

**Тема урока:** Графические возможности Паскаля. Цикл с параметром.

**Девиз урока:** Знание только тогда знание, когда оно добыто усилием собственной мысли, а не памятью.

Л.Н.

Толстой

**Цель наших занятий:** Создание проекта: «Я и мои друзья».

**Цель нашего урока:** Познакомиться с циклом «со счетчиком», его называют «цикл с параметром», также «цикл для каждого».

## Устная работа.

*Как подключить библиотеку графических процедур?*

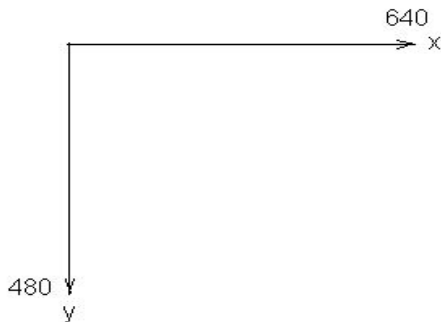
`uses graph; { подключение библиотеки графических процедур }`

*Как должна заканчиваться любая графическая программа?*

`CloseGraph;`

*Как выглядит координатная плоскость на экране?*

Начало координат находится в левом верхнем углу экрана. Ось  $x$  направлена слева направо, а ось  $y$  — сверху вниз. Размер экрана  $640 \times 480$  пикселей.



## Устная работа.

*Назовите основные графические примитивы.*

1. точка цвета  $c$  (может меняться от 0 до 15) с координатами  $(x, y)$  -

`putpixel(x, y, c);`

2. отрезок (линия) с началом в точке с координатами  $(x_1, y_1)$  и концом  $(x_2, y_2)$  -

`line(x1, y1, x2, y2);`

3. прямоугольник со сторонами параллельными осям координат и диагональю с началом в точке  $(x_1, y_1)$  и концом  $(x_2, y_2)$  -

`rectangle(x1, y1, x2, y2);`

4. окружность с центром в точке  $(x, y)$  и радиусом  $r$  -  
`circle(x, y, r).`

## Устная работа.

*С помощью, какой команды мы можем ввести в переменную **a** значение во время выполнения программы?*

С помощью команды read(a).

*С помощью, какой команды мы можем вывести значение **a** на экран компьютера?*

С помощью команды write(a);

Что получится в результате выполнения фрагмента программы:

x:=1;

x:=x+10;

x:=x-2;

write('x=',x);

Ответ: x=9

## Устная работа.

*Какие этапы разработки алгоритма вы знаете?*

Этап 1.

Разрабатывается примерный алгоритм достижения цели.

Этап 2.

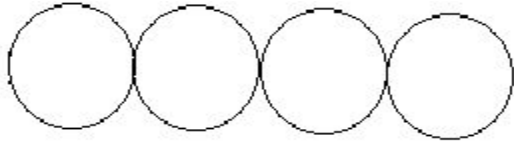
На втором этапе алгоритм детализируется с учётом среды его будущего выполнения.

Этап 3.

На третьем этапе алгоритм (программа) подвергается дальнейшему уточнению и полностью записывается на языке программирования с помощью команд.

## Новая тема.

Составить программу построения следующей картинki:



```
writeln('vvedite x,y,r');  
readln(x,y,r);  
l:=1;  
Circle(x,y,r);  
l:=2;  
X:=X+2*R;  
Circle(x,y,r);  
l:=3;  
X:=X+2*R;  
Circle(x,y,r);  
l:=4;  
X:=X+2*R;  
Circle(x,y,r);
```

**Вопрос:** Какой вид программы мы получили?

**Ответ:** Линейный.

**Вопрос:** Что такое линейная программа?

**Ответ:** Программа, в которой все действия выполняются последовательно одно за другим.

**Вопрос:** Что такое параметр l?

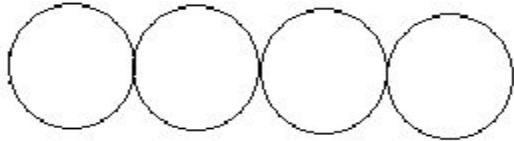
**Ответ:** Параметр, который указывает номер выводимой окружности.

**Вопрос:** Какие значения принимает эта величина?

**Ответ:** Это переменная величина, которая последовательно принимает значения 1, 2, 3, 4.

## Новая тема.

Составить программу построения следующей картинki:



```
writeln('vvedite x,y,r');  
readln(x,y,r);  
l:=1;  
Circle(x,y,r);  
l:=2;  
X:=X+2*R;  
Circle(x,y,r);  
l:=3;  
X:=X+2*R;  
Circle(x,y,r);  
l:=4;  
X:=X+2*R;  
Circle(x,y,r);
```

Посмотреть: [urokz1.pas](#)

**Вопрос:** Какой недостаток в данной программе?

**Ответ:** В программе встречаются повторяющиеся действия (команды).

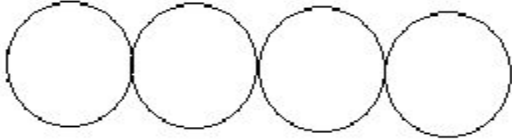
**Вопрос:** Как исправить этот недостаток?

**Ответ:** Изменить структуру программы.

**Вопрос:** Каким образом приведённое описание процедуры построения окружностей можно сделать более кратким?

## Новая тема.

Составить программу построения следующей картинki:



```
writeln('vvedite x,y,r');  
readln(x,y,r);  
l:=1;  
Circle(x,y,r);  
l:=2;  
X:=X+2*R;  
Circle(x,y,r);  
l:=3;  
X:=X+2*R;  
Circle(x,y,r);  
l:=4;  
X:=X+2*R;  
Circle(x,y,r);
```

**Ответ:**

Для  $l$ , изменяющегося от 2 до 4, повторять серию операторов:

**$X:=X+2*R$ ;**

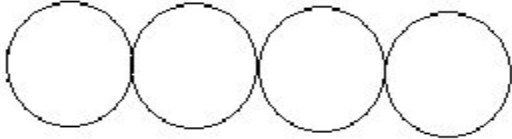
**Circle(x,y,r);**

Условимся причастный оборот «изменяющегося от 2 до 4» записывать как «:=2 до 4», а серию операторов заключать в фигурные скобки. Тогда возникает конструкция вида:



## Новая тема.

Составить программу построения следующей картинki:

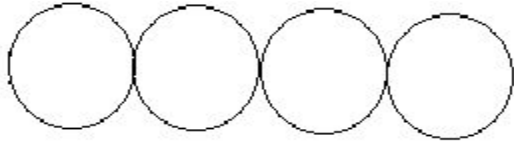


```
writeln('vvedite x,y,r');  
readln(x,y,r);  
I:=1;  
Circle(x,y,r);  
Для I:=2 до 4 повторять  
{X:=X+2*R;  
Circle(x,y,r);}
```

Мы получили циклическую конструкцию. Циклическая конструкция является приказом многократно повторять серию операторов. В данном случае число повторений заранее известно и определяется начальным и конечным значениями переменной I. Переменная I является указателем числа повторений. Её обычно называют счетчиком или параметром цикла.

## Новая тема.

Составить программу построения следующей картинki:



```
writeln('vvedite x,y,r');  
readln(x,y,r);  
l:=1;  
Circle(x,y,r);  
Для l:=2 до 4 повторять  
{X:=X+2*R;  
Circle(x,y,r);}
```

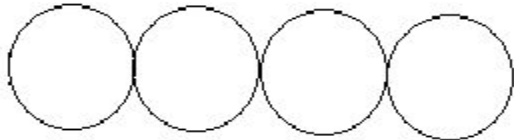
Что называется циклом с параметром?

Повторение определённого набора действий заданное число раз называется циклом «по счетчику», либо «цикл с параметром», либо «цикл для каждого».

- Серия операторов, выполняемая с каждым значением параметра, называется телом цикла.
- Шаг цикла – это однократное выполнение тела цикла.  
Используется, когда число шагов заранее известно

## Новая тема.

Составить программу построения следующей картинki:



```
writeln('vvedite x,y,r');  
readln(x,y,r);  
l:=1;  
Circle(x,y,r);  
For l:=2 to 4 do  
Begin  
X:=X+2*R;  
Circle(x,y,r);  
End;
```

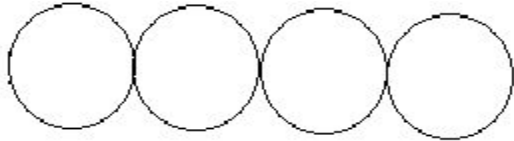
Описанной циклической конструкции в языке Паскаль соответствует оператор цикла с параметром. Чтобы перейти к его записи, слова «для», «до», «повторять» следует заменить ключевыми словами FOR («Для», «Фоо»), TO («ДО», «Ту»), DO (выполнять), а фигурные скобки – операторными скобками Begin и End.

