



ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

УПРАВЛЕНИЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ
И ПОВЫШЕНИЯ КВАЛИФИКАЦИИ

Кафедра «Естественные науки»

ИНФОРМАТИКА

ГРАФИЧЕСКИЕ ВОЗМОЖНОСТИ ЯЗЫКА TURBO PASCAL

Методические указания к лабораторно-практическим работам
для иностранных слушателей дополнительных общеобразовательных
программ инженерно-технической и технологической направленности

Авторы

Моренко Б.Н.

Бабакова Л.Д.

Ростов-на-Дону, 2017



ОГЛАВЛЕНИЕ

	с.
Аннотация	3
1. Методические рекомендации	4
– Система координат монитора	4
– Изображение точки.....	6
– Изображение линии.....	7
– Изображение прямоугольника	7
– Изображение окружности	7
– Изображение дуги окружности	8
– Изображение параллелепипеда	8
– Изображение эллипса	9
2. Практическое задание	9
3. Задания для самостоятельной работы	11
4. Литература и Интернет-источники	11



АННОТАЦИЯ

Методические указания к лабораторно-практическим работам по информатике предназначены для иностранных слушателей дополнительных общеобразовательных программ инженерно-технической и технологической направленности обучения. Содержат рекомендации по использованию графических возможностей языка Turbo Pascal, а также дополнительные задания для самостоятельной работы.

Рекомендуются для лабораторно-практических занятий и самостоятельной работы по информатике.

Рецензент преп. Т.А. Тюрина

АВТОРЫ

Моренко Б.Н. – к.т.н., доцент кафедры
«Естественные науки»

Бабакова Л.Д. – доцент кафедры
«Русский язык как иностранный»

1. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

Монитор современного компьютера может работать в двух режимах: текстовом и графическом.

В текстовом режиме работы экран монитора разбивается на отдельные ячейки (знакоместа), в каждой из которых может быть отображён один символ (буква, цифра или знак). Изображение символа формируется в прямоугольной ячейке, которая имеет размеры 8x16 пикселей. Пиксель (от англ. pixel – Picture Element) это – минимальный элемент изображения на экране монитора. В текстовом режиме позиция верхнего левого угла экрана определяется координатами (1,1). Положение символа на экране задаётся двумя координатами – вертикальной (Y) и горизонтальной (X), где X – это номер позиции в строке, а Y – номер строки. Чаще всего на экране монитора, который работает в текстовом режиме, можно поместить 25 строк по 80 позиций (символов) в каждой.

При работе в графическом режиме весь экран монитора разбивается на отдельные точки – «пиксели», которые являются наименьшими элементами графического изображения. Положение пикселя задаётся двумя координатами – X и Y (Рис.1). Координата X увеличивается слева направо, а координата Y – сверху вниз. Следует помнить, что координаты X и Y могут принимать только целочисленные значения. Позиция верхнего левого угла определяется координатами (0, 0).



Рис. 1. Система координат монитора

Количество точек по горизонтали и вертикали определяет разрешающую способность монитора. Например, выражение «Разрешающая способность экрана монитора 640 x 480» означает, что монитор в данном режиме выводит 640 точек по горизон-

тали и 480 точек по вертикали. Чем больше разрешающая способность монитора, тем выше качество изображения. Геометрические размеры пикселя определяются разрешением монитора, которое задается количеством пикселей по горизонтали и вертикали, например, 640x480, 800x600, 1024x768, 1280x1024 и др. Любое графическое изображение на экране монитора формируется из отдельных пикселей. Управляя расположением точек (пикселей) на экране монитора можно создавать графические изображения любой сложности. Управляет работой монитора специальное устройство компьютера – видеоадаптер. Видеоадаптер управляет каждым пикселем на экране монитора.

Для работы в графическом режиме его нужно сначала инициализировать. Делается это с помощью процедуры *InitGraph* (*GraphDriver*, *GraphMode*: *Integer*, *Path*: *String*);

GraphDriver – параметр, который задаёт тип графического драйвера, то есть специальной программы, которая управляет экраном в графическом режиме.

GraphMode – параметр, который определяет режим работы с выбранным драйвером. Режим работы определяет количество точек на экране (разрешение), палитру цветов и т.д.

