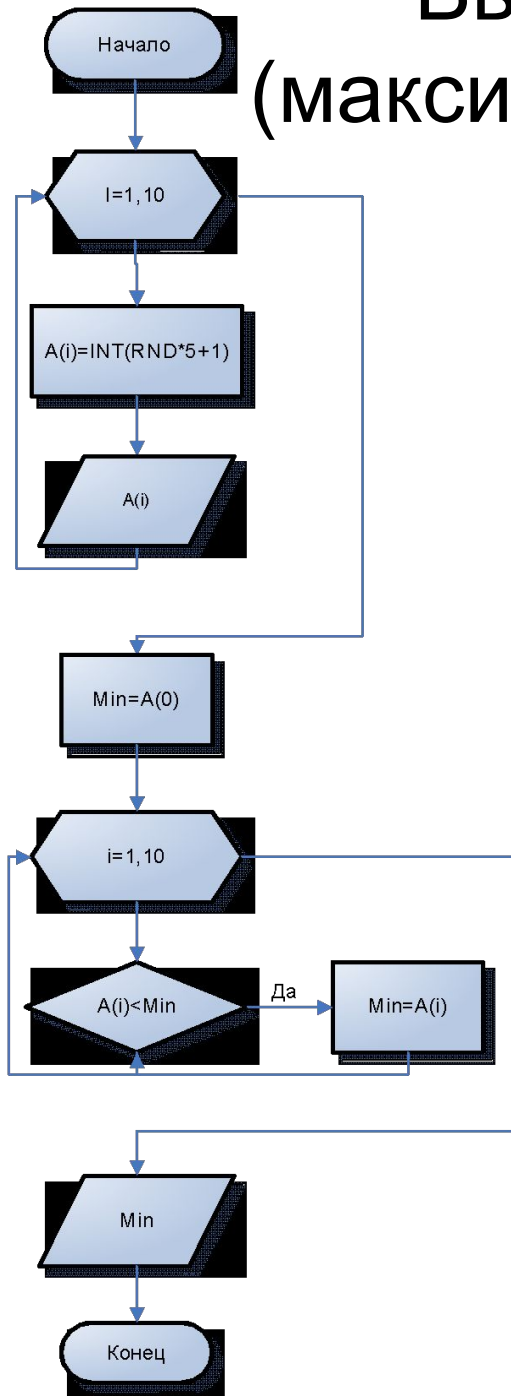
```
Private Sub CommandButton2_Click()  
Dim i As Integer  
Dim strd As String  
For i = 0 To 10  
strd = strd & " " & a(i)  
Next i  
MsgBox strd  
End Sub
```

