

# Двумерные массивы

## Сегодня вы научитесь:

- Заполнять числовой массив.
- Выводить массив на экран.
- Работать с двумерными массивами.
- Искать максимальный элемент в одномерном и двумерном массиве.

В языках программирования подобные типы данных описываются с помощью массивов.

# Массив

это набор однотипных данных, к которым можно обратиться с помощью единственного имени

Каждое значение в массиве называется **элементом**.

Номер элемента в списке называется **индексом**.

Элементы представляют собой символьные или числовые значения, которые можно использовать в выражениях ПАСКАЛЬ

```
a = array[1..5,1..10] of integer;
```

- Общий вид двумерного массива следующий:

```
имя = array[строки,столбцы] of тип;
```

где:

- **имя** — имя массива;
  - **строки** — диапазон строк в массиве;
  - **столбцы** — диапазон столбцов в массиве;
  - **тип** — определяет тип элементов массива.
- Наглядно представить двумерный массив можно с помощью следующего примера. Представьте зал кинотеатра, в котором 7 рядов по 10 мест в каждом. Этот кинотеатр можно представить двумерным массивом и объявить следующим образом:

```
кинотеатр = array [1..7,1..10] of зритель;
```

Массивы часто используются для хранения и обработки большого количества однотипных данных, причём не только в графике. Представим, что необходимо в некотором довольно большом множестве чисел отыскать наибольшее. Пусть, например, вы — директор детского спортивного лагеря и вам совершенно необходимо срочно узнать, кто из новоприбывших детей самый высокий... Представили? Тогда за работу — составим программу, не последний же сезон вы директорствуете, и программка вам ещё не раз пригодится.

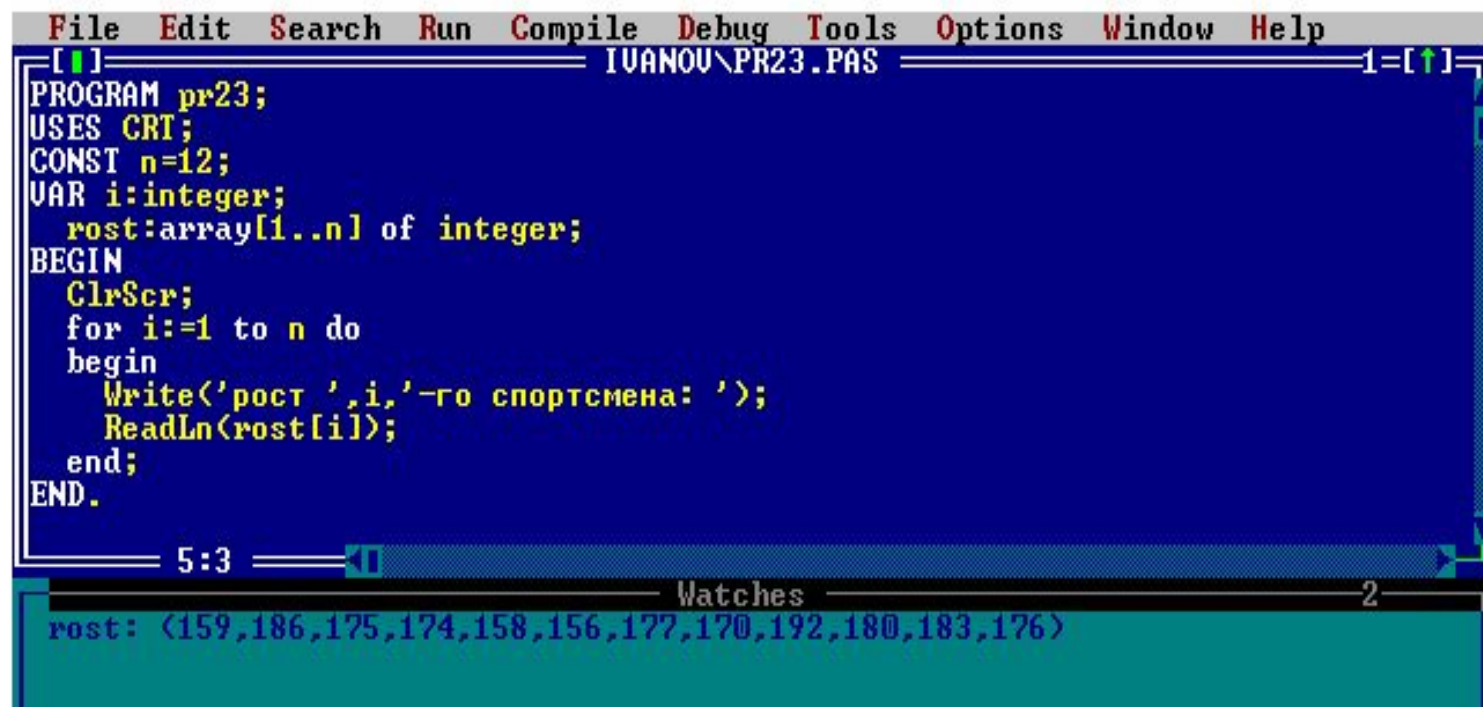
Итак, мы встречаем детей, которые по приезду регистрируются, сообщая свои данные, в том числе и рост. Составим программу, принимающую значения роста каждого спортсмена и сохраняющую их в массиве.

```
PROGRAM pr23;
USES CRT;
CONST
    n=12;
VAR
    i:integer;
    rost:array[1..n] of integer;
BEGIN
    ClrScr;
    for i:=1 to n do
    begin
        Write('рост ',i,'-го спортсмена: ');
        ReadLn(rost[i]);
    end;
```

