

Программирование

Занятие 1

Pascal ABC. Основные математические функции.

Функция	Описание
<code>sqr(n)</code>	Возвращает квадрат аргумента. Необходимо помнить о диапазоне возвращаемого значения.
<code>abs(n)</code>	Возвращает модуль (абсолютное значение) числа.
<code>pred(n)</code>	Возвращает значение, на единицу меньше аргумента.
<code>succ(n)</code>	Возвращает значение, на единицу больше аргумента.
<code>odd(n)</code>	Возвращает TRUE, если аргумент нечетный, иначе – FALSE
<code>even(n)</code>	Возвращает TRUE, если аргумент четный, иначе – FALSE
<code>chr(n)</code>	Возвращает символ (тип <code>char</code>) с кодом <code>n</code> .
<code>random(n)</code>	Возвращает случайное число в диапазоне от 0 до <code>n-1</code> .

Pascal ABC. Основные математические функции.

$\text{sqrt}(x)$ - корень квадратный из X

$\text{int}(x)$ - целая часть X (результат - вещественное число)

$\text{trunc}(x)$ - целая часть X (результат - целое число)

$\text{round}(x)$ - округляет число X до целого (результат - целое число)

$\text{frac}(x)$ - дробная часть от X

pi - выдаёт число 3.1415926..... (19 знаков после запятой)

$\text{sin}(x)$ - синус X

$\text{cos}(x)$ - косинус X

$\text{arctan}(x)$ - арктангенс X

$\text{exp}(x)$ - возведение числа $e=2.7183$ в степень X (e^x)

$\text{ln}(x)$ - логорифм числа X

Random - выбирается случайное число от 0 до 1 (вещественное)

Pascal ABC. Основные математические функции.

Возведение в степень в Паскаль

1 способ: {a - число}, {b - степень}

r:=power(a,b)

2 способ: $X^Y = \exp(\ln(X) * Y)$

var
3 способ:
num, deg: integer;
res: real;
i: byte;
begin
write (`Основание степени: `); readln (num);
write (`Показатель степени: `); readln (deg);
res := 1;
i := 0;

```
while i `меньше` abs(deg) do begin  
  res := res `умножить` num;  
  i := i + 1  
end;  
if deg `меньше` 0 then  
  res := 1 / res;  
writeln (res:10:5);  
readln (res)  
end.
```

Целочисленные вычисления на языке Pascal

1. Записать следующие выражения на языке Pascal, считая все переменные и действия целочисленными.

$(a+b)(a-b)$	
$(1+x)^2$	
$\frac{15x}{y}$	
$2x \cdot 2y$	

Целочисленные вычисления на языке Pascal

2. Вычислить (в уме) значение выражения

$$(a+1)*(-a), \text{ если } a=10$$

$$(x+1) \operatorname{div} (x-1), \text{ если } x=2 \text{ и если } x=4$$

$$2*4 \bmod 3$$

$$2*(4 \bmod 3)$$

$$-a*(-1), \text{ если } a=431$$

Целочисленные вычисления на языке Pascal

3. Напишите и выполните программу на компьютере

- a. Программа вводит одно число, и вычисляет его квадрат, вычитая из него удвоенное значение введенного числа.
- b. Программа вводит два числа, выводит сумму квадратов этих чисел минус их произведение.
- c. Программа вводит два числа, выводит сумму их модулей (см. функцию *abs*)

Целочисленные вычисления на языке Pascal

4. Напишите программу, которая решает следующую задачу

Оплата Интернета в школе состоит из двух частей: 100 долларов в месяц за доступ и 5 долларов в месяц за поддержку школьного сайта. Платеж можно делать независимо за любое количество месяцев за доступ и за поддержку сайта. Напишите программу, которая позволяла бы вводить отдельно количество оплачиваемых месяцев для доступа и поддержки и выводила бы сумму оплаты.

