

Использование графических возможностей языка программирования

**(точка, отрезок, окружность,
прямоугольник)**

При работе в графическом режиме изображение на экране строится не из символов, а из точек – **пикселов**. *Каждый пиксель (точка) имеет две координаты: x и y*

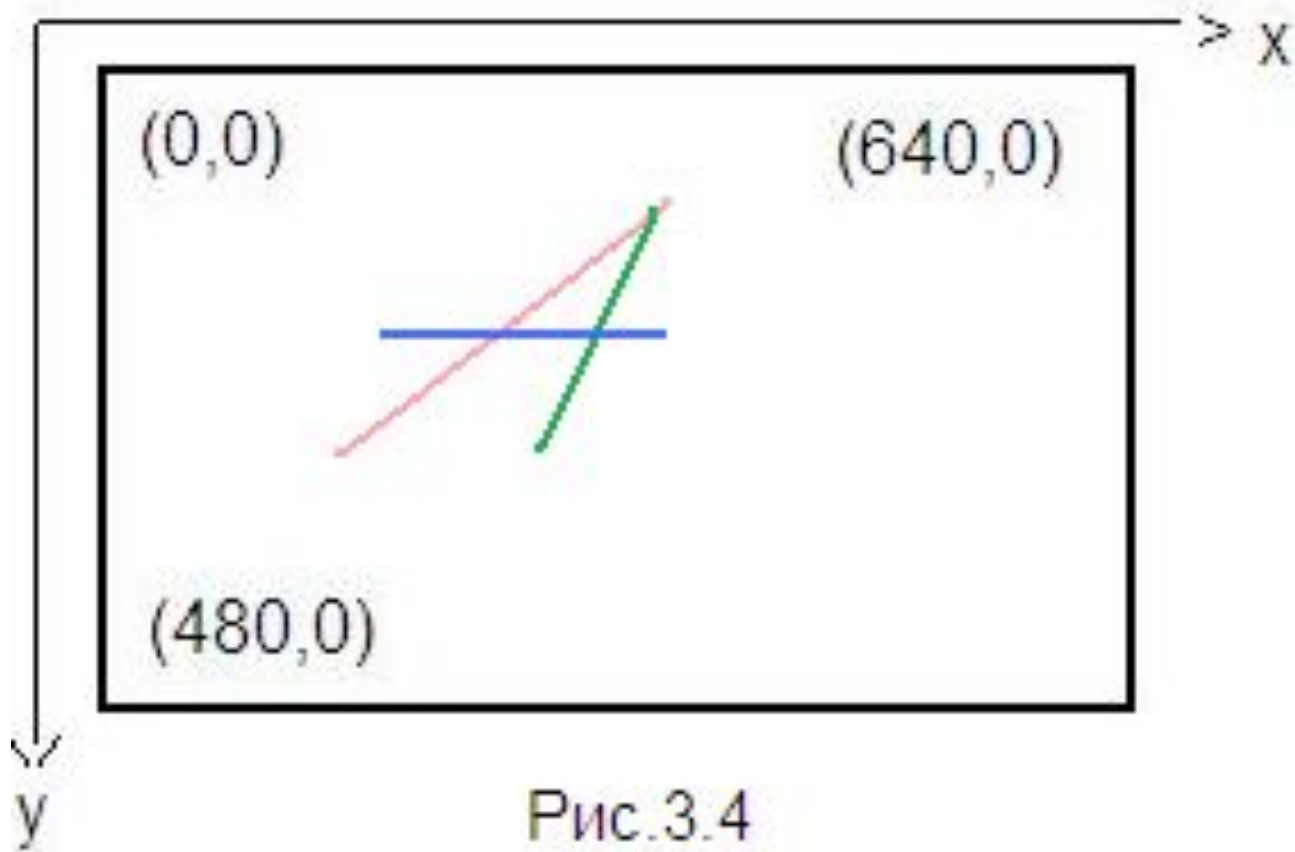


Рис.3.4

Геометрические размеры пикселя определяются *разрешением* монитора. Разрешение монитора задается в виде $x \times y$, где x – количество возможных пикселов на экране по горизонтали, а y – количество пикселов по вертикали. Например, известны следующие разрешения мониторов:

- 320x200;
- 640x480;
- 800x600;
- 1024x768;
- 1280x1024 – и т.д.

Любая графическая картинка формируется из простых геометрических фигур. Это точки, отрезки (линии), прямоугольники, окружности и т.д. Из геометрии известно, что положение геометрического объекта и его форма задаются координатами его точек.