Path – параметр, который содержит путь к нужному драйверу, т.е. указывает, в каком каталоге находится драйвер.

При работе монитора в графическом режиме может быть осуществлён также и вывод текстовой информации. Для этого используют процедуру *OutTextXY* (*X*, *Y*, *TextString*),

где *X*, *Y* – координаты точки начала вывода текста,

TextString – константа или переменная строкового типа *string*.

Например, для вывода строки «Графический режим работы монитора» с точки (10, 100) в тексте программы нужно записать:

OutTextXY(10, 100, 'Графический режим работы монитора');

В языке программирования Turbo Pascal имеется справочная система, которая позволяет программисту получать подробное описание стандартных подпрограмм с примерами их применения.

Графические возможности языка Turbo Pascal сосредоточены в стандартных библиотеках [Graph](#) и [CRT](#) ([Cathode Ray Tube](#)). Библиотека [Graph](#) содержит константы, процедуры и функции для управления графическим режимом работы монитора.

Для инициализации графического режима работы используется процедура *InitGraph*. В этой процедуре указываются параметры, которые определяют графический драйвер и путь к нему, задают нужный программисту режим работы монитора.

Процедура *CloseGraph* – восстанавливает текстовый режим работы монитора, который использовался до перехода в графический режим.

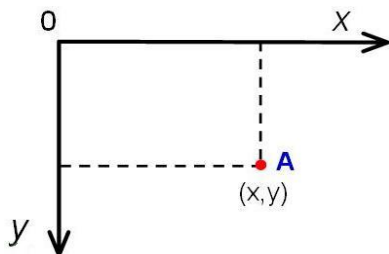
Процедура *SetColor (Color: Word)* – устанавливает текущий цвет, который можно выбрать из прилагаемой таблицы (Табл.1).

Таблица 1

КОНСТАНТЫ И КОДЫ ЦВЕТОВ

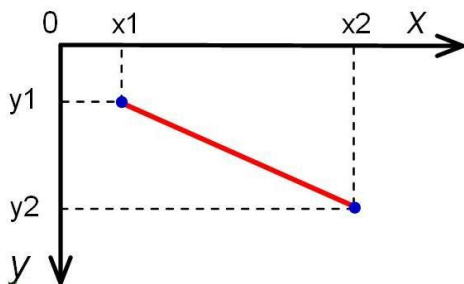
Имя константы	Код	Цвет
Black	0	Чёрный
Blue	1	Тёмно-синий
Green	2	Тёмно-зелёный
Cyan	3	Бирюзовый
Red	4	Красный
Magenta	5	Фиолетовый
Brown	6	Коричневый
LightGray	7	Светло-серый
DarkGray	8	Темно-серый
LightBlue	9	Синий
LightGreen	10	Светло-зелёный
LightCyan	11	Светло-бирюзовый
LightRed	12	Розовый
LightMagenta	13	Малиновый
Yellow	14	Жёлтый
White	15	Белый

Для изображения точки на экране монитора используют процедуру *PutPixel (X, Y: Integer; Color : Word)*,

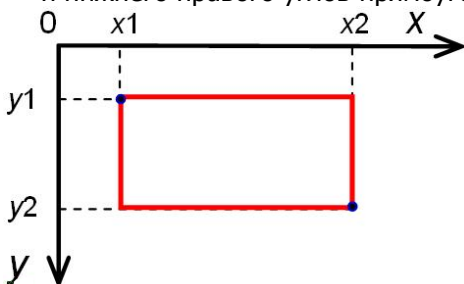


где x и y – координаты точки A относительно верхнего левого угла экрана монитора; $Color$ – код цвета точки.

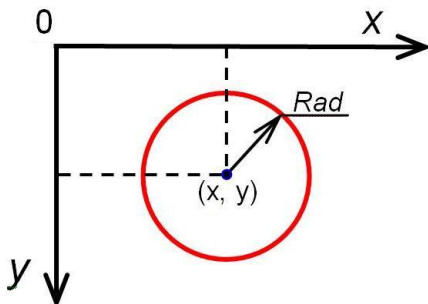
Для изображения на экране монитора линии используют процедуру *PutPixel (X1, Y1, X2, Y2: Integer)*, где $(x1, y1)$ и $(x2, y2)$ – координаты начала и конца линии.