Где же введённые нами значения роста?

Они сохранились в переменной `rost`, и это можно увидеть в окне отладки.

- Установите курсор на имя переменной `rost`, нажмите аккорд `Ctrl+F7`, а затем в окне **Add Watch** кнопку **OK**:



The screenshot shows a Pascal IDE window titled "IUANOU\PR23.PAS". The code in the editor is as follows:

```
PROGRAM pr23;  
USES CRT;  
CONST n=12;  
VAR i:integer;  
    rost:array[1..n] of integer;  
BEGIN  
    ClrScr;  
    for i:=1 to n do  
    begin  
        Write('рост ',i,'-го спортсмена: ');  
        ReadLn(rost[i]);  
    end;  
END.
```

At the bottom of the IDE, there is a "Watches" window with the following content:

```
rost: <159,186,175,174,158,156,177,170,192,180,183,176>
```

Как вы понимаете, нашу программу мы сегодня будем усложнять, дополнять новыми командами, и при этом запускать, запускать, запускать... И каждый раз придётся вводить эти 12 значений?

Нет, конечно. На этапе тестирования программы программисты обычно заполняют массив случайными числами, что мы сейчас и сделаем:

- Измените программу:



```
BEGIN

  randomize;

  ClrScr;

  for i:=1 to n do

  begin

    Write('пост ', i, ' ро спортсмена: ');

    ReadLn(rost[i]);

    rost[i] := random(40) + 150;

  end;
```

Для вывода массива на экран достаточно так же в цикле выполнить команду `Write(rost[i])`.

```
CONST
```

```
    n=12;
```

```
VAR
```

```
    i:integer;
```

```
    rost:array[1..n] of integer;
```

```
BEGIN
```

```
    randomize;
```

```
    ClrScr;
```

```
    for i:=1 to n do
```

```
    begin
```

```
        rost[i]:=random(40)+150;
```

```
        Write(rost[i]:5);
```

```
    end;
```

```
    ReadLn;
```

```
END.
```

- Запустите программу — теперь хорошо виден весь массив:

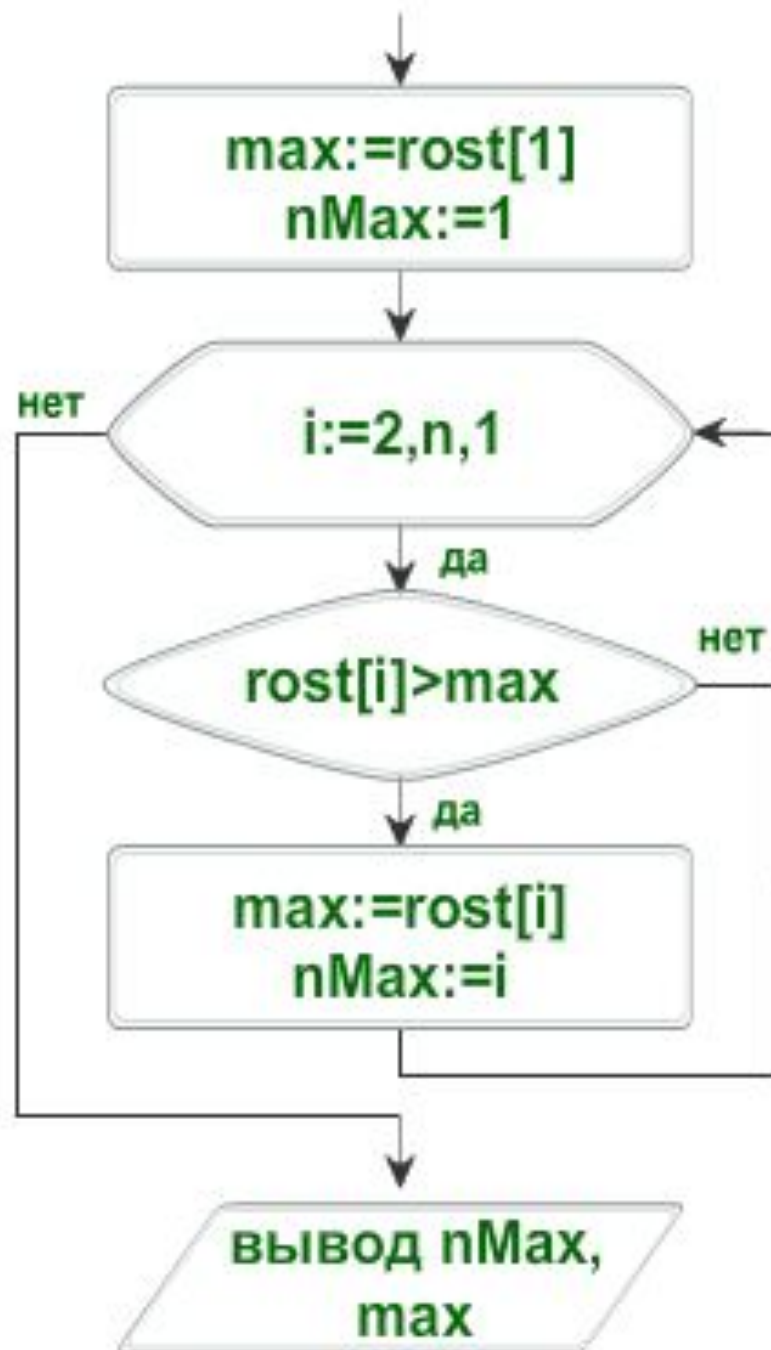
```
150 179 183 174 180 153 157 177 178 170 156 159
```

А теперь реализуем поиск самого высокого спортсмена, или, иначе говоря, самого большого числа в массиве. Как это сделать?

- сначала самым высоким (переменная `max`) считаем первого спортсмена (и это действительно так, он пока один — кто скажет, что есть кто-то выше его?);
- затем сравниваем рост второго спортсмена с `max` — если он больше, то помещаем его значение в `max`;
- последнее действие повторяем для всех последующих элементов массива.

**Замечание.** Переменная `nMax` содержит номер максимального элемента.

- Дополните программу поиском максимального элемента, не забудьте объявить новые переменные.



- Добавьте в команду вывода результата печать двух символов:

```
...  
WriteLn(#13,#10,'Самый высокий спортсмен под номером ',nMax,' имеет рост ',max,' см');  
...
```

Что это за символы?

Мы хорошо знаем, что при наборе текста клавиша  осуществляет так называемый «перевод строки», то есть, перемещает курсор в начало новой строки. Но, возможно, для вас будет новостью, что на самом деле клавиша  генерирует ввод двух символов:

- символ с номером **13 (#13)** переводит курсор к началу строки:

```
WriteLn<#13,'Самый высокий спортсмен под номером ',nMax,' имеет рост ',max,'  
ReadLn;  
END.
```

```
Самый высокий спортсмен под номером 4 имеет рост 188 см 179
```

Обратите внимание, текст результата «затёр» строчку с выведенным массивом, от массива остался небольшой «хвостик» — число 179.

о символ с номером 10 (#10) переводит курсор на следующую строку:

```
WriteLn(#10, 'Самый высокий спортсмен под номером ', nMax, ' имеет рост ', max, '  
ReadLn;  
END.
```

```
185 169 187 169 162 168 187 178 171 188 169 185  
Самый высокий спортсмен под номером 10 имеет рост 188 см
```

Программа, определяющая самого высокого в прибывшей группе, готова. Но ведь после первой придёт вторая, а затем третья... Вы предлагаете для каждой группы использовать программу, а потом просто выбрать из полученных «гигантов» самого-самого? Да, так можно поступить, но зачем же мы будем выполнять работу, которую может выполнить компьютер?