# Вычисление суммы элементов массива

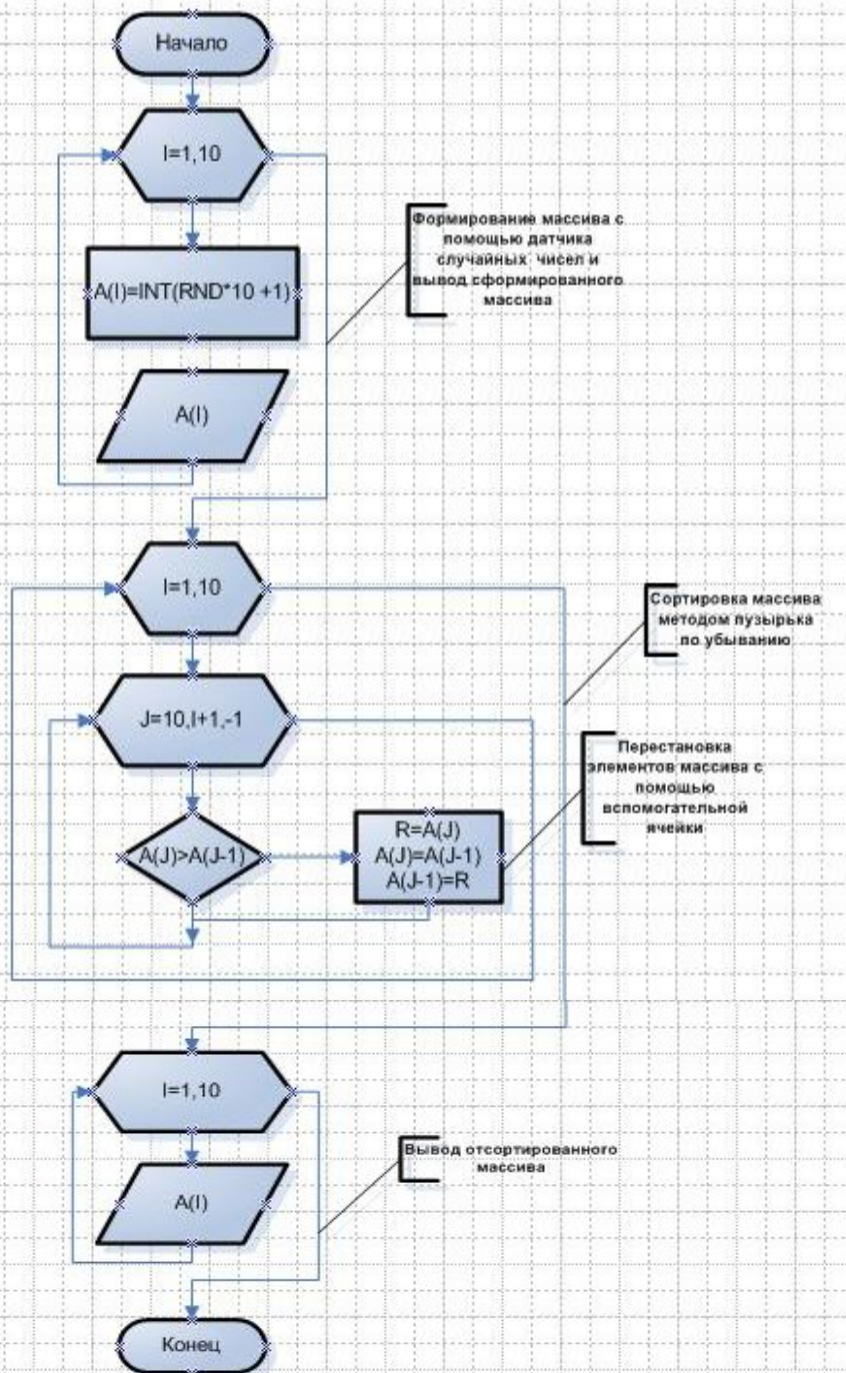




# Вычисление минимального (максимального) элемента массива



# Сортировка массива (метод пузырька)



# Пример сортировки методом «Пузырька»

25	1	7	3	4	10
1	25	7	3	4	10
1	7	25	3	4	10
1	7	3	25	4	10
1	3	7	4	25	10
1	3	4	7	25	10
1	3	4	7	10	25

25 1 7 3 4 10

i= 0

i= 1

j= 1

25 1

1 25

1 25 7 3 4 10

1 25 7 3 4 10

i= 2

j= 2

25 7

7 25

1 7 25 3 4 10

1 7 25 3 4 10

i= 3

j= 3

25 3

3 25

1 7 3 25 4 10

j= 2

7 3

3 7

1 3 7 25 4 10

1 3 7 25 4 10

i= 4

j= 4

25 4

4 25

1 3 7 4 25 10

j= 3

7 4

4 7

1 3 4 7 25 10

1 3 4 7 25 10

i= 5

j= 5

25 10

10 25

1 3 4 7 10 25

Начальный массив

25 1 7 3 4 10

Отсортированный массив

1 3 4 7 10 25

# ДИНАМИЧЕСКИЕ МАССИВЫ

Бывает так, что заранее точный размер массива неизвестен, и необходимо иметь возможность изменять его размер во время исполнения программы. Чтобы изменить размер массива динамически, используется оператор **Redim**

По умолчанию, когда изменяется размер массива, все его элементы удаляются. Если необходимо **оставить текущие значения элементов массива**, то нужно использовать ключевое слово **Preserve** с оператором **Redim**, что и сделано в примере.