Использование модуля CRT

- Язык Паскаль имеет специальную библиотеку работы с экраном в текстовом режиме. Она называется CRT. Эта библиотека содержит набор подпрограмм, позволяющих задавать цвет выводимых букв, цвет фона, устанавливать курсор в желаемую позицию.
- В текстовом режиме считается, что окно содержит 25 строк по 80 колонок. В каждой позиции может быть один символ (буква, цифра, специальные знаки).
- Графические объекты (прямые, окружности и т.п.) в текстовом окне не допустимы.

Подключение библиотеки к программе осуществляется предложением `Uses`:

```
Program Card;
```

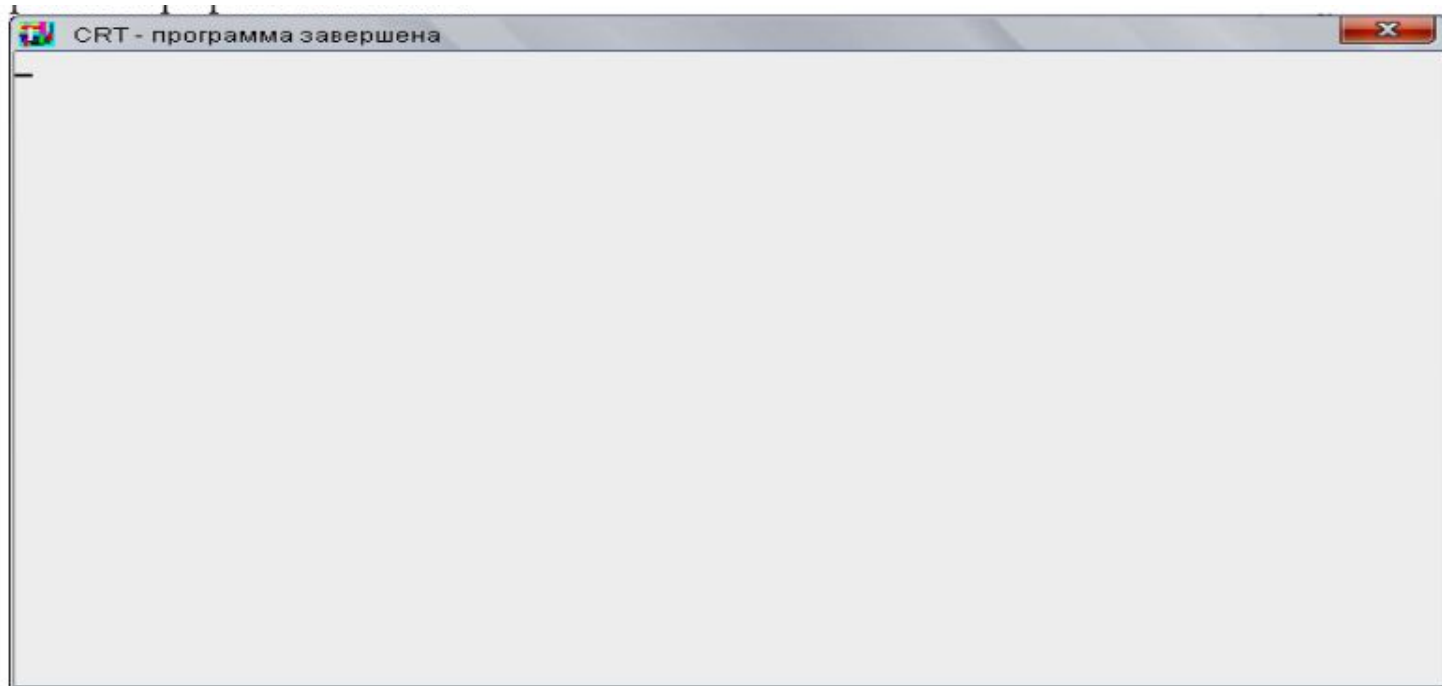
```
Uses CRT; { Это предложение вставляется сразу после Program }
```

Pascal ABC. Графический модуль

Для осуществления диалога с пользователем необходимо подключить в программе графический модуль, который подключается инструкцией `Uses CRT;`

Подключение этого модуля даёт возможность при выполнении вашей программы отображать графическое окно

После подключения графического модуля следует раздел описания переменных



Использование модуля CRT

Рассмотрим использование подпрограмм этой библиотеки на примере создания заставки программы.

```
Program Card;  
Uses CRT;  { Подключить модуль CRT }  
begin  
  ClrScr;           { Очистить экран }  
  TextColor(White); { Установить белый цвет букв }  
  TextBackGround(Blue); { Установить синий цвет фона }  
  GotoXY(36,13);    { Поставить курсор в 36 колонку, 13 строку }  
  write(' Привет '); { Вывести текст }  
  ReadKey;         { Ожидать нажатия любой клавиши }  
end.
```

Наберите текст этой программы и выполните её.

Модуль CRT. Отдельные процедуры

ClrScr – очищает экран или текущее окно (от англ. Clear Screen), закрашивая его текущим цветом фона, установленным процедурой TextBackGround. По умолчанию – цвет экрана белый.

TextColor (*цвет*) – устанавливает текущий цвет букв, который будет использоваться операторами write и writeln. По умолчанию используется светло-серый цвет букв.

TextBackgroud (*цвет*) – устанавливает текущий цвет фона, который будет использоваться операторами write и writeln.

GotoXY (*колонка, строка*) – переводит курсор в указанную позицию; вывод, осуществляемый последующим оператором write или writeln будет начинаться с этой позиции. Обычно экран в текстовом режиме содержит **80 колонок** и **25 строк**.

ReadKey – ожидание нажатия на любую клавишу (далее мы рассмотрим другое применение этой подпрограммы).

Модуль CRT. Отдельные процедуры

В качестве цвета может использоваться число от 0 до 15, но лучше пользоваться определенными в модуле CRT константами:

Таблица цветов

Black	Черный	DarkGray	Темно-серый
Blue	Синий	LightBlue	Светло-синий
Green	Зеленый	LightGreen	Светло-зеленый
Cyan	Небесно голубой	LightCyan	Ярко-голубой
Red	Красный	LightRed	Светло-красный
Magenta	Малиновый	LightMagenta	Светло-малиновый
Brown	Коричневый	Yellow	Желтый
LightGray	Светло-серый	White	Белый

Модуль CRT. Отдельные процедуры

Имя константы	Номер цвета	Цвет
Black	0	Черный
Blue	1	Темно-синий
Green	2	Темно-зеленый
Cyan	3	Бирюзовый
Red	4	Красный
Magenta	5	Фиолетовый
Brown	6	Коричневый

Модуль CRT. Отдельные процедуры

DarkGray	8	Темно-серый
LightBlue	9	Синий
LightGreen	10	Светло - зеленый
LightCyan	11	Светло-бирюзовый
LightRed	12	Розовый
LightMagenta	13	Малиновый
Yellow	14	Желтый
White	15	Белый

Модуль CRT. Полезные подпрограммы

Window (*x1, y1, x2, y2*) – создание окна вывода. В случае задания окна, все операторы `write` и `writeln` выводят только в него, процедура `ClrScr` будет очищать только это окно.

WhereX, WhereY – две функции, позволяющие узнать координаты местонахождения курсора. *Пример использования:* `GotoXY (WhereX+10, WhereY+2)`.

Модуль CRT. Полезные подпрограммы

Модуль `Sounds` содержит процедуры и функции для работы со звуком:

`n := LoadSound (fname)` – загружает звук из файла с именем `fname` в оперативную память и возвращает описатель звука в целую переменную `n` (звуковой файл должен иметь любой формат);

`PlaySound (n)` – начинает проигрывание звука с описателем `n`;

`StopSound (n)` – останавливает проигрывание звука с описателем `n`;

`RewindSound (n)` – "перематывает" звук с описателем `n` на начало;

`DestroySound (n)` – удаляет звук с описателем `n` из оперативной памяти, описатель звука при этом становится недействительным;

`SoundLength (n)` – возвращает длительность звука с описателем `n` в миллисекундах.

Модуль CRT. Полезные подпрограммы

Пример использования модуля sound:

```
Program Player;  
Uses Sounds, CRT;  
var n: integer;  
begin  
  n:=LoadSound('C:\WINDOWS\MEDIA\ringin.wav'); // укажите расположение файла  
  PlaySound(n);  
  Sleep(SoundLength(n));  
  DestroySound(n);  
end.
```

Модуль CRT. Задание №1

Напишите программу «Визитная карточка», которая выводила бы на экран хорошо оформленные ваши личные данные: имя, фамилию, год рождения, адрес, телефон. Используйте разные цвета, хорошо скомпонуйте.

Используйте возможности
модуля

Sounds |

Графический модуль GraphABC

Для подключения после заголовка программы необходимо написать

Uses GraphABC.

Модуль GraphABC содержит константы, типы, процедуры и функции для рисования в графическом окне.

Они подразделяются на несколько групп:

Графические примитивы

Действия с цветом

Действия с точками и прямоугольниками

Действия с пером

Действия с кистью

Действия со шрифтом

Действия с рисунками

Действия с графическим окном

Графический модуль GraphABC

Пример программы: (выполните)

```
Program Gr01;  
Uses GraphABC;  
begin  
  SetWindowSize(256,256);  
  SetPenColor( RGB(0,0,255) );  
  Line(0,0, WindowWidth, WindowHeight)  
  Line(0,WindowHeight, WindowWidth, 0)|  
end.
```

Графический модуль GraphABC

```
Program Gr01;  
Uses GraphABC;  
begin  
  SetWindowSize(256,256);  
  SetPenColor( RGB(0,0,255) );  
  Line(0,0, WindowWidth, WindowHeight)  
  Line(0,WindowHeight, WindowWidth, 0)|  
end.
```

Процедура Line рисует линию из точки, заданной первой парой координат, в точку, заданную второй парой координат. Функции WindowWidth и WindowHeight возвращают текущие значения ширины и высоты графического окна (следовательно, если вы поменяете размер окна в процедуре SetWindowSize, то линии все равно будут проводиться из угла в угол)

Процедура SetWindowSize устанавливает размер графического окна 256×256 пикселей.

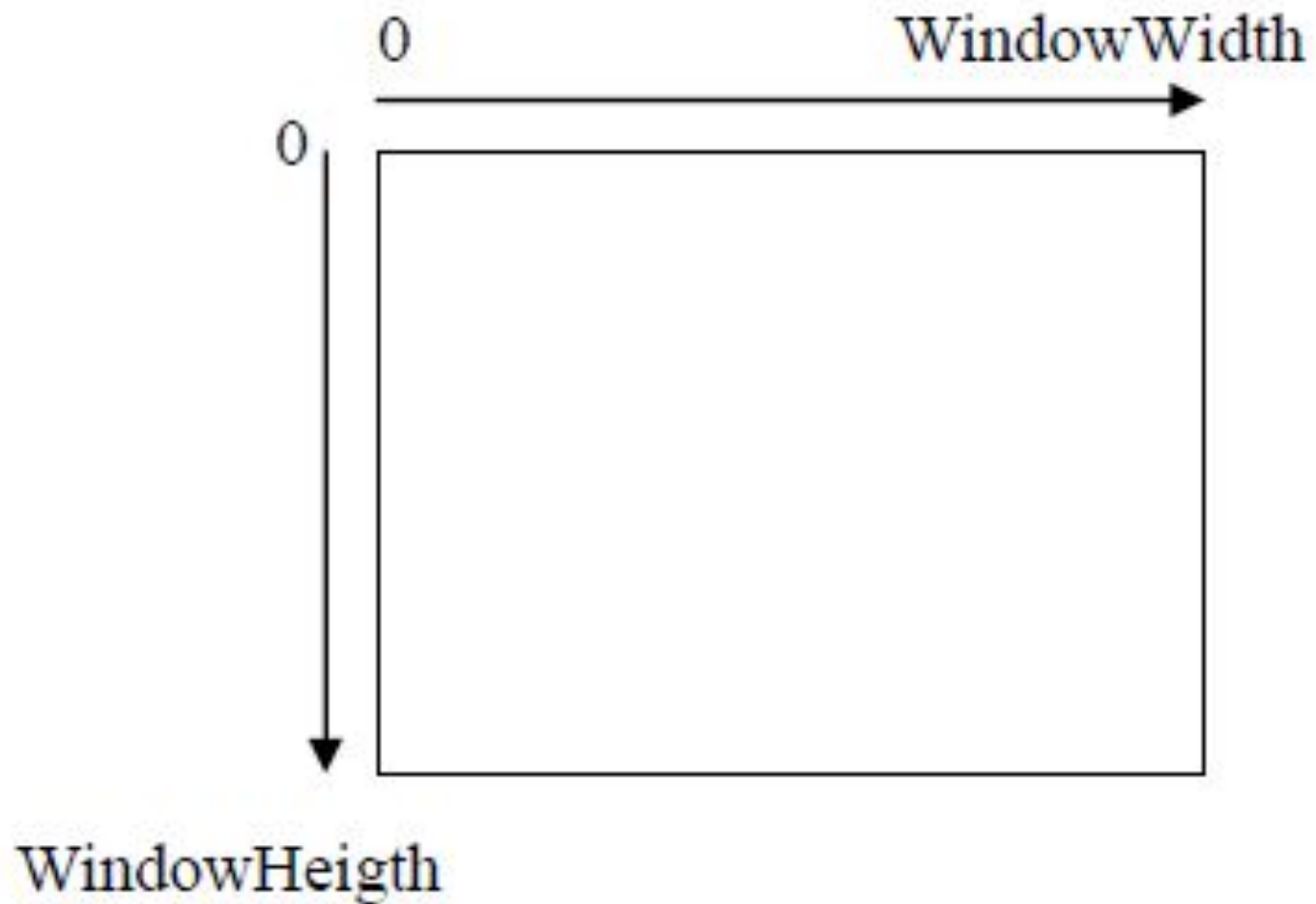
Процедура SetPenColor определяет цвет рисования линий.

Функция RGB возвращает цвет, заданный своими *Red*, *Green*, *Blue* компонентами:

```
function RGB(r,g,b: byte): Color;
```

Возвращает цвет, который содержит красную (r), зеленую (g) и синюю (b) составляющие (r,g и b - в диапазоне от 0 до 255)

Графический модуль GraphABC



Графический модуль GraphABC

Задание цвета

Как известно восприятие цвета у человека трехцветное.

Основываясь на этом, в графических библиотеках используется именно такое трехкомпонентное представление цвета.

цвет задается с помощью функции

`Rgb (Red,Green,Blue)`

Здесь

- Red – число от 0 до 255, соответствующее интенсивности красной компоненты
- Green – число от 0 до 255, соответствующее интенсивности зеленой компоненты
- Blue – число от 0 до 255, соответствующее интенсивности синей компоненты

Графический модуль GraphABC

Задание стиля и цвета пера

- SetPenColor(цвет) – устанавливает цвет пера, задаваемый параметром color.
- SetPenWidth(ширина) – устанавливает ширину пера.
- SetPenStyle(стиль) – устанавливает стиль пера (сплошной, пунктир и т.п.),

ВОЗМОЖНЫЕ

значения указаны в таблице (стиль применим только к ширине пера 1 пиксел)

psSolid

psClear

psDash

psDot

psDashDot

psDashDotDot

Текущее перо используется для рисования линий, прямоугольников, ломаных, окружностей, эллипсов, дуг и т.п.

Графический модуль GraphABC

Пример программы: (выполните)

```
Program Gr02;  
Uses GraphABC;  
Var i : integer;  
begin  
  SetWindowSize(512,512);  
  SetPenStyle(psDash);  
  SetBrushStyle(bsClear);  
  Circle(256,256,WindowHeight div 2);  
end.
```

Графический модуль GraphABC

Задание стиля и цвета кисти

- `SetBrushColor(цвет)` – устанавливает цвет кисти
- `SetBrushPicture(имя файла)` – устанавливает в качестве образца для закраски кистью

образец, хранящийся в файле, при этом текущий цвет кисти при закраске игнорируется.

