Курс графики на языке Компонентный Паскаль. Задание 1.

Мы, с вами в этом курсе, будем изучать программирование на примерах графических задач. Начнем с простейших геометрических объектов: точки, линии, окружности, прямоугольника, с помощью которых, можно рисовать сложные объекты. Но сейчас главная задача – изучить команды языка программирования для того, чтобы позже перейти к более сложному курсу, состоящему из задач с большим математическим содержанием.

Для того, чтобы решать даже простые графические задачи необходимо изучить общее устройство программы и основные графические команды. Начнем с того, как устроена программа.

Что такое модуль и процедура

Процедура – это последовательность команд, выполняющих необходимые действия. Процедура описывается так:

```
PROCEDURE Имя процедуры;
BEGIN
    Набор команд
END Имя процедуры;
```

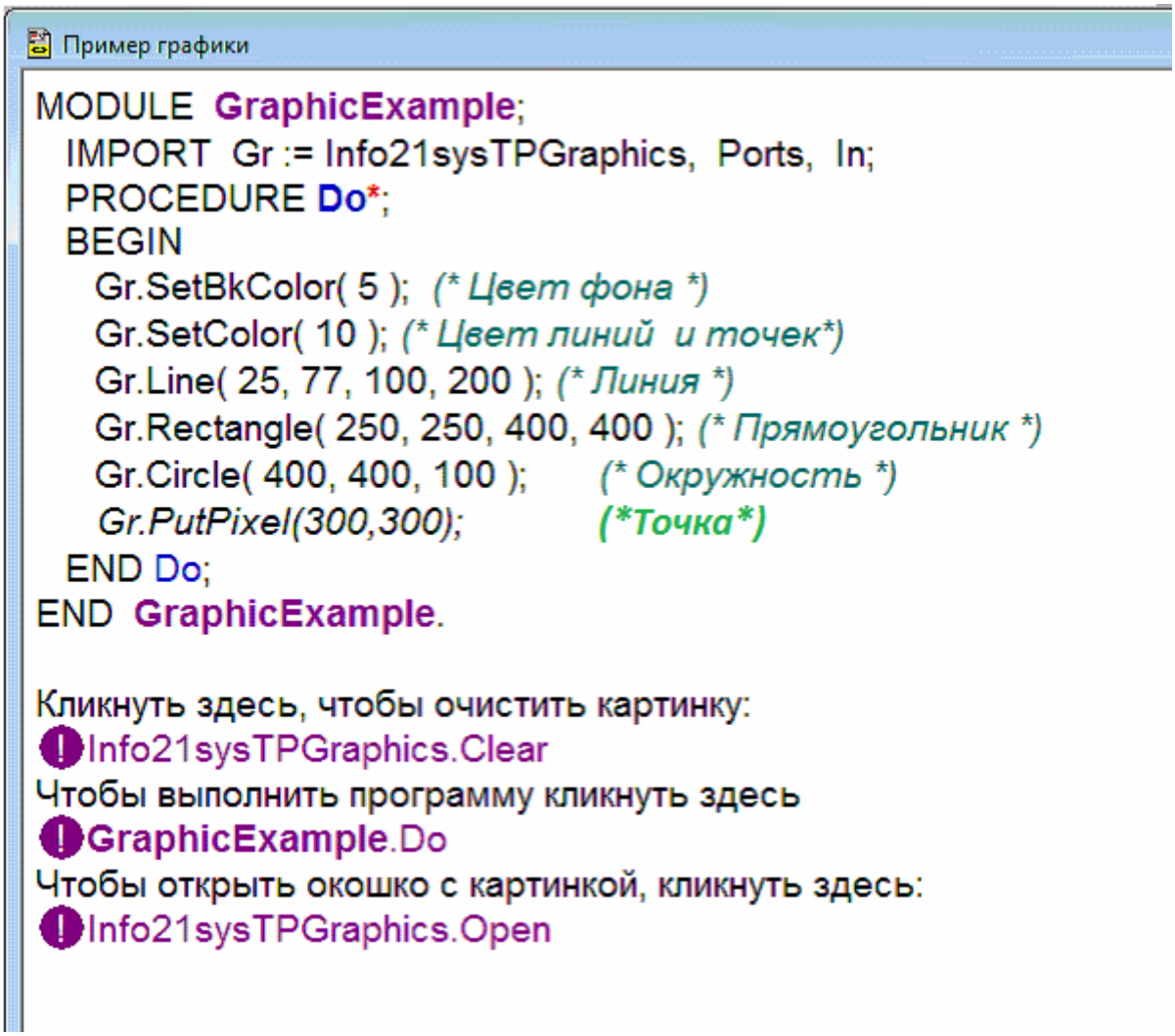
Здесь слова, записанные заглавными буквами, называются ключевыми, они обязательны, и их необходимо писать по правилам языка именно так. Имя процедуры это любое имя, которое вы хотите дать своей программе. Например:

```
PROCEDURE MyPro;
BEGIN
    Набор команд
END MyPro;
```

Ваши имена не должны совпадать с ключевыми словами. Это запрещено. Процедуры размещаются внутри модулей. Модуль это своего рода ящик, в котором хранятся процедуры. Пример:

```
MODULE Имя Модуля
PROCEDURE MyPro;
BEGIN
    Набор команд
END MyPro;
END Имя Модуля.
```

Имя модуля, как и имя процедуры, может быть любым, не совпадающим с ключевыми словами. Внутри модуля может быть размещено любое количество процедур, но в этом курсе, мы везде будем создавать одну процедуру в одном модуле. Теперь рассмотрим несколько графических команд: установки цвета для фона, линий и точек. Команды рисования точек, линий, прямоугольников и окружностей. Ниже на рисунке пример такой программы:



Из вводного урока вы уже знаете, что набранный текст необходимо компилировать, то есть переводить его в машинный код. В BlackBox это делается нажатием на кнопку *F8* . Если в программе есть ошибки, то в месте ошибки появится неблшой серый квадрат:

```
MODULE Info21TPGraphicsПример;
  IMPORT Gr := Info21sysTPGraphics, Ports, In;

  PROCEDURE Do*;
    VAR i, j: INTEGER;
  BEGIN
    Gr.Line( 25, 77, 100 200 );
  END Do;
```

В это фрагмента программы после координаты 100 нет запятой. Это ошибка. Если в вашем тексте не будет ошибок, то это означает, что система создала исполняемый код и можно щелкать мышкой на коммандер и исполнять программу.

Сейчас ваша задача набрать текст от строки MODULE GraphicExample до строки END GraphicExample. Здесь есть такая строка:

```
IMPORT Gr:=Info21sysTPGraphics, Port, In;
```

Это важная строка, которую необходимо дополнительно пояснить. Язык программирования содержит очень большое количество команд. Их настолько много, что приходится часть команд распределять по специальным файлам, которые называются библиотеками. В этой программе используются три библиотеки: Info21sysTPGraphics, Port, In. Если команда взята из библиотеки, то необходимо указать имя библиотеки через точку. Например, команда рисование окружности должна выглядеть так:

```
Info21sysTPGraphics.Circle(400,400,100);
```

Но эта запись очень велика по длине, поэтому используется более имя. В нашем примере это имя Gr. Сейчас рассмотрим, как работают графические команды. Для этого прежде всего, необходимо знать, что экран монитора представляет собой координатную плоскость с началом координат в верхней левой точке. То есть в верхнем левом углу монитора находится точка с координатами ***x=0*** и ***y=0.*** Ось ***Y*** идет сверху вниз, а ось ***X*** слева направо. Размеры вашего экрана в точках (пикселях) могут несколько отличаться, но если вы примете размеры по оси X от 0 до 600 и размеры по оси Y от 0 до 450, то ошибка в любом случае будет небольшой. В крайнем случае, эти размеры можно определить экспериментально, рисуя линии разной длины. А теперь рассмотрим команды нашего примера и опишем их.

Gr.SetBkColor(Цвет) – команда, задающая цвет фона экрана монитора. Цвет это число от 0 до 15. Какое число, какому цвету соответствует, определите экспериментально.

Gr.SetColor(Цвет) – команда, задающая цвет линий и точек. Цвет это число от 0 до 15, также как и цвет фона.

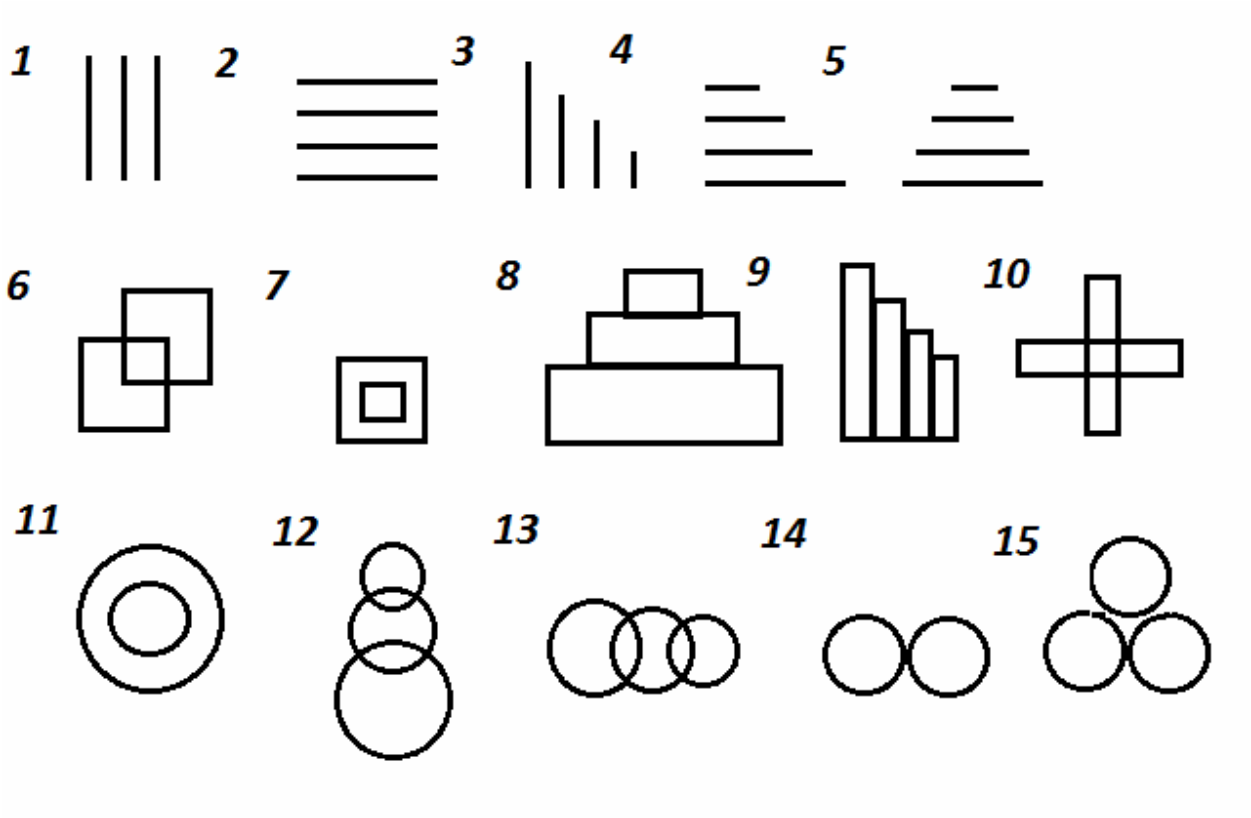
Gr.Line(x1,y1,x2,y2) – команда, рисующая линию от точки координатами *x1, y1* до точки с координатами *x2, y2*

Gr.Rectangle(x1,y1,x2,y2) – команда, рисующая прямоугольник с диагональю от точки с координатами *x1, y1,* до точки с координатами *x2,y2*

Gr.Circle(x,y,R) – команда, рисующая окружность от точки *x, y,* радиусом *R*

Gr.PutPixel(x,y) – команда, рисующая точку с координатами *x,y*

Сейчас ваша задача понять, как работают описанные выше команды. Для этого, изложенной теории может оказаться недостаточно. Вы должны поэкспериментировать с программой, изменяя данные внутри команд, наблюдая и анализируя самостоятельно результат. Для запуска этого примера графической процедуры необходимо выполнение трех команд. Первая – очистка графического окна. Вторая – выполнение процедуры. Третья команда открывает графическое окно и показывает результат. Если все понятно и все получилось, то необходимо выполнить десять заданий, представленных ниже. Это десять простых рисунков.


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15