Для освобождения памяти, занимаемой динамическим массивом, можно использовать инструкцию **Erase**

# Пример динамического массива

Не указывается количество  
элементов

**Dim Фрукты ( ) As String**

**Redim Фрукты (3)**

Указание количества элементов

**Фрукты(0)="Яблоки"**

**Фрукты(1)="Груши"**

**Фрукты(2)="Ананасы"**

**Фрукты(3)="Персики"**

Переопределение количества  
элементов

**Redim Preserve Фрукты (4)**

**Фрукты (4)="Сливы "**

# Использование функции **Array** при заполнении динамического массива

Dim Фрукты

Фрукты = Array("Яблоки", "Груши", "Сливы", "Ананасы")

Dim MyWeek

MyWeek = Array("Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun")

```
Dim MyWeek, MyDay, stra
```

```
MyWeek = Array("Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun")
```

```
For i = 0 To UBound(MyWeek)
```

```
stra = stra & " " & MyWeek(i)
```

```
Next i
```

```
MsgBox stra
```

```
MyDay = MyWeek(2)
```

```
MsgBox "второй день недели " & MyDay
```

```
' MyDay contains "Tue".
```

```
MyDay = MyWeek(4)
```

```
' MyDay contains "Thu".
```

```
MsgBox "четвертый день недели " & MyDay
```

# Определение размера динамического массива

Для того, чтобы определить размер массива в любой момент, используются функции

- **LBound** – нижняя граница
- **UBound** – верхняя граница

```
Private Sub CommandButton1_Click()  
Dim MyWeek  
Dim MyDay, stra  
MyWeek = Array("Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun")  
  
For i = LBound(MyWeek) To UBound(MyWeek)  
stra = stra & " " & MyWeek(i)  
Next i  
MsgBox stra  
MyDay = MyWeek(2)  
MsgBox "второй день недели " & MyDay  
MyDay = MyWeek(4)  
MsgBox "четвертый день недели " & MyDay  
End Sub
```



# Использование цикла For ...Each

```
Private Sub CommandButton2_Click()  
Dim multiarray(1, 1) As String  
Dim vntelement As Variant  
  
multiarray(0, 0) = "Здание1, Помещение1"  
multiarray(0, 1) = "Здание1, Помещение2"  
multiarray(1, 0) = "Здание2, Помещение1"  
multiarray(1, 1) = "Здание2, Помещение2"  
For Each vntelement In multiarray  
MsgBox vntelement  
Next  
End Sub
```

# Массивы и элементы управления

Список (ListBox)

Поле(TextBox)

The screenshot shows a Windows form titled "UserForm1" with a green background. On the left side, there is a large, empty white rectangular area representing a list box. To the right of this area, there are several buttons and text boxes arranged in a grid-like fashion. The buttons are labeled: "Заполнение Списка" (Fill List), "Очистка списка" (Clear List), "Поиск MAX" (Find MAX), "Вычисление суммы элементов" (Calculate sum of elements), "Вычисление суммы выделенных элементов" (Calculate sum of selected elements), and "Выход" (Exit). There are three empty white rectangular areas representing text boxes, located to the right of the "Очистка списка", "Вычисление суммы элементов", and "Вычисление суммы выделенных элементов" buttons. At the bottom of the form, there is a dark green button labeled "Сортировка" (Sort).

# Свойства списка ComboBox и поля со списком ListBox

ListIndex	Номер текущего элемента списка
ListCount	Число элементов списка
ColumCount	Число столбцов в списке
Text	Выбранный в списке элемент
List	Элемент на пересечении строки и столбца List(1,2)
RowSource	Диапазон, содержащий элементы списка
MultiSelect	Только для ListBox
Selected	Только для ListBox
ColumnWidths	Ширина столбцов списка With ListBox1 .ColumnCount=3 .ColumnWidths="20;30;30" End With
ListRows	Устанавливает число элементов , отображаемых в раскрывающем списке (ComboBox)

# Методы

<b>Clear</b>	Удаляет все элементы из списка
<b>RemoveItem</b>	Удаляет, указанный элемент RemoveItem (5)
<b>AddItem</b>	Добавляет элемент