Для изображения на экране монитора прямоугольника используют процедуру *Rectangle (X1, Y1, X2, Y2: Integer)*, где $(x1, y1)$ и $(x2, y2)$ – координаты верхнего левого и нижнего правого углов прямоугольника.



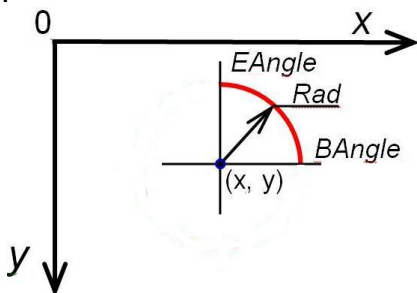
Для изображения на экране монитора окружности используют процедуру *Circle (X, Y: Integer; Rad: Word)*, где (x, y) – координаты центра окружности, *Rad* – радиус окружности в пикселях.



Для изображения на экране монитора дуги окружности используют процедуру *Arc* (*X, Y: Integer; BAngle, EAngle, Rad: Word*),

где (x, y) – координаты центра, *BAngle*, *EAngle* – начальный и конечный углы дуги, *Rad* – радиус дуги в пикселях.

Примечание: Углы для процедуры *Arc* отсчитываются против часовой стрелки.

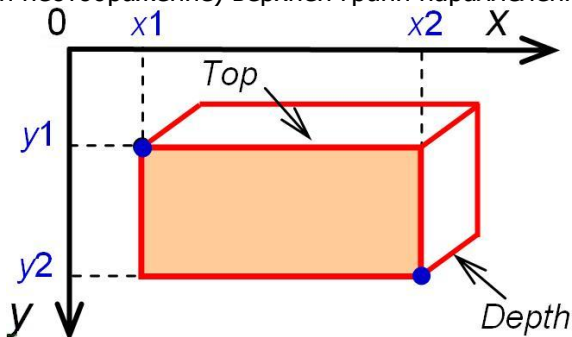


Для изображения на экране монитора параллелепипеда с заполненной передней гранью используют процедуру:

Bar3D (*X1, Y1, X2, Y2: Integer; Depth: Word; Top: Boolean*),

где $(X1, Y1)$ и $(X2, Y2)$ – координаты верхнего левого и нижнего правого углов прямоугольника,

Depth – параметр, который определяет «глубину» параллелепипеда (в пикселях), *Top* – параметр, который задаёт отображение (или неотображение) верхней грани параллелепипеда.



При заполнении передней грани параллелепипеда (или прямоугольника) используют шаблон и цвет, которые определяются процедурой *SetFillStyle* (*Pattern: Word; Color: Word*).

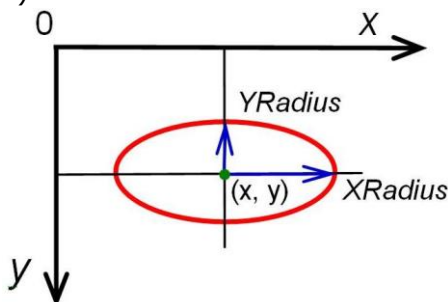
где *Pattern* – шаблон заполнения:

EmptyFill – сплошное заполнение цветом фона (код – 0),

SolidFill – сплошное заполнение текущим цветом (код – 1),
LineFill – заполнение горизонтальными линиями (код – 2),
SlashFill – заполнение наклонными линиями (код – 4),
Color – цвет заполнения передней грани параллелепипеда.

Для изображения на экране монитора эллипса используют процедуру *Ellipse* (*X, Y: Integer; BAngle, EAngle: Word; XRadius, YRadius: Word*),

где (*x, y*) – координаты центра, *BAngle, Eangle* – начальный и конечный углы дуги, *XRadius, YRadius* – соответственно горизонтальный и вертикальный радиусы эллипса (в пикселях).



2. Практическое задание. Написать программу построения на экране монитора геометрических примитивов и фигур: точки, прямой линии, прямоугольника, заштрихованного прямоугольника, окружности, полуокружности, эллипсов, параллелепипеда.

```

program Graph_Demo;
    Uses
        Crt, Graph;
var
    GrDriver : integer;
    GrMode : integer;
begin
    ClrScr;
    GrDriver := Detect;    {Инициализация графического режима}
    InitGraph (GrDriver, GrMode, 'c:\tp\bgi');
    If GraphResult < > grOk Then Halt (1);
    
```



{Изображение точки}

PutPixel (300, 100, Green); {Координаты и цвет точки}

{Рисование линии заданного цвета}

SetColor (Red); {Цвет линии}
Line (100, 180, 550, 180); {Координаты начала и конца линии}

{Рисование прямоугольника}

SetColor (LightRed); {Цвет рамки прямоугольника}
Rectangle (10, 10, 630, 460); {Координаты углов прямоугольника}

{Рисование окружности}

SetColor (Brown); {Цвет окружности}
Circle (300, 100, 60); {Координаты центра и радиус окружности}

{Рисование полуокружности}

SetColor (Yellow); {Цвет контура полуокружности}
Arc (300, 100, 300, 240, 50); {Координаты центра, углов и радиуса}

{Рисование параллелепипеда}

SetFillStyle (8, Red); {Тип и цвет заполнения передней грани
параллелепипеда}
Bar3D (150, 240, 480, 350, 20, TopOn);

{Рисование заштрихованного прямоугольника}

SetColor (Green); {Цвет контура прямоугольника}
Rectangle (100, 410, 550, 450); {Координаты углов прямоугольника}
SetColor (5); {Цвет штриховки прямоугольника}
SetFillStyle (11, Yellow); {Тип штриховки прямоугольника}
Bar (101, 411, 549, 449);

{Рисование эллипсов}

SetColor (LightBlue); {Цвет контура эллипса}
Ellipse (100, 100, 0, 360, 30, 50); {Большая ось – вертикальная}
Ellipse (500, 100, 0, 360, 50, 30); {Большая ось – горизонтальная}
Repeat Until KeyPressed; {Ожидание нажатия любой клавиши}
CloseGraph; {Закрытие графического режима}

end.

3. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Составьте программу изображения окружности зелёного цвета радиусом 100 пикселей в центре экрана монитора. Центр окружности – красная точка.
2. Составьте программу построения изображения компакт-диска.
3. Составьте программу построения прямоугольника, который отстоит от границ экрана монитора на 20 пикселей. Цвет контура прямоугольника – жёлтый. Цвет заливки прямоугольника – зелёный.
4. Составьте программу построения изображения монитора.
5. Составьте программу построения изображения системного блока.
6. Составьте программу построения изображения принтера.
7. Составьте программу построения изображения прямоугольного треугольника.
8. Составьте программу построения изображения жёсткого диска.
9. Составьте программу построения изображения флешки.
10. Составьте программу построения изображения здания нашего университета.

4. ЛИТЕРАТУРА И ИНТЕРНЕТ-ИСТОЧНИКИ

1. Моренко Б.Н. Компьютерная графика в Turbo Pascal. Методические указания для практических работ по информатике / Б.Н. Моренко. – Ростов н/Д: Издательский центр ДГТУ, 2010. – 8 с.
2. Шауцукова Л.З. Информатика. Теория (с задачами и решениями). Режим доступа: <http://book.kbsu.ru/theory/index.html>
3. Шпак Ю.А. Turbo Pascal. Просто как дважды два / Ю.А. Шпак. – М.: Эксмо, 2008. – 400 с.
4. Меженный О.А. Turbo Pascal: учитесь программировать / О.А. Меженный – М.: Издательский дом "Вильямс", 2001. – 448 с.: ил.
5. Культин Н.Б. Turbo Pascal в задачах и примерах / Н.Б. Культин. – СПб.: БХВ - Санкт-Петербург, 2000. – 256 с.: ил.



6. Соболев Б.В. Информатика: Учебник / Б.В. Соболев
А.Б. Галин, Ю.В. Панов, Е.В. Рашидова, Н.Н. Садовой. – Ростов
н/Д: Феникс, 2005. – 448 с.