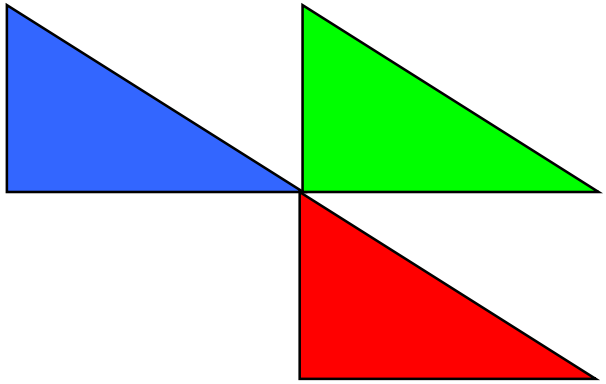
Программирование на языке Паскаль

Процедуры

Процедуры

Задача: Построить фигуру:



Можно ли решить известными методами?

Особенность: Три похожие фигуры.

общее: размеры, угол поворота

отличия: координаты, цвет

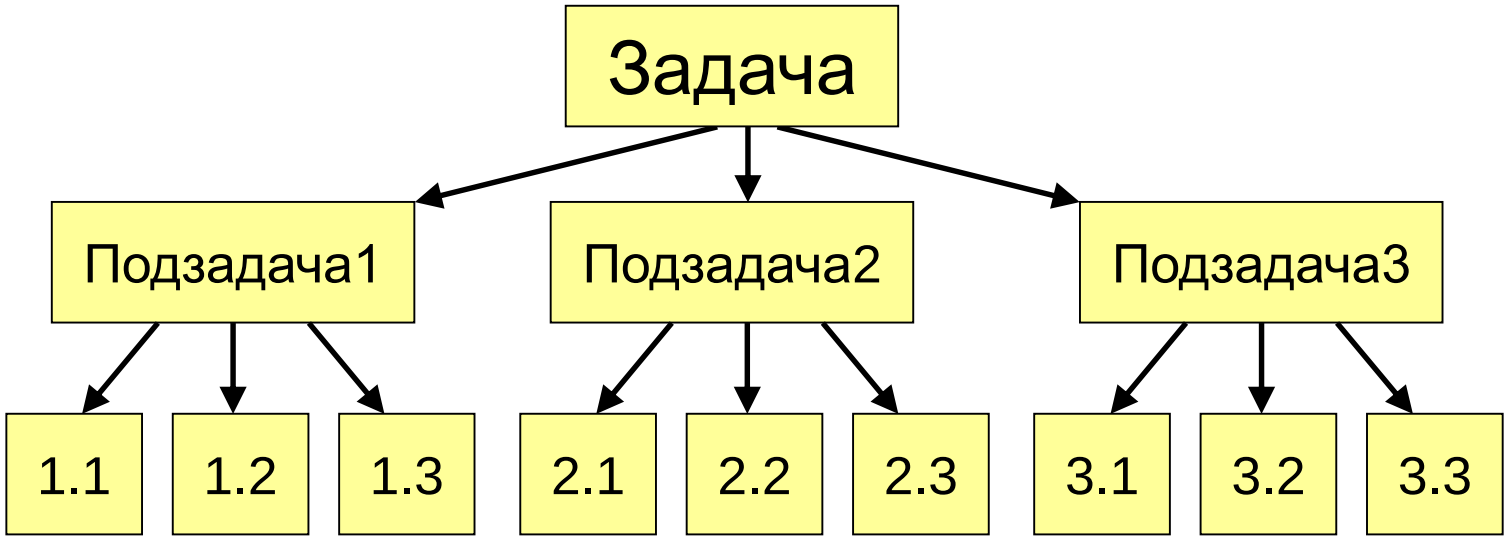
Сколько координат надо задать?

Процедуры

Процедура – это вспомогательный алгоритм, который предназначен для выполнения некоторых действий.

Применение:

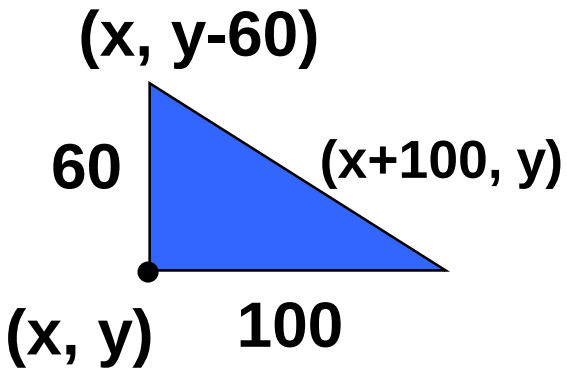
- выполнение одинаковых действий в разных местах программы
- разбивка программы (или другой процедуры) на подзадачи для лучшего восприятия



Процедуры

Порядок разработки:

- выделить одинаковые или похожие действия (три фигуры)
- найти в них **общее** (размеры, форма, угол поворота) и **отличия** (координаты, цвет)
- отличия записать в виде неизвестных переменных, они будут **параметрами** процедуры



заголовок

параметры

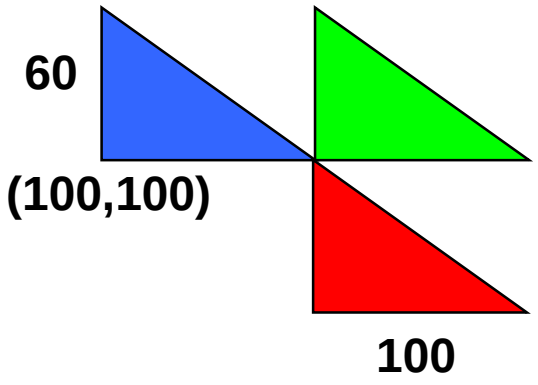
```
procedure Tr( x, y, r, g, b: integer);
begin
    MoveTo(x, y);
    LineTo(x, y-60);
    LineTo(x+100, y);
    LineTo(x, y);
    Brush(1, r, g, b);
    Fill(x+20, y-20);
end;
```

цвет

координаты

тело процедуры

Программа



формальные параметры

```
program qq;
  procedure Tr( x, y, r, g, b: integer);
  begin
    ...
  end;
begin
    Pen(1, 255, 0, 255);
    Tr(100, 100, 0, 0, 255);
    Tr(200, 100, 0, 255, 0);
    Tr(200, 160, 255, 0, 0);
end.
```

процедура

вызовы процедуры

фактические параметры

Процедуры

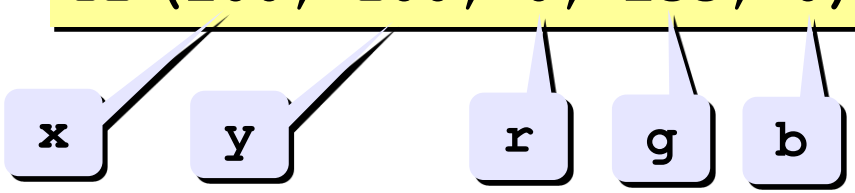
Особенности:

- все процедуры расположены **выше** основной программы
- в заголовке процедуры перечисляются **формальные** параметры, они обозначаются именами, поскольку могут меняться

```
procedure Tr( x, y, r, g, b: integer);
```

- при вызове процедуры в скобках указывают **фактические** параметры (числа или арифметические выражения) **в том же порядке**

```
Tr (200, 100, 0, 255, 0);
```



Процедуры

Особенности:

- для каждого формального параметра после двоеточия указывают его тип

```
procedure A (x: real; y: integer; z: real);
```

- если однотипные параметры стоят рядом, их перечисляют через запятую

```
procedure A (x, z: real; y, k, l: integer);
```

- внутри процедуры параметры используются так же, как и переменные

Процедуры

Особенности:

- в процедуре можно объявлять дополнительные **локальные** переменные, остальные процедуры не имеют к ним доступа

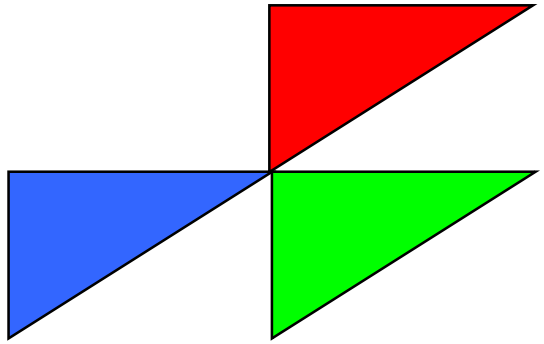
```
program qq;

  procedure A(x, y: integer);
    var a, b: real;
    begin
      a := (x + y)/6;
      ...
    end;
begin
  ...
end.
```

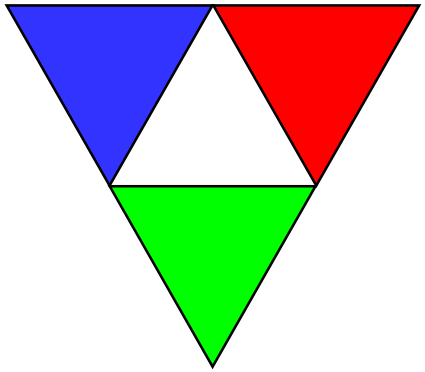
локальные переменные

Задания

«3»: Используя одну процедуру, построить фигуру.

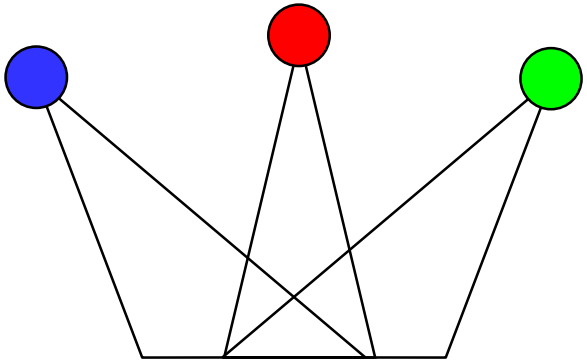


«4»: Используя одну процедуру, построить фигуру.



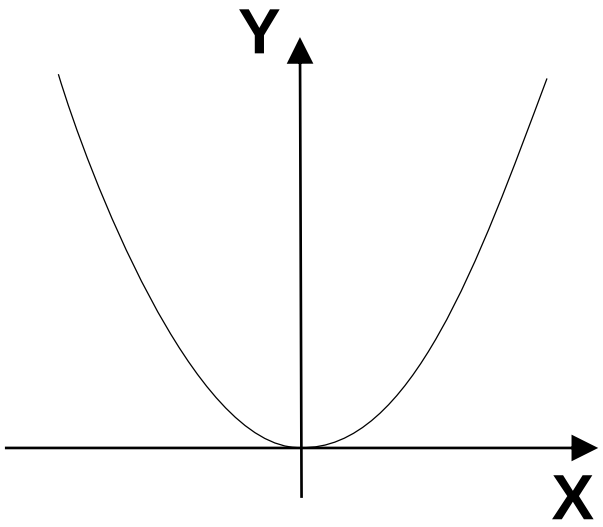
Задания

«5»: Используя одну процедуру, построить фигуру.



Построение графиков функций

Задача: построить график функции $y = x^2$ на интервале от -2 до 2.



Анализ:

максимальное значение

$y_{max} = 4$ при $x = \pm 2$

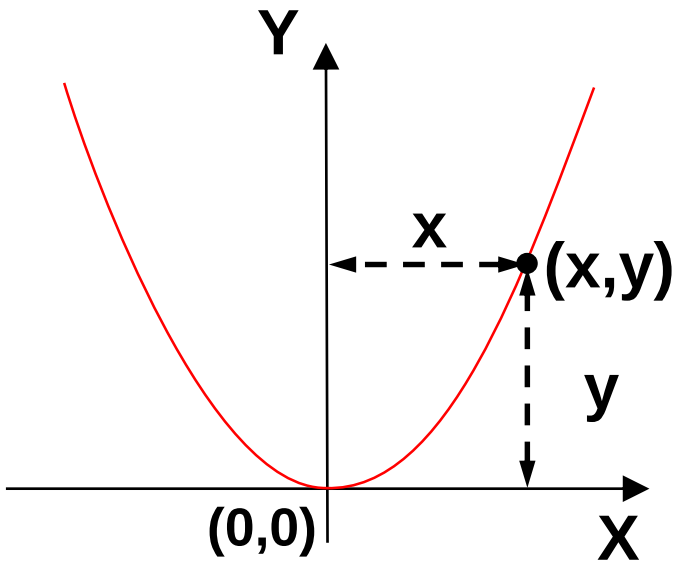
минимальное значение

$y_{min} = 0$ при $x = 0$

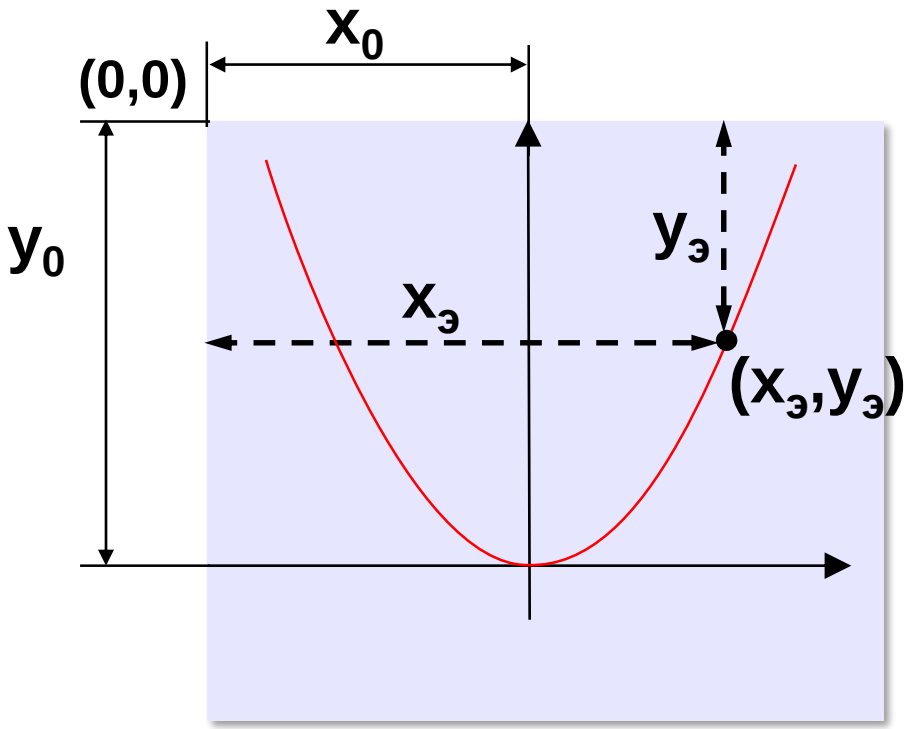
Проблема: функция задана в математической системе координат, строить надо на экране, указывая координаты в пикселях.

Преобразование координат

Математическая система координат



Экранная система координат (пиксели)



k – *масштаб* (длина изображения единичного отрезка на экране)

$x_э =$

$y_э =$

Программа

```
program qq;
const x0 = 150; y0 = 250; k = 50;
      xmin = -2; xmax = 2;
var x, y, h: real;
    xe, ye: integer;
begin
  Line(0, y0, x0+150, y0);
  Line(x0, 0, x0, y0+20);
  x := xmin; h := 0.02;
  while x <= xmax do begin
    y := x*x;
    xe := x0 + round(k*x);
    ye := y0 - round(k*y);
    Point (xe, ye);
    x := x + h;
  end;
end.
```

h – шаг изменения x

на экране

оси координат

цикл построения графика

? Что плохо?

Как соединить точки?

Алгоритм:

Если первая точка
  перейти в точку ($x_э$,$y_э$)
иначе
  отрезок в точку ($x_э$,$y_э$)

Программа:

```
var first: boolean;
    ...
begin
  ...
  first := True;
  while x <= xmax do begin
    ...
    if first then begin
       MoveTo(xe, ye);
       first := False;
    end
    else LineTo(xe, ye);
    ...
  end;
end.
```

логическая переменная

начальное значение

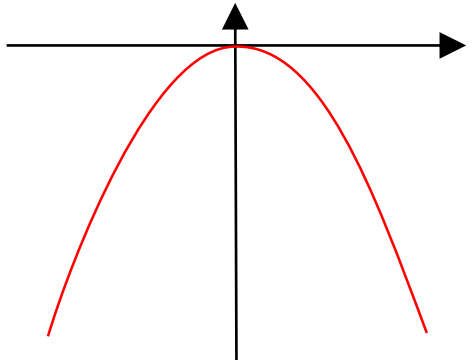
выбор варианта действий

Задания

«3»: Построить график функции

$$y = -x^2$$

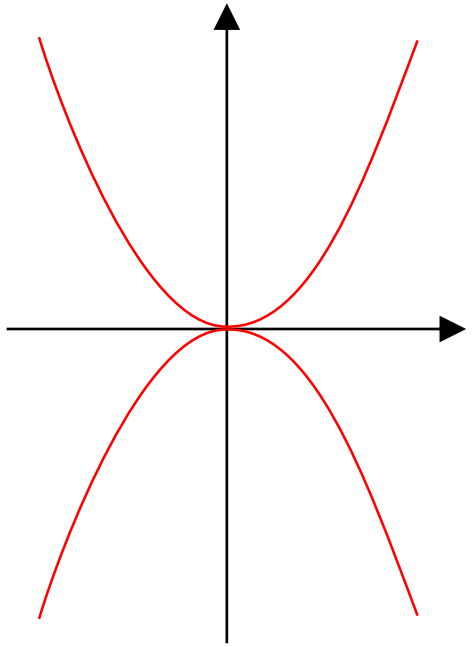
на интервале [-2,2].



«4»: Построить графики функций

$y = x^2$ **и** $y = -x^2$

на интервале [-2,2].

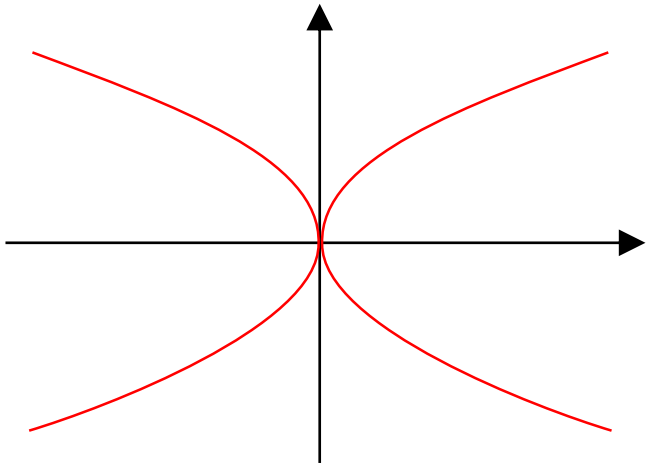


Задания

«5»: Построить графики функций

$x = y^2$ **и** $x = -y^2$

на интервале [-2,2].



Программирование на языке Паскаль

Функции

Функции

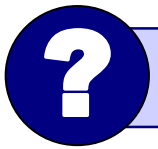
Функция – это вспомогательный алгоритм (подпрограмма), результатом работы которого является некоторое значение.

Примеры:

- **Вычисление модуля и других функций**
- **расчет значений по сложным формулам**
- **ответ на вопрос (простое число или нет?)**

Зачем?

- **для выполнения одинаковых расчетов в различных местах программы**
- **для создания общедоступных библиотек функций**

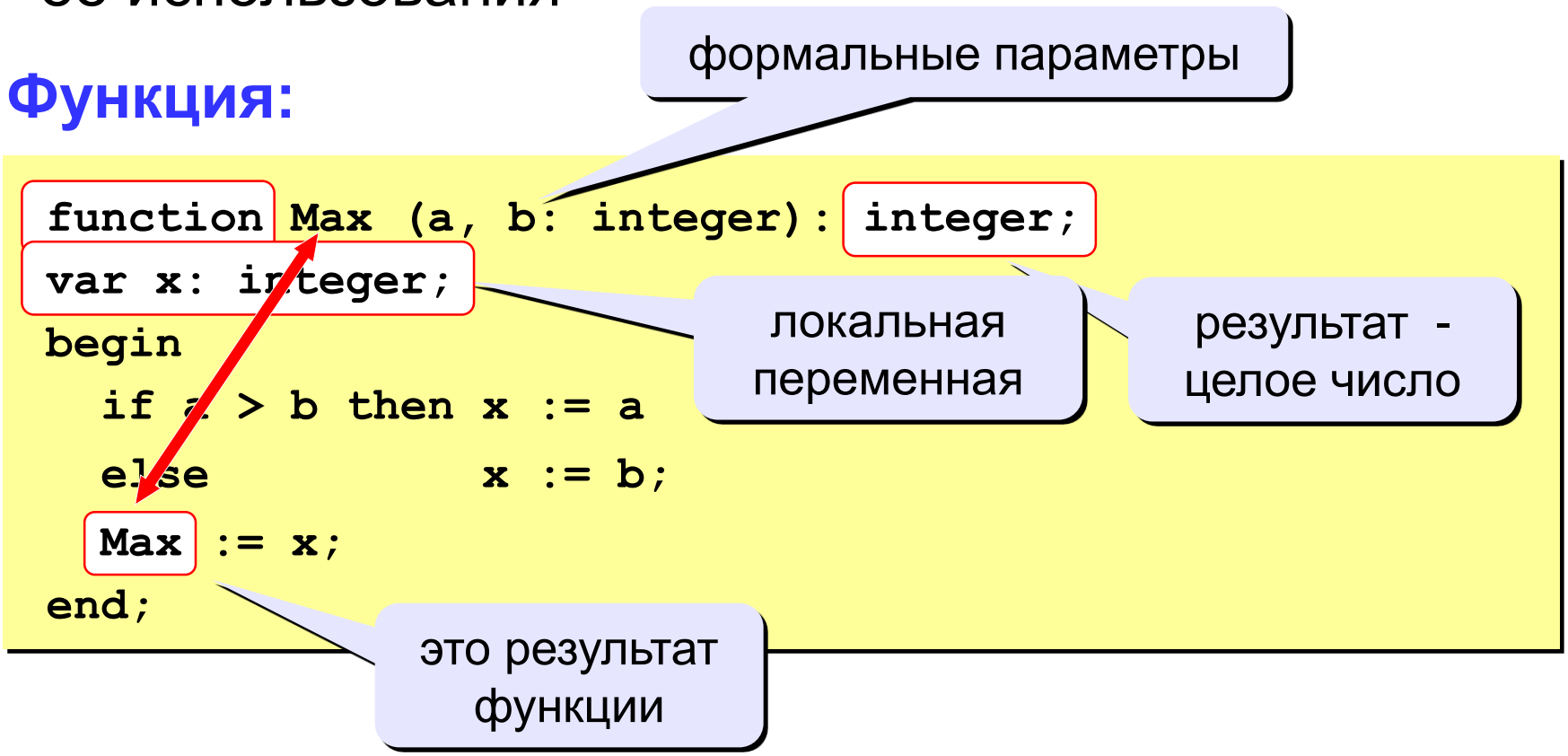


В чем отличие от процедур?

Функции

Задача: составить функцию, которая вычисляет наибольшее из двух значений, и привести пример ее использования

Функция:



Функции

Особенности:

- заголовок начинается словом **function**

```
function Max (a, b: integer): integer;
```

- формальные параметры описываются так же, как и для процедур

```
function qq( a, b: integer; x: real ): real;
```

- в конце заголовка через двоеточие указывается **тип результата**

```
function Max (a, b: integer): integer;
```

- функции располагаются **ВЫШЕ** основной программы

Функции

Особенности:

- можно объявлять и использовать **локальные переменные**

```
function qq (a, b: integer): integer;
var x, y: real;
begin
   ...
end;
```

- значение, которое является результатом, записывается в переменную, **имя** которой совпадает с **названием функции**; объявлять ее **НЕ НАДО**:

```
function Max (a, b: integer): integer;
begin
   ...
   Max := a;
end;
```

Программа

```
program qq;
var a, b, c : integer;
    function Max (a, b: integer): integer;
    begin
       ...
    end;
begin
   writeln('Введите два числа');
   read(a, b);
   c := Max ( a, b );
   writeln('Наибольшее число ', c );
end.
```

фактические параметры

вызов функции

Имена переменных, функций и процедур не должны совпадать!

Задания

«3»: Составить функцию, которая определяет наименьшее из трех чисел и привести пример ее использования.

Пример:

```
Введите два числа:
28 15
Наименьшее число – 15.
```

«4»: Составить функцию, которая определяет наибольшее из трех чисел и привести пример ее использования.

Пример:

```
Введите три числа:
28 15 10
Наибольшее число – 28.
```

Задания

«5»: Составить функцию, которая определяет сумму всех чисел от 1 до N и привести пример ее использования.

Пример:

```
Введите число:
100
сумма = 5050
```

Логические функции

Задача: составить функцию, которая определяет, верно ли, что заданное число – четное.

Особенности:

- ответ – логическое значение (`True` или `False`)
- результат функции можно использовать как логическую величину в условиях (`if, while`)

Алгоритм: если число делится на 2, оно четное.

```
if N mod 2 = 0 then
      { число N четное}
else { число N составное }
```

Логические функции

```
program qq;
var N: integer;
function Chet(N: integer): boolean;
begin
  if N mod 2 = 0 then
       Chet := True
  else Chet := False;
end;
begin
  writeln('Введите целое число');
  read(N);
  if Chet(N) then
       writeln(N, ' - четное число')
  else writeln(N, ' - нечетное число');
end.
```

результат – логическое значение

или
`Chet:=(N mod 2)= 0;`

вызов функции

Задания

«3»: Составить функцию, которая определяет, верно ли, что число оканчивается на 0.

Пример:

```
Введите число:  Введите число:
170   237
Верно.      Неверно.
```

«4»: Составить функцию, которая определяет, верно ли, что в числе вторая цифра с конца больше 6.

Пример:

```
Введите число:  Введите число:
178   237
Верно.      Неверно.
```

Задания

«5»: Составить функцию, которая определяет, верно ли, что переданное ей число – простое (делится только на само себя и на единицу).

Пример:

```
Введите число: Введите число:
29   28
Простое число. Составное число.
```