Итак, наша задача изменяется: в нашем распоряжении не один массив `rost`, а, например, пять: `rost1`, `rost2`, `rost3`, `rost4`, `rost5`. И мы теперь заполним все эти массивы, используя пять циклов, затем найдём в каждом максимальный элемент, затем среди пяти найденных чисел опять будем искать максимальный...

У вас уже вертится на языке вопрос:

— А нельзя ли массивы назвать не `rost1`, `rost2`, ..., а `rost[1]`, `rost[2]`, ..., тем самым создав из них новый массив (массив массивов)?

- о вместе же эти два символа «выполняют работу» клавиши `Enter`:

```
WriteLn(#13,#10,'Самый высокий спортсмен под номером ',nMax,' имеет рост ',m);
ReadLn;
END.
```

```
172 181 166 157 166 189 187 180 164 162 160 156
Самый высокий спортсмен под номером 6 имеет рост 189 см
```



Измените программу (одновременно читая комментарии к изменениям):

```
PROGRAM pr23;  
  
USES CRT;  
  
CONST
```

- о константа **m** содержит размерность нового «супермассива»:

```
    n=12; m=5;  
  
VAR
```

- о счётчиком групп будет переменная **j**, а в переменной **mMax** будем хранить номер группы с самым высоким спортсменом:

```
i,j:integer;

max,nMax,mMax:integer;
```

о теперь переменная **rost** — это массив массивов:

```
rost:array[1..m] of array[1..n] of integer;

BEGIN

  randomize;

  ClrScr;
```

внешний цикл по **j** перебирает строки массива, внутренний же цикл (по **i**) перебирает элементы текущей строки:

```
for j:=1 to m do
begin
  for i:=1 to n do
  begin
    rost[j,i]:=random(40)+150;
    Write(rost[j,i]:5);
  end;
  WriteLn;
end;
```

```
max:=rost[1,1]; nMax:=1; mMax:=1;
```

```
for j:=1 to m do
```

```
  for i:=1 to n do
```

```
    if rost[j,i]>max
```

```
      then
```

```
        begin
```

```
          max:=rost[j,i];
```

```
          nMax:=i;
```

```
          mMax:=j;
```

```
        end;
```

```
WriteLn(#13,#10,'Самый высокий спортсмен под номером ',nMax,' из группы ',mMax,' имеет рост ',max,' см');
```

```
ReadLn;
```

```
END.
```

Запустите программу:

```
159 156 162 165 157 168 180 153 170 175 185 183
157 186 185 187 172 171 170 172 189 156 160 166
163 177 180 172 167 161 151 186 184 168 162 178
158 169 157 162 167 168 157 169 170 164 177 186
151 153 179 159 172 162 152 185 151 187 171 156
```

Самый высокий спортсмен под номером 9 из группы 2 имеет рост 189 см

Действительно, девятое число во второй строке равно **189** — максимум!

Массив массивов используется в программировании настолько часто, что он получил собственное название **двумерный массив**, более того, вместо объявления вида `rost:array[1..m] of array[1..n] of integer` можно написать более короткое и логичное: `rost:array[1..m,1..n] of integer`.

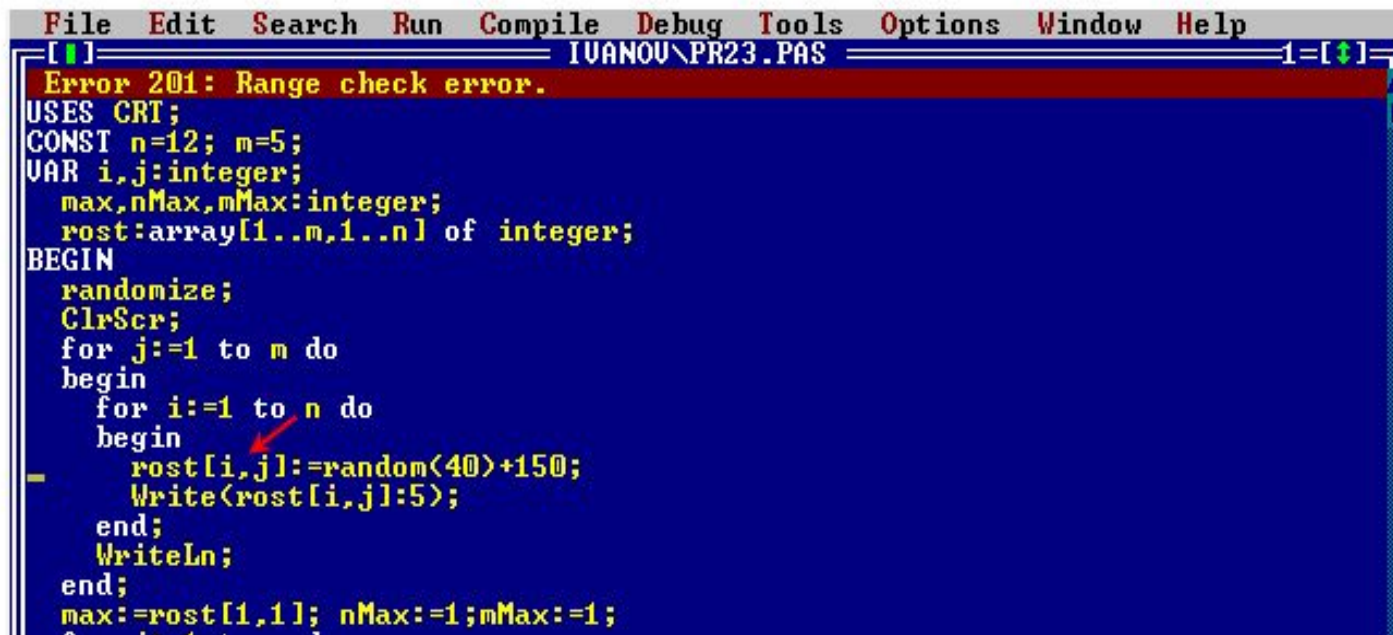
- Измените объявление массива `rost`:

```
PROGRAM pr23;
USES CRT;
CONST
  n=12; m=5;
VAR
  i,j:integer;
  max,nMax,mMax:integer;
  rost:array[1..m,1..n] of integer;
```

# Типичные ошибки

При переборе элементов двумерного массива будьте очень внимательны с индексами. Если размерности внешнего и внутреннего массивов разные, возникнет **runtime**-ошибка выхода за пределы массива, если же размерности одинаковые, то ошибки **runtime** не возникнет, но результаты работы программы могут оказаться неверными.

В примере индексы **i** и **j** переставлены местами: `rost[i, j] := random(40) + 150;`



```
File Edit Search Run Compile Debug Tools Options Window Help
I\ANOV\PR23.PAS 1=[↑]
Error 201: Range check error.
USES CRT;
CONST n=12; m=5;
VAR i, j: integer;
    max, nMax, mMax: integer;
    rost: array[1..m, 1..n] of integer;
BEGIN
    randomize;
    ClrScr;
    for j:=1 to m do
    begin
        for i:=1 to n do
        begin
            rost[i, j] := random(40) + 150;
            Write(rost[i, j]:5);
        end;
        WriteLn;
    end;
    max := rost[1, 1]; nMax := 1; mMax := 1;
```

# Задание для практической работы:

Задача 1.

Написать программу, которая вычисляет *среднюю* за неделю температуру воздуха. Исходные данные должны вводиться во время работы программы.

Рекомендуемый *вид экрана* при работе программы:

**Введите температуру воздуха за неделю.**

**1 день= 2**

**2 день= 0**

**3 день= -3**

**4 день= 1**

**5 день= 3**

**6 день= -2**

**7 день= 0**

**Средняя температура: 0.00 град.**

# Задания для практической работы:

Написать программу, которая вводит с клавиатуры одномерный массив из 5 целых чисел и выводит количество ненулевых элементов. Перед вводом каждого элемента должна появляться подсказка с номером.

Рекомендуемый вид экрана при работе программы:

**После ввода каждого числа нажмите <ENTER>**

**A[1]= 12**

**A[2]= 0**

**A[3]= -3**

**A[4]= 1**

**A[5]= 0**

**В массива 3 ненулевых элемента**



## Задание 1

Измените программу, созданную на уроке, так, чтобы она искала минимальный (самый маленький) элемент:

```
157 189 158 184 152 152 161 186 183 167 170 159
189 183 161 171 164 164 164 171 156 187 181 182
157 157 175 163 166 152 179 163 166 170 184 165
161 162 171 187 159 165 186 160 171 160 172 159
162 186 170 170 157 177 177 162 186 177 171 164
```

Самый маленький спортсмен под номером 5 из группы 1 имеет рост 152 см

## Задание 2\*

Создайте программу, которая:

- создаёт одномерный массив из **10** элементов;
- заполняет его случайными двузначными числами;
- подсчитывает в массиве количество чисел, меньших **30**:

```
31 13 11 40 58 41 93 12 24 65
4 ч. меньше 30
```