Общий вид оператора цикла:

```
For <параметр> :=  
<начальное значение> to  
<конечное значение> do  
<операторы тела цикла>;
```

## Новая тема.

Составить программу построения следующей картинki:



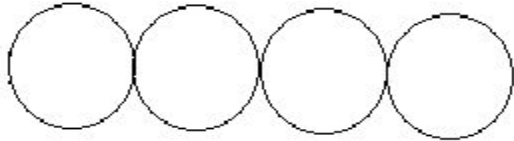
```
writeln('vvedite x,y,r');  
readln(x,y,r);  
l:=1;  
Circle(x,y,r);  
For l:=2 to 4 do  
Begin  
X:=X+2*R;  
Circle(x,y,r);  
End;
```

### ПОРЯДОК ВЫПОЛНЕНИЯ ОПЕРАТОРА FOR.

1. Вычисляются начальное и конечное значение.
2. Параметру цикла присваивается начальное значение, если оно не больше конечного.
3. Выполняются операторы цикла (тело цикла).
4. Значение параметра цикла увеличивается на один шаг.
5. Проверяется условие: если значение параметра цикла не больше конечного, то происходит переход к п.3, иначе - конец цикла и продолжение программы.

## Новая тема.

Составить программу построения следующей картинki:



```
writeln('vvedite x,y,r');  
readln(x,y,r);  
l:=1;  
Circle(x,y,r);  
For l:=2 to 4 do  
Begin  
X:=X+2*R;  
Circle(x,y,r);  
End;
```

Посмотреть: [urokz2.pas](#)

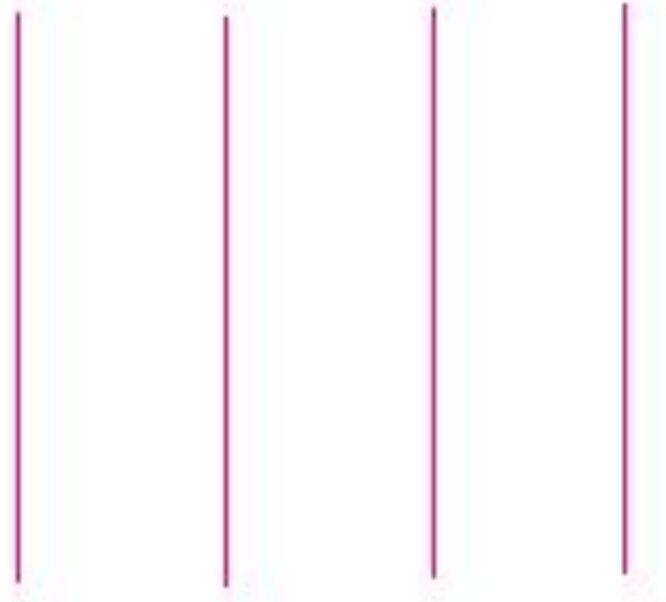
### ПРАВИЛА ИСПОЛЬЗОВАНИЯ ОПЕРАТОРА ЦИКЛА.

1. Запрещается вход в цикл, минуя оператор FOR, то есть передача извне управления на операторы тела цикла недопустима.
2. В операторе FOR можно задавать и отрицательный шаг, тогда общий вид оператора цикла будет таким:  
For <параметр> := <начальное значение> downto <конечное значение> do <операторы тела цикла>;
3. Переопределение значения параметра цикла в теле цикла разрешено, но этим следует пользоваться осторожно, т. к. это может привести к ошибкам.

## Закрепление материала.

**Задача 1.** Что получится в результате выполнения программы?

```
x:=100; y:=100;  
SetColor(5);  
For i:=1 to 4 do  
Begin  
line(x,y,x,y+150);  
x:=x+50;  
End;
```

