

Лабораторная работа N1.

Построение графиков функций.

Модуль *Graph* представляет собой мощную библиотеку графических подпрограмм универсального назначения, рассчитанную на работу с наиболее распространенными графическими адаптерами. Подпрограммы модуля *Graph* обеспечивают различные режимы работы многорежимных адаптеров, полностью используют их цветовые возможности и разрешающую способность.

При исполнении графических программ требуется автономный драйвер графического адаптера (BGI - файл). Если программа использует штриховые шрифты, то кроме того нужен один или несколько шрифтовых файлов (CHR - файлы). При необходимости драйвер и шрифты могут быть включены в тело программы еще на этапе компиляции.

Во время выполнения программы процедура *InitGraph* автоматически распознает графический адаптер, установленный на ПК, загружает и инициализирует соответствующий графический драйвер, переводит адаптер в графический режим и возвращает управление вызывающей программе. Процедура *CloseGraph* выгружает драйвер из памяти и восстанавливает текстовый режим работы адаптера.

Подпрограммы модуля *Graph* позволяют адресоваться к любому элементу (пикселю) растрового графического экрана и управлять светимостью этого элемента. Для указания пикселя используется следующая система координат: верхний левый угол графического экрана имеет координаты 0,0; горизонтальная координата X увеличивается слева направо, вертикальная координата Y увеличивается сверху вниз. Например, в режиме 640x480 (адаптер VGA) правый нижний угол экрана имеет координаты 639, 479, а центр экрана - координаты 319, 239.

Некоторые графические подпрограммы используют понятие текущего указателя. Указатель содержит координаты того пикселя, начиная с которого будет строиться изображение подпрограммами *LineTo*, *LineRel*, *OutText* и др. В этом смысле указатель подобен текстовому курсору, но, в отличие от него, не имеет видимого изображения на экране.

Для вывода текстовых сообщений на графический экран модуль *Graph* предоставляет один матричный и 4 штриховых шрифтов. Каждый символ матричного шрифта на экране реализуется в виде матрицы из 808 пикселей. Штриховые шрифты для каждого символа определяют набор штрихов (векторов), с помощью которых на экране создается (вычерчивается) соответствующий символ. Штриховые шрифты позволяют изменять размеры текстовых надписей в широких пределах без существенного ухудшения качества изображения символов. Однако стандартные штриховые шрифты не содержат символы кириллицы.

В модуле *Graph* имеется несколько процедур для рисования элементарных графических фигур - точек, линий, окружностей, дуг и т.п. При необходимости замкнутые фигуры могут быть закрашены различными цветами и стилями (образцами закрашки).

Процедура *SetViewPort* создает на экране графическое окно. Если окно определено, весь дальнейший графический вывод осуществляется относительно координат этого окна и отсекается его границами.

В модуле имеются средства сохранения и выдачи изображений, работы с несколькими графическими страницами, установки необходимых цветов.

Некоторые процедуры модуля *Graph*:

Circle (*X*, *Y*: integer; *R*: word) Рисует окружность радиуса *R*, используя *X*, *Y* как координаты центра.

ClearDevice Очищает экран.

ClearViewPort Очищает окно.

CloseGraph Закрывает графический режим.

DetectGraph (*var Driver*, *Mode*: integer) Возвращает тип *Driver* установленного драйвера и текущий режим *Mode* его работы.

DrawPoly(*NumPoints*: word; *var PolyPoints*) Рисует многоугольник из *NumPoints* вершин с координатами в *PolyPoints* текущим цветом и типом линии.

FillPoly(*NumPoints*: word; *var Polypoints*) Рисует и штрихует многоугольник, содержащий *NumPoints* вершин с координатами в *PolyPoints*.

GetImage(*X1*, *Y1*, *X2*, *Y2*: integer; *var BitMap*) Сохраняет в переменной *BitMap* битовый образ указанной части экрана.

Компьютерная графика.

Лабораторные работы.

GraphDefaults Устанавливает стандартные параметры графического режима.

InitGraph(var Driver, Mode: integer; Path: string) Инициализирует графический режим. Переменные *Driver* и *Mode* должны содержать тип графического драйвера и его режим работы. Допускается указать *Driver=0* для автоматического определения этих параметров по результатам тестирования аппаратуры. Параметр *Path* определяет маршрут поиска файла графического драйвера.

Line(X1, Y1, X2, Y2: integer) Рисует линию от точки X_1, Y_1 до точки X_2, Y_2 .

LineRel(DX, DY: integer) Рисует линию от текущего указателя к точке, заданной приращением координат.

LineTo(X, Y: integer) Рисует линию от текущего указателя к точке.

MoveRel(DX, DY) Смещает текущий указатель к точке, заданной приращением координат.

MoveTo(X, Y: integer) Смещает текущий указатель к точке X, Y .

OutText(TextString: string) Выводит текстовую строку на экран.

OutTextXY(X, Y: integer, TextString: string) Выводит текст в заданное место экрана.

PutImage(X, Y: integer, var BitMap; BitBlt: word) Выводит битовый образ на экран.

PutPixel(X, Y: integer, Color: word) Выводит точку цветом *Color* с координатами X, Y .

Rectangle(X1, Y1, X2, Y2: integer) Рисует прямоугольник, используя текущий цвет и тип линии.

SetBkColor(Color: word) Устанавливает цвет фона.

SetColor(Color: word) Устанавливает основной цвет, которым будет осуществляться рисование.

SetLineStyle(LineStyle, Pattern, Thickness: word) Устанавливает толщину и стиль линии.

SetViewport(X1, Y1, X2, Y2: integer, ClipOn: Boolean) Устанавливает текущее окно для графического вывода.

Некоторые функции модуля *Graph*:

GetX: integer Возвращает координату X текущего указателя.

GetY: integer Возвращает координату Y текущего указателя.

GraphErrorMsg(ErrorCode: integer): string Возвращает строку сообщения об ошибке для заданного кода *ErrorCode*.

Компьютерная графика.

Лабораторные работы.

GraphResult: integer Возвращает код ошибки для последней графической операции.

Более полное описание этих и других процедур и функций можно найти в справочной системе среды Pascal или в соответствующей литературе.

Задание.

Написать программу вычерчивающую графики заданных функций в одной системе координат.

Варианты заданий.

Номер варианта	Функции	Начальное значение x	Конечное значение x
1	$\ln x$ и $\lg x$	1	20
2	$\sin e^{-x}$ и $\cos e^{-x}$	0,05	1
3	$\arcsin e^{-x}$ и $\arccos 10^{-x}$	2	5
4	$\sqrt[3]{x}$ и $\sqrt[5]{x}$	20	100
5	e^{-x} и 10^{-x}	1	2
6	$\operatorname{tg}^3 x$ и $\operatorname{ctg}^3 x$	0,6	1,5
7	$e^x \sin x$ и $e^{-x} \sin x$	0	1
8	$\sqrt{\ln x}$ и $\sqrt[4]{\ln x}$	1	5
9	$\cos \sqrt{x}$ и $\operatorname{tg} \sqrt{x}$	0,2	2
10	10^{-x} и $10^{\sqrt{x}}$	0,1	2

Номер варианта	Функции	Начальное значение x	Конечное значение x
11	$\arcsin \frac{1}{x^2} \text{ и } \arccos \frac{1}{x^3}$	1,05	2
12	$\ln \frac{1}{x} \text{ и } \lg \frac{1}{x}$	0,1	1
13	$\sin \frac{1}{\sqrt{x}} \text{ и } \cos \frac{1}{\sqrt{x}}$	0,2	2
14	$x^{0,3(\pi+1)} \text{ и } x^{(e-1)}$	0,5	4
15	$e^x \operatorname{tg} x \text{ и } e^{-x} \operatorname{ctg} x$	0,05	1