Графические координаты задают положение точки на экране монитора. Поскольку минимальным элементом, к которому имеет доступ программист, является пиксель, в *качестве графических координат используют порядковые номера пикселов.*

Началом отсчета является левый верхний угол экрана. Значения x – координаты отсчитывается слева направо, а значения y – координаты – сверху вниз

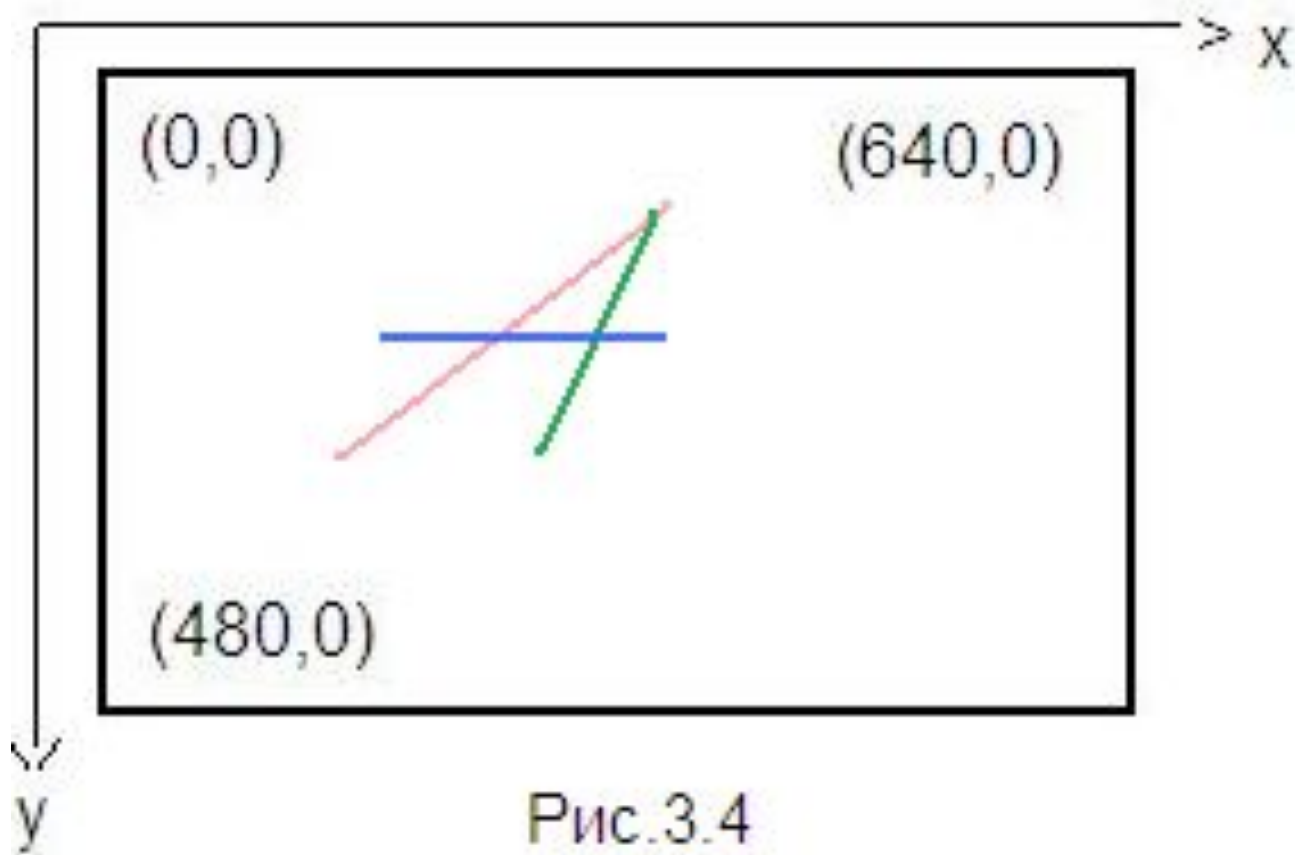


Рис.3.4

Важно помнить:

1. Графические координаты принимают только целочисленные значения.
2. Графические координаты принимают значения, ограниченные как снизу (нулевым значением), так и сверху (значением разрешения экрана монитора).
3. Графическая координата y отсчитывается сверху вниз.

Рисование различных геометрических фигур осуществляется с помощью специальных стандартных команд (процедур).

Команды для работы в графическом режиме хранятся в библиотечном модуле **GraphAbs.**

Подключение графического модуля:
uses GraphABC;

В модуле GraphAbs с помощью команды **setwindowsize(X,Y)** можно задавать *размеры графического окна.*

По умолчанию графическое окно будет принимать размеры экрана компьютера.

Рисование точки

setpixel(x:integer,y:integer,c:color) – рисует пиксел (точку) с координатами (x,y) цветом c.

Стандартные цвета c задаются символическими константами:

clBlack – черный

clPurple – фиолетовый

clWhite – белый

clMaroon – темно-красный

clRed – красный

clNavy – темно-синий

clGreen – зеленый

clBrown – коричневый

clBlue – синий

clSkyBlue – голубой

clYellow – желтый

clCream – кремовый

clAqua – бирюзовый

clOlive – оливковый

clFuchsia – сиреневый

clTeal – сине-зеленый

clGray – темно-серый

clLime – ярко-зеленый

clMoneyGreen – цвет
зеленых денег

clLtGray – светло-серый

clDkGray – темно-серый

clMedGray – серый

clSilver – серебряный

Пример 1. Демонстрация подключения модуля GraphAbs

```
program точка;
```

```
uses graphabc;           {подключение модуля GraphAbs}
```

```
begin
```

```
setwindowsize(640,480); {устанавливает размеры  
графического окна}
```

```
setpixel(100,120,clBlack); {устанавливает черный цвет пера  
и рисует точку с координатами (100,120)}
```

```
end.
```

Рисование линий

Общий вид команды:

Line(x1,y1,x2,y2), где **(x1,y1)** и **(x2,y2)** -
координаты точек отрезка, которого
соединяет линия.

Пример 2. Демонстрация рисования линии

Программа рисования отрезка, соединяющего две точки с координатами (120,150) и (150,80) красным цветом пера, МОЖЕТ ВЫГЛЯДЕТЬ ТАК:

```
program Linii;
```

```
  uses graphabc;
```

```
begin
```

```
  setwindowsize(640,480);
```

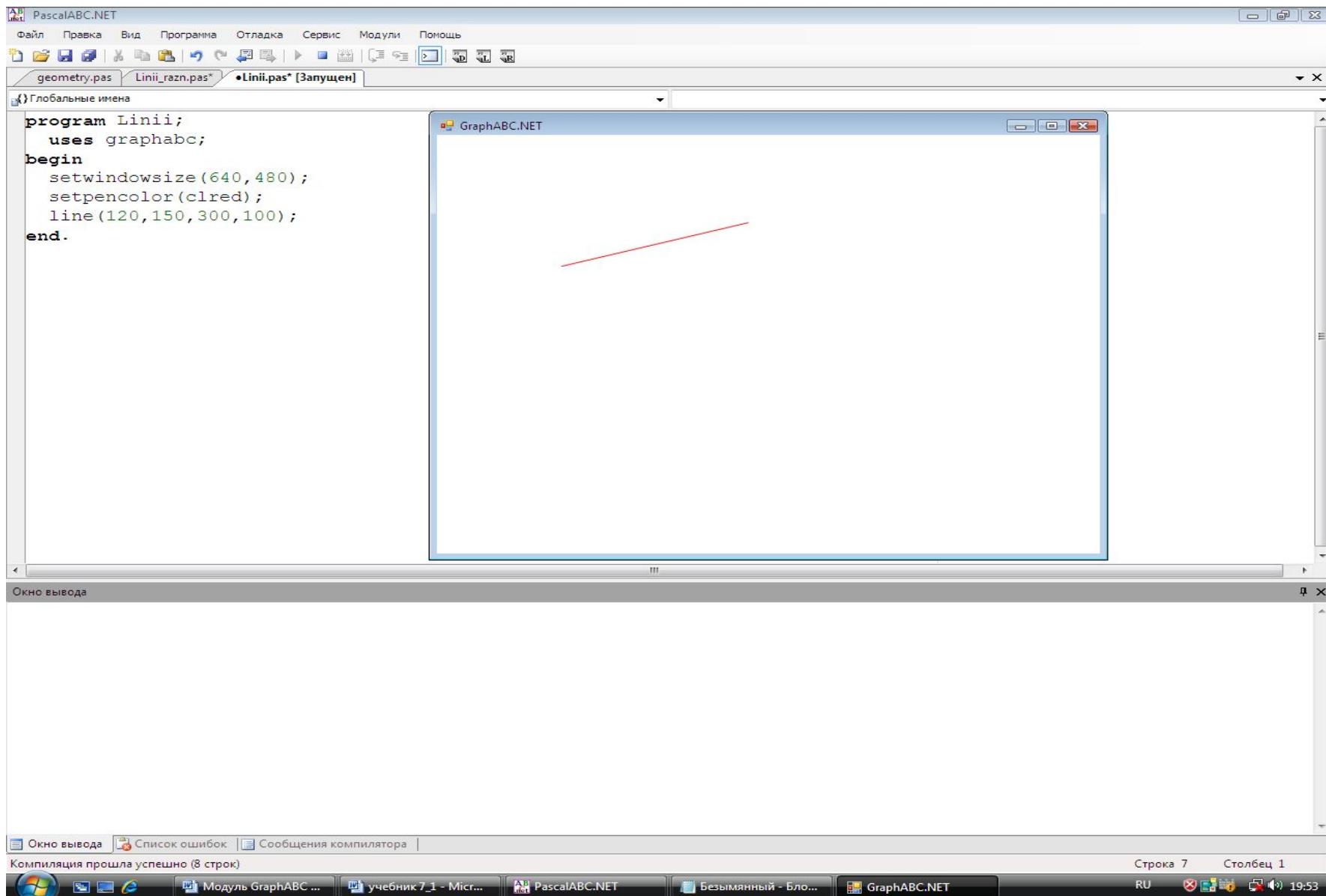
```
  setpencolor(clred);      {устанавливает красный цвет пера}
```

```
  line(120,150,300,100); {рисует отрезок от точки с  
                           координатами(120,150)до точки с  
                           координатами с координатами
```

```
(300,100)}
```

```
  end.
```

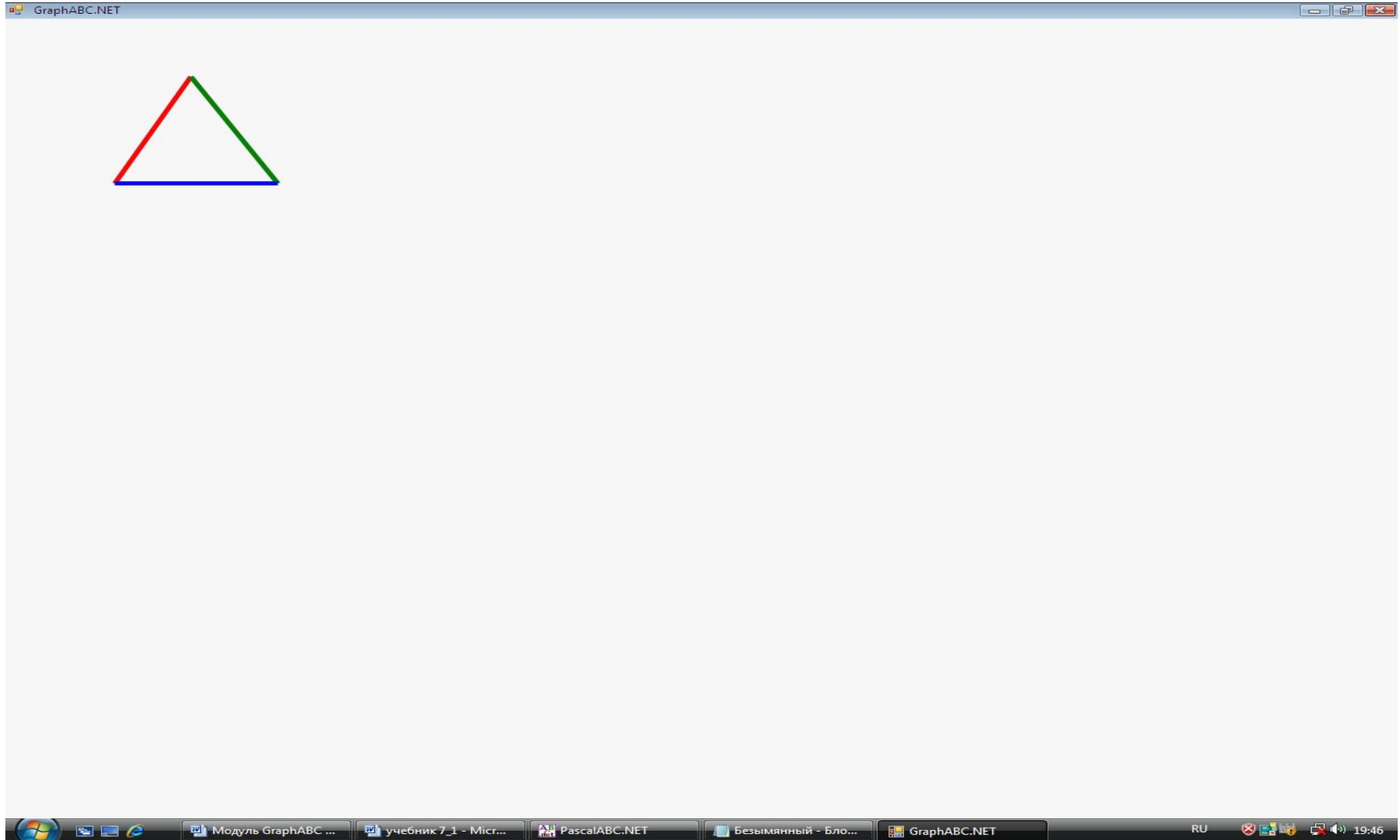
И вот что мы увидим на экране монитора после выполнения данной программы:



Пример 3. Демонстрация рисования линий разным цветом пера

```
program treugolnik;  
uses graphabc;  
begin  
  setwindowsize(640,480);  
  setpenwidth(5);      {устанавливает ширину текущего пера. В скобках  
                        указывает количество пикселей, образующих ширину линии}  
  setpencolor(clred);  {устанавливает красный цвет пера}  
  line(100,200,170,70); {рисует отрезок от точки с координатами (100,200) до  
                        точки с координатами (170,70)}  
  setpencolor(clGreen); {устанавливает зеленый цвет пера}  
  line(170,70,250,200); {рисует отрезок от точки(170,70) до точки(250,200)}  
  setpencolor(clBlue); {устанавливает синий цвет пера}  
  line(250,200,100,200); {рисует отрезок от точки(250,200) до точки(100,200)}  
  {результат - треугольник со сторонами разных цветов}  
end.
```

После выполнения программы на экране монитора появится графическое окно со следующим рисунком:



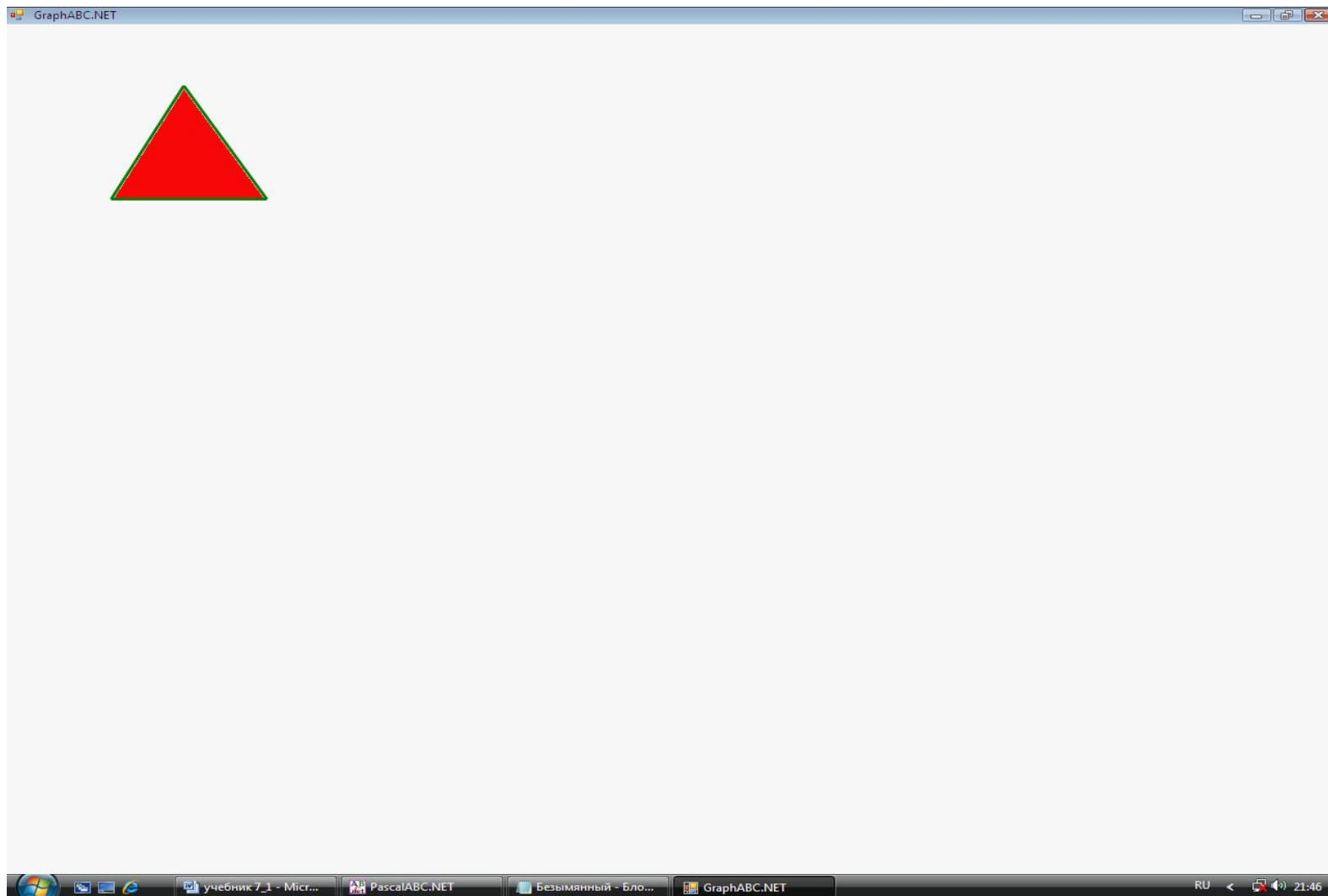
Важно помнить:

1. Рисуя линии, можно задавать ее размер (координатами ее концов), цвет, ширину (толщину) и стиль.
2. Для задания стиля линии в модуле GraphAbs существует процедура SetPenStyle (Style), где Style - константы стилей пера (см. Приложение к главе 3).
3. Линия может быть сплошной, пунктирной, штрихпунктирной, штриховой.

Пример 4. Демонстрация рисования замкнутых фигур с помощью линий и их закрашивание

```
Program treug_zakrash;  
  uses graphabc;  
begin  
  setwindow(640,480);  
  clearwindow(clWhite);    {очищает графическое окно белым  
    ЦВЕТОМ}  
  setpenwidth(3);         {устанавливает ширину текущего пера}  
  setpenstyle(pssolid);   {устанавливает стиль линии –  
    сплошная линия}  
  setpencolor(clgreen);   {устанавливает зеленый цвет пера}  
  line(100,200,170,70);   {рисует линии зеленым цветом}  
  line(170,70,250,200);  
  line(250,200,100,200);  
  floodfill(440,120,clred); {Закрашивает треугольник красным  
    цветом}  
end.
```

В результате выполнения программы на экране монитора в графическом окне появится треугольник, нарисованный зеленым и закрашенный красным цветом:



Важно помнить:

1. Закрашивать можно только замкнутые фигуры, контур которых нарисован одним цветом.
2. В процедуре заливки **floodfill(x,y,c)** указывается координата точки (x,y), которая обязательно должна попасть во внутреннюю область закрашиваемой фигуры.

Рисование прямоугольников и окружностей

Прямоугольники можно рисовать с помощью команды **rectangle(x1,y1,x2,y2)**

Окружности можно рисовать с помощью команды **circle(x,y,r)**

Пример 5. Демонстрация рисования прямоугольника и окружности

```
program gemetry;
```

```
uses graphabc;
```

```
begin
```

```
  setwindowsize(640,480);
```

```
  setpencolor(clBlue);
```

*{устанавливает голубой цвет пера для
рисования контура прямоугольника}*

```
  setpenwidth(6);
```

{устанавливает ширину пера}

```
  rectangle(50,50,250,150);
```

*{рисует прямоугольник, заданный
координатами противоположных вершин}*

```
  setpencolor(clred);
```

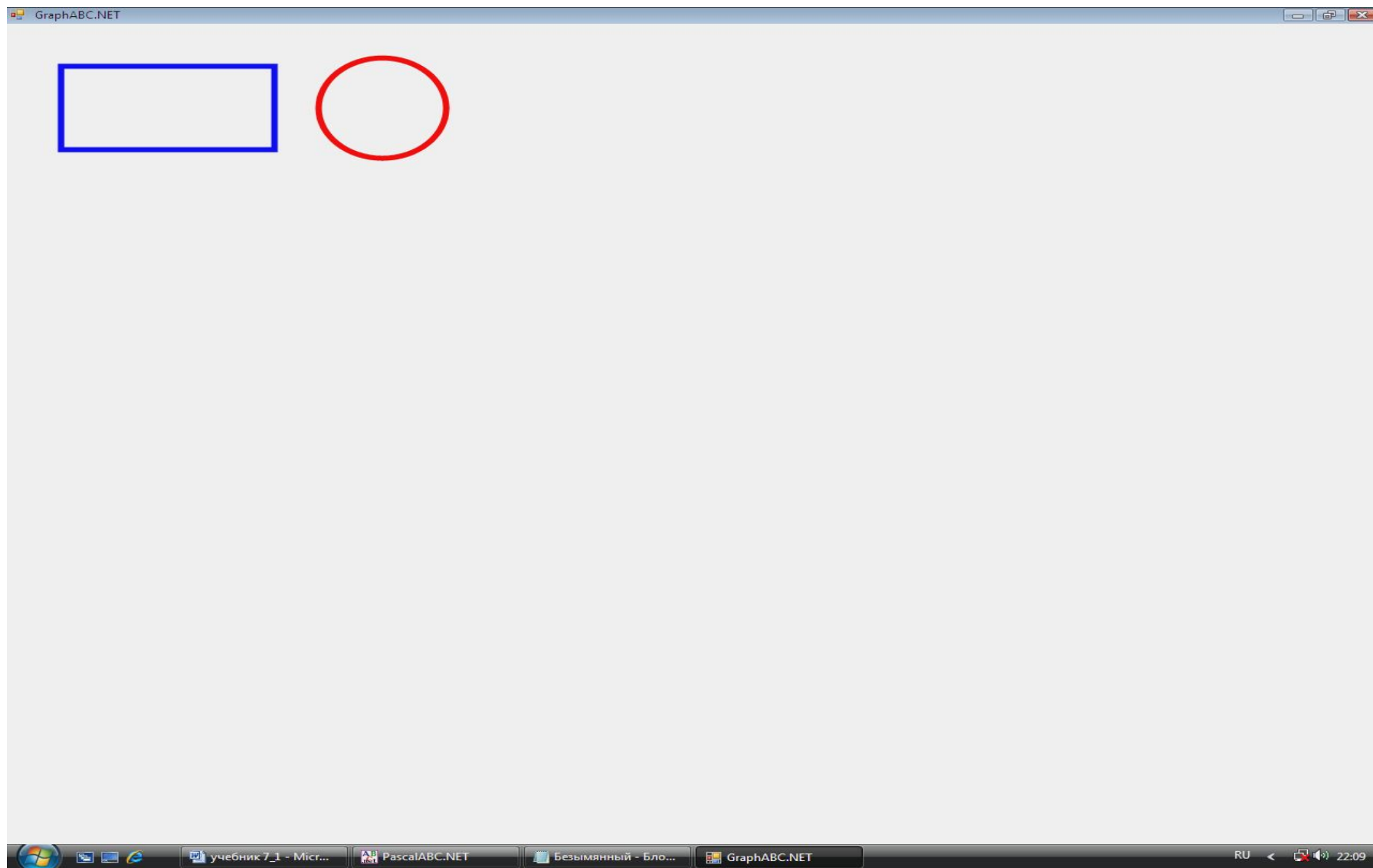
*{устанавливает красный цвет пера
для рисования контура окружности}*

```
  circle(350,100,60);
```

*{рисует окружность с центром в точке
с координатами (350,100) и радиусом 60}*

```
end.
```

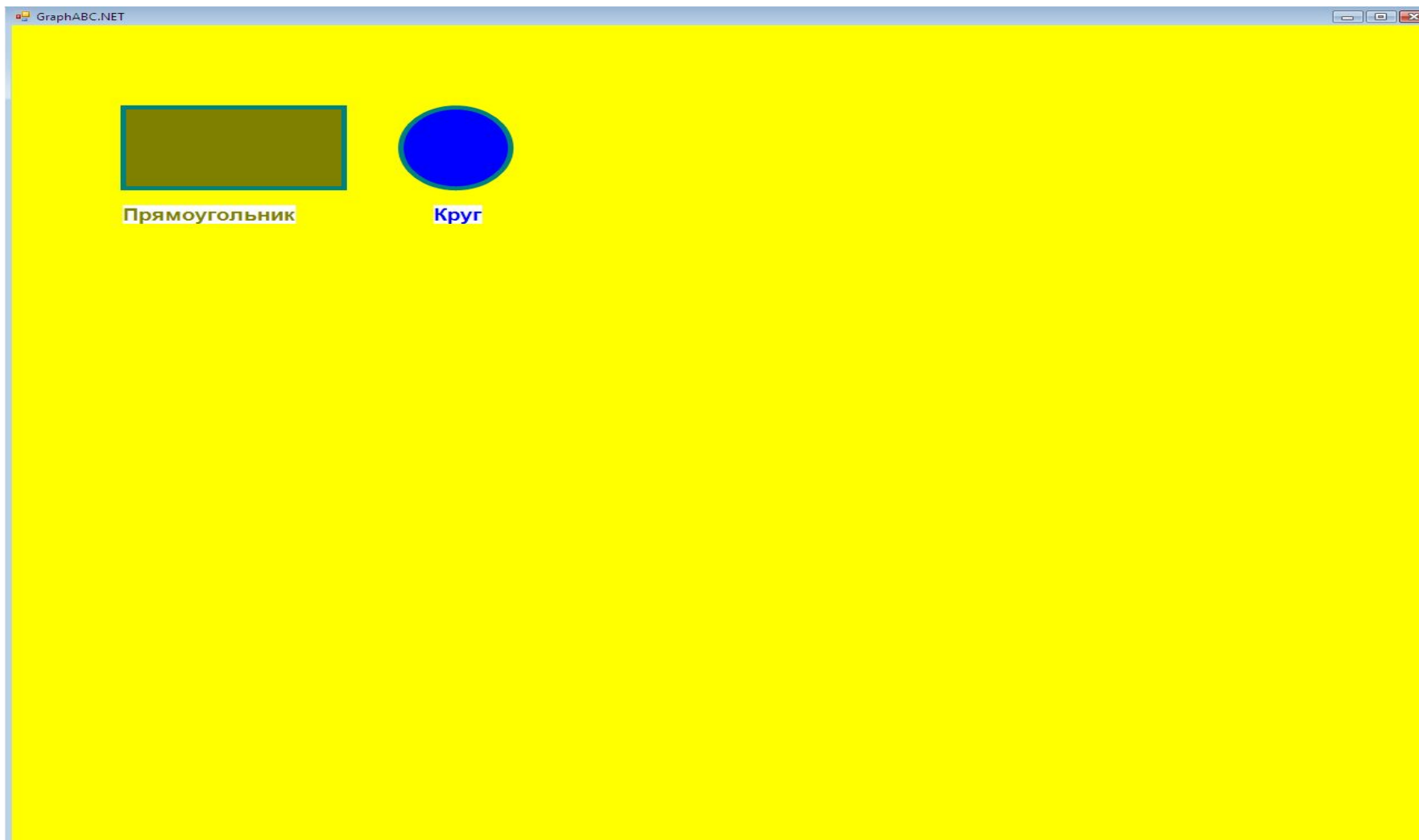
Результат выполнения программы



Пример 6. Демонстрация закрашивания прямоугольника и окружности и их надписи

```
program gemetry3;  
uses graphabc;  
Begin  
  setwindowsize(640,480);  
  clearwindow(clYellow);           {устанавливает желтый цвет фона}  
  setpencolor(clteal);              {устанавливает сине-зеленый цвет пера}  
  setpenwidth(5);                  {устанавливает ширину линии}  
  setbrushcolor(clolive);          {устанавливает оливковый цвет кисти}  
  rectangle(100,100,300,200);      {рисует закрашенный оливковым цветом прямоугольник}  
  setbrushcolor(clblue);          {устанавливает синий цвет кисти}  
  circle(400,150,50);              {рисует закрашенный синим цветом круг}  
  setfontstyle(fsbold);           {устанавливает стиль шрифта}  
  setfontsize(15);                {устанавливает размер шрифта}  
  setbrushcolor(clwhite);         {устанавливает белый цвет кисти}  
  setfontcolor(clolive);          {устанавливает оливковый цвет шрифта}  
  textout(100,220,'Прямоугольник'); {делает надпись}  
  setfontcolor(clblue);            {устанавливает синий цвет шрифта}  
  textout(380,220,'Круг');        {делает надпись}  
end.
```

После выполнения программы графическое окно на экране монитора будет выглядеть так:



Из приведенного выше примера видно, что рисовать можно на «холсте» определенного цвета, который задается с помощью *процедуры установки цвета графического окна* **clearwindow(color)**.

Закрасить прямоугольник и круг можно, используя процедуру закраски кистью **setbrushcolor (color)**

С помощью процедур **setfontcolor**, **setfontstyle**, **setfontsize**, **setbrushcolor**, **textout** рисунки можно подписать.

Создать такую картинку:



```
program skvoreshnik;  
uses graphabc;  
begin  
  setwindow(1280,1024);  
  clearwindow(clwhite);    {устанавливает белый цвет фона}  
  setpencolor(clteal);     {устанавливает сине-зеленый цвет пера}  
  setpenwidth(5);         {устанавливает ширину линии}  
  rectangle(100,150,250,300); {рисует прямоугольник}  
  setbrushcolor(clyellow); {устанавливает желтый цвет кисти для  
    закраски круга}  
  circle(170,200,25);     {рисует закрашенный желтым цветом круг}  
  floodfill(150,160,clBrown); {закрашивает коричневым цветом  
    прямоугольник - стену скворечника}  
  line (100,150,175,90);   {эти линии рисуют крышу}  
  line (175,90,250,150);  
  floodfill(175,100,clgreen); {Закрашивает треугольник (крышу)  
    коричневым цветом}  
end.
```

В модуле **graphabc** имеется еще и другие команды рисования графических примитивов, например, команда рисования эллипса. Зная основы работы в графическом режиме, их вы сможете изучить самостоятельно (см. Приложение к главе 3).