- `ClearBrushPicture` – очищает рисунок-образец, выбранный для кисти.
- `SetBrushStyle(стиль)` – устанавливает стиль кисти, задаваемый параметром `bs`.

Возможные стили кисти:

`bsSolid bsClear bsCross bsDiagCross bsHorizontal bsBDiagonal bsVertical bsFDiagonal`

Текущей кистью закрашиваются все замкнутые фигуры, контур фигур рисуется текущим пером.

Чтобы нарисовать не закрашенную фигуру, используйте `SetBrushColor(bsClear)`.

Графический модуль GraphABC

Пример программы: (выполните)

```
Program Gr03;  
Uses GraphABC;  
Var i,j : integer;  
begin  
  SetWindowSize (512, 512);  
  SetBrushColor (rgb (128, 0, 255));  
  SetBrushStyle (bsDiagCross);  
  Rectangle (10, 10, WindowWidth-10, WindowHeight-10);  
end.
```

Графический модуль GraphABC

$\text{Line}(x_1, y_1, x_2, y_2)$ – рисует отрезок с началом в точке (x_1, y_1) и концом в точке (x_2, y_2) .

- $\text{Circle}(x, y, r)$ – рисует окружность с центром в точке (x, y) и радиусом r .
- $\text{Ellipse}(x_1, y_1, x_2, y_2)$ – рисует эллипс, заданный своим описанным прямоугольником с координатами противоположных вершин (x_1, y_1) и (x_2, y_2) .
- $\text{Rectangle}(x_1, y_1, x_2, y_2)$ – рисует прямоугольник, заданный координатами противоположных вершин (x_1, y_1) и (x_2, y_2) .
- $\text{RoundRect}(x_1, y_1, x_2, y_2, w, h)$ – рисует прямоугольник со скругленными краями; (x_1, y_1) и (x_2, y_2) задают пару противоположных вершин, а w и h – ширину и высоту эллипса, используемого для скругления краев.

Графический модуль GraphABC

$Arc(x,y,r,a1,a2)$ – рисует дугу окружности с центром в точке (x,y) и радиусом r ,

заклученной между двумя лучами, образующими углы $a1$ и $a2$ с осью Ox ($a1$ и $a2$ – вещественные, задаются в градусах и отсчитываются против часовой стрелки).

- $Pie(x,y,r,a1,a2)$ – рисует сектор окружности, ограниченный дугой (параметры процедуры имеют тот же смысл, что и в процедуре Arc).
- $Chord(x,y,r,a1,a2)$ – рисует фигуру, ограниченную дугой окружности и отрезком, соединяющим ее концы (параметры процедуры имеют тот же смысл, что и в процедуре Arc).
- $FloodFill(x,y,color)$ – закрашивает область одного цвета, начиная с точки (x,y) цветом $color$.

Графический модуль GraphABC. Задание 1

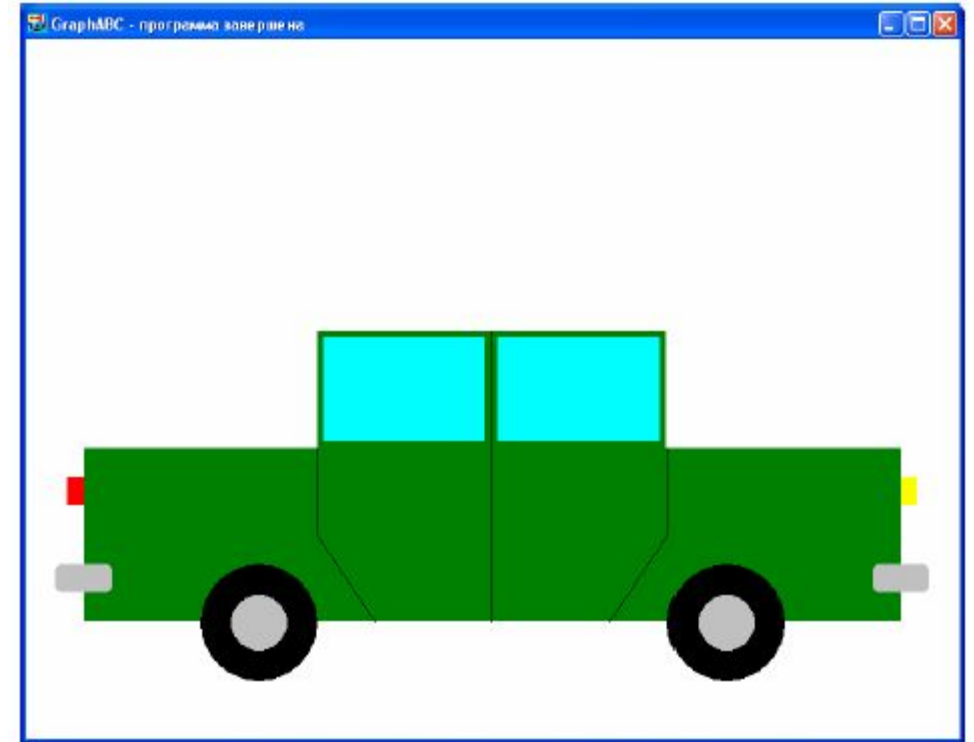
Напишите программу, использующую не менее трех процедур рисования, которая рисовала бы несложный рисунок (домик, автомобиль, и т.п.)

Графический модуль GraphABC.

Задание 1

Пример:

```
program avto;  
uses graphABC;  
var i : integer;  
begin  
SetWindowSize (800,600);  
SetPenStyle (psClear); // верхняя часть  
автомобиля  
setBrushColor (Clgreen);  
rectangle (250,250,550,351);  
setBrushColor (ClAqua); // окна  
rectangle (255,255,395,345);  
rectangle (405,255,545,345);
```



Графический модуль GraphABC.

Задание 1

Пример: (продолжение)

```
setBrushColor (Clgreen); // корпус
```

```
rectangle (50,350,750,500);
```

```
setBrushColor (ClYellow); // фары
```

```
rectangle (749,375,764,400);
```

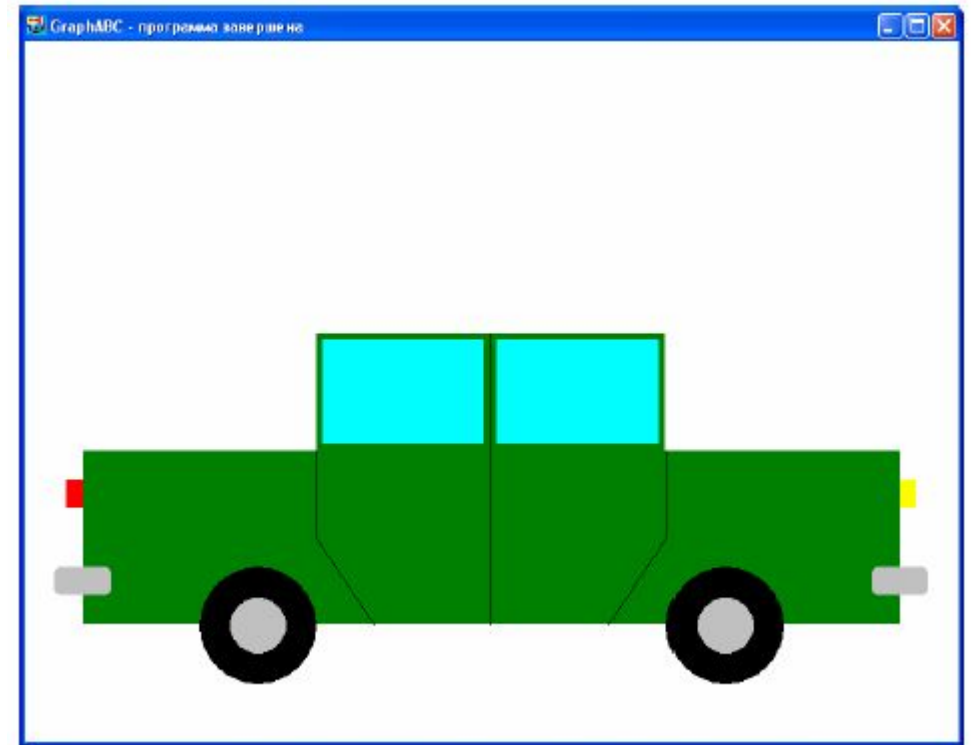
```
setBrushColor (ClRed);
```

```
rectangle (35,375,51,400);
```

```
setBrushColor (ClLightGray); // бампер
```

```
RoundRect (725,450,775,475,10,10);
```

```
RoundRect (25,450,75,475,10,10);
```

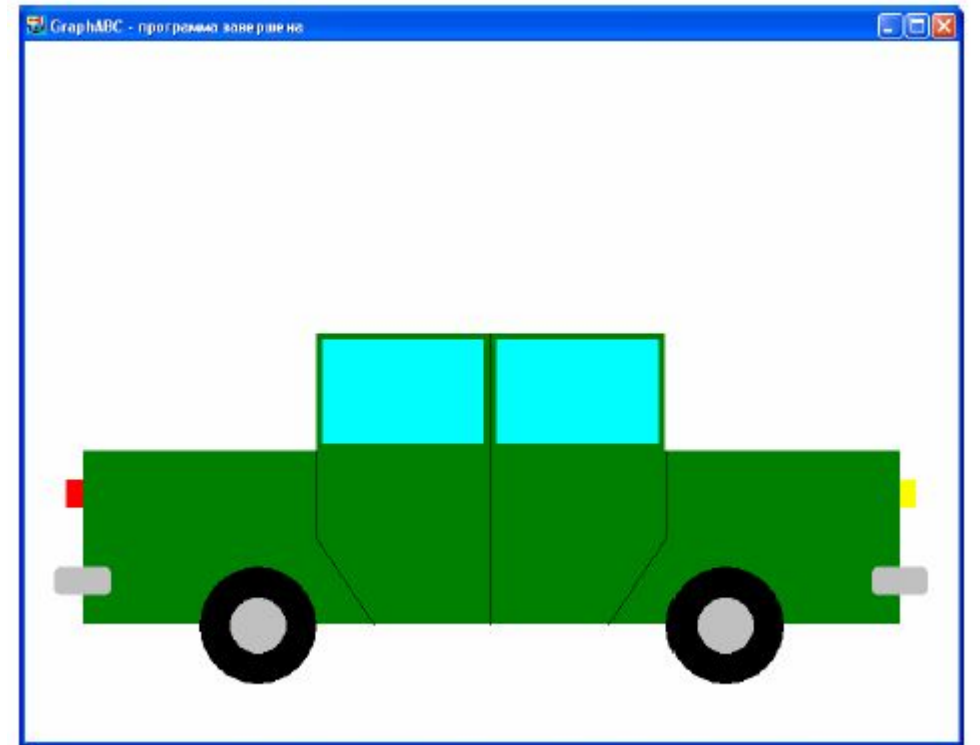


Графический модуль GraphABC.

Задание 1

Пример: (продолжение)

```
SetPenStyle (psSolid); // двери  
Line(250,350,250,425);  
Line(250,425,300,500);  
Line (400,250,400,500);  
Line (550,350,550,425);  
Line (550,425,500,500);  
setBrushColor (CIBlack); // колеса  
Circle (200,500,50);  
setBrushColor (CILightGray);  
Circle (200,500,25);
```



Графический модуль GraphABC. Задание 1

Пример: (продолжение)

```
setBrushColor (CIBlack);
```

```
Circle (600,500,50);
```

```
setBrushColor (CILightGray);
```

```
Circle (600,500,25);
```

```
end.
```