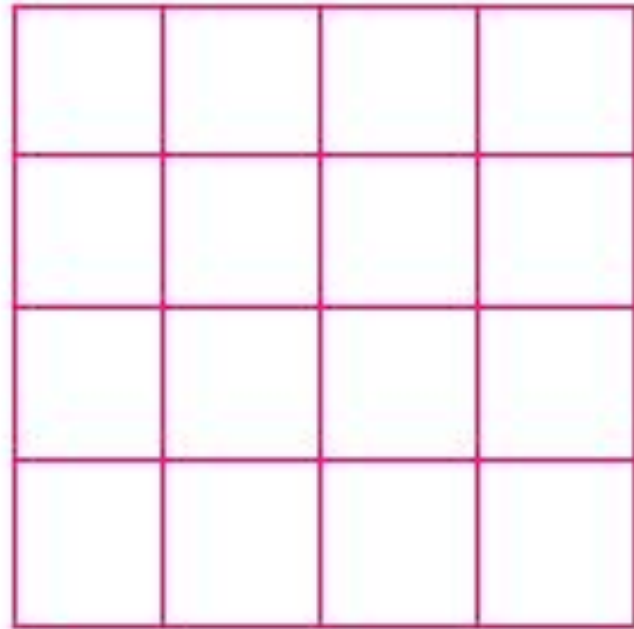


## Закрепление материала.

### Задача 2.

Допишите программу таким образом, чтобы компьютер выводил следующую картинку.

```
x:=100; y:=100;  
SetColor(5);  
For i:=1 to 4 do  
Begin  
line(x,y,x,y+150);  
x:=x+50;  
End;  
x:=100; y:=100;  
For i:=1 to 4 do  
Begin  
line(x,y,x+150,y);  
y:=y+50;  
End;
```



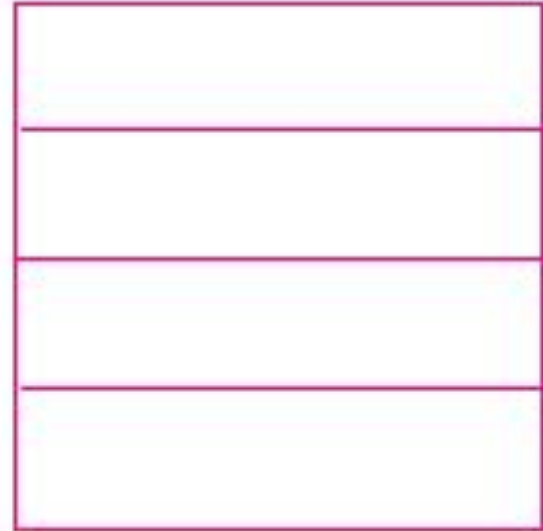
Посмотреть: [urok42.pas](#)

## Закрепление материала.

**Задача 3. Измените программу таким образом, чтобы компьютер нарисовал следующую картинку:**

```
x:=100; y:=100;  
SetColor(5);  
For i:=1 to 2 do  
Begin  
line(x,y,x,y+150);  
x:=x+150;  
End;  
x:=100; y:=100;  
For i:=1 to 4 do  
Begin  
line(x,y,x+150,y);  
y:=y+50;  
End;
```

**Посмотреть: [urok43.pas](#)**





# Тест

Посмотреть: [urok13.pas](#)

Посмотреть: [urok14.pas](#)

## Вариант 1

№ вопроса	<b>1</b>	<b>3</b>	<b>5</b>	<b>7</b>	<b>9</b>	<b>11</b>	<b>13</b>		
№ правильно го ответа	1	3	2	1-2- 3-4- 5	4	1	<table border="1"><tr><td></td><td></td></tr></table>		

## Вариант 2

№ вопроса	<b>2</b>	<b>4</b>	<b>6</b>	<b>8</b>	<b>10</b>	<b>12</b>	<b>14</b>		
№ правильно го ответа	3	2,4	1	2	3	1	<table border="1"><tr><td></td></tr><tr><td></td></tr></table>		

# Домашнее задание

Измените исходную задачу таким образом, чтобы за параметр цикла взять X, а не I. Вместо ? поставить нужный параметр.

Напишите алгоритм рисование 4-х окружностей:

```
Y:=100;
```

```
R:=50;
```

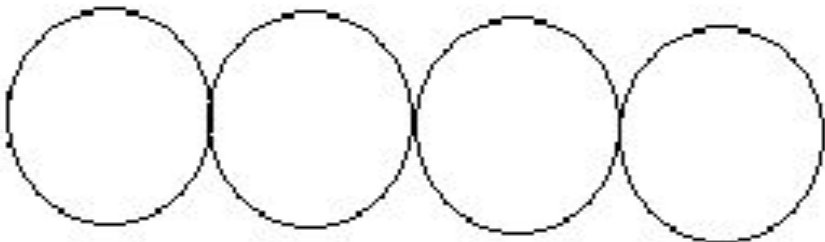
```
X:= 100;
```

```
Пока X<=? выполнять
```

```
{ Circle(x,y,r);
```

```
X:=X+?;}
```

Посмотреть: [urokdom.pas](#)



- Урок хочется закончить словами: **«Не возможно научить, возможно, научиться».**