Модуль CRT

Функции управления клавиатурой

KeyPressed: Boolean - определяет факт нажатия на любую клавишу

True

ReadKey: Char - читает код нажатой клавиши

Позиционирование курсора

WhereX: Byte - возвращает горизонтальную координату

WhereY: Byte - возвращает вертикальную координату

GotoXY(X,Y:byte); - перемещает курсор

Управление звуковым генератором

Sound(F:word); - включает звук генератора, F - частота

NoSound; - выключает звук генератора.

Работа с экраном

ClrScr; - очищает экран

Window(X1,Y1,X2,Y2:byte); - определяет размеры окна

TextMode(Mode:word); - устанавливает нужный текстовый режим: 0,1,2...

Работа со строками

ClrEol; - удаляет все символы справа от курсора до конца строки

InsLine; - вставляет пустую строку

DelLine; - удаляет текущую строку.

Настройка цвета

TextColor(Color:byte); - устанавливает цвет символа (0..15).
TextBackGround(Color:byte); - устанавливает цвет фона экрана
HighVideo; - устанавливает высокую яркость символов.
LowVideo; - устанавливает низкую яркость символов.
NormVideo; - устанавливает нормальную яркость символов.

Процедуры управления дисплеем

AssignCrt(Var F:Text); - связывает с файловой переменной устройство CON (клавиатуру для ввода и дисплей для вывода).

Delay(D: word); - приостанавливает работу программы

Цвета:

0-черный
1-синий
2-зеленый
3-голубой
4-красный
5- фиолетовый
6- коричневый
7- светло-серый
8- темно-серый
9- светло-синий
10- светло-зеленый
11- светло-голубой
12-розовый
13-светло-фиолетовый (малиновый)
14-желтый
15-белый.
128-мерцание

Пример:

```
Program Text;
Uses CRT;
Begin
      ClrScr;                очистка экрана
      TextBackGround(10);  яркозеленый фон
      Sound(12500);     включить звук
      TextColor(5);              фиолетовые символы
      GotoXY(20,10);                {10 строка, 20 позиция}
      WriteLn('Включен звук !');
      Delay(1000);
      NoSound      выключить звук
         Repeat  Until  KeyPressed;  {задержка до нажатия на любую клавишу}
End.
```

Модуль GRAPH

.BGI GRAPH.TPU.

PROGRAM **Имя**;

USES GRAPH;

описание меток, переменных программы, в частности

VAR driver, Mode: integer;

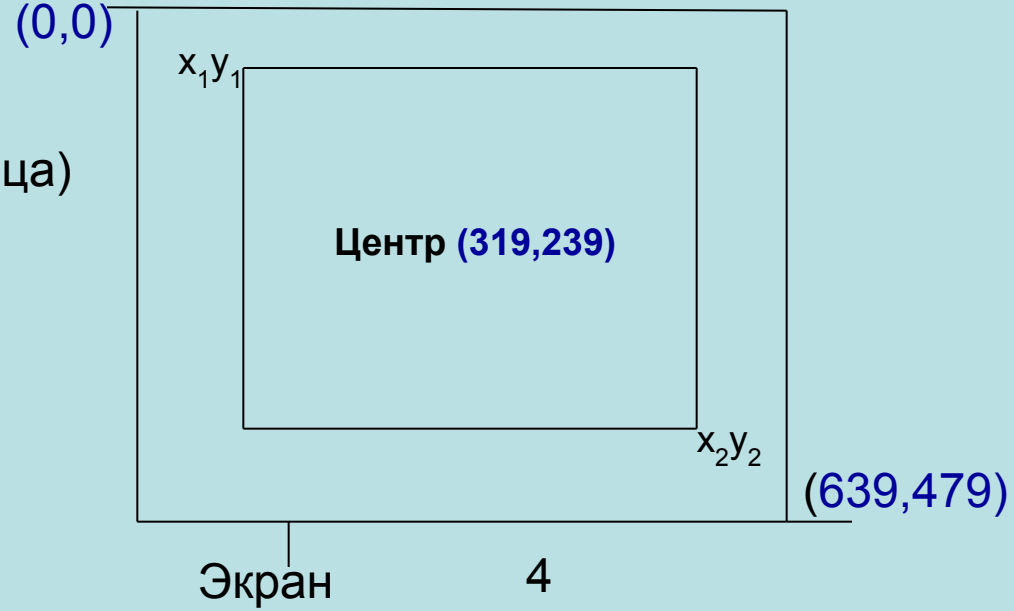
Begin

InitGraph(<тип драйвера>,<видеорежим>:integer;
<путь к графическому драйверу>:string); {Инициализация графич.режима}

Графическая информация на экране дисплея отражается совокупностью светящихся точек - ПИКСЕЛЕЙ.

Режим по вертикали: 0 - 200 пикселей (4 страницы),
1 - 350 пикселей (2 страницы),
2 - 480 пикселей (1 страница).

Режим 2 (640x480, 16 цветов,1 страница)



{Задержка графического режима}

CloseGraph

end.

DetectGraph(var Driver, Mode:integer); - возвращает тип драйвера и текущий режим его работы.

0 <= Driver <= 10 ,

0<= Mode <= 5 (Зависит от драйвера)

```
PROGRAM
USES GRAPH;
VAR a, b: integer;
BEGIN
    DetectGraph(a,b) ;
    initgraph (a, b,`C:\progDOS\BP\BGI`);
```

-

НЕКОТОРЫЕ ПРОЦЕДУРЫ И ФУНКЦИИ модуля GRAPH

установочные

ClearDevice; - Очищает экран

SetViewPort(x1,y1,x2,y2:integer; Clip:boolean); - Устанавливает текущее окно

Если **Clip=true**, то все изображения отсекаются на границах вывода;

ClearViewPort; - Очищает текущее окно;

GetMaxX:integer - Возвращает максимальную горизонтальную координату;

GetMaxY:integer - Возвращает максимальную вертикальную координату;

GetX:integer - Возвращает координату X текущего указателя в окне;

GetY:integer - Возвращает координату Y текущего указателя в окне;

SetLineStyle(Line,Pattern,Thickness:word); - Устанавливает стиль (0..4), шаблон штриховки (0..12) и толщину (1-норм, 3-утроенная);

SetFillStyle(Pattern,Color:word); - Устанавливает образец штриховки и цвет (0..15 и 128-мерцание);

SetGraphMode(Mode:integer); - Устанавливает новый графический режим и очищает экран;

SetColor(Color:word); - Устанавливает основной цвет, которым выполняется рисование (0..15);

SetBkColor(Color:word); - Установка цвета фона.

-

Графические примитивы

PutPixel(X,Y:integer;Color:word); - Выводит точку цветом Color с координатой X,Y;

LineTO(X,Y:integer); - Рисует линию от текущего указателя к точке с координатой (X,У);

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

Line(X1,Y1,X2,Y2:integer); - Рисует линию от точки (X1,Y1) к точке с координатой (X2,У2;)

MoveTO(X,Y:integer); - Смещает текущий указатель к точке с координатой X,У;

MoveRel(DX,DY:integer); - Смещает текущий указатель к точке, заданной приращением координат;

Rectangle(X1,Y1,X2,Y2:integer); - Рисует прямоугольник, используя текущий цвет и тип линии по верхней левой и нижней правой точкам;

Bar(X1,Y1,X2,Y2:integer); - Рисует закрашенный прямоугольник, используя установку SetFillStyle;

–

Bar3D(X1,Y1,X2,Y2:integer;Depth:word;Top:Boolean); - Рисует закрашенный параллелепипед. Depth - глубина в Pixel (1/4 ширины).

Если Top=True, то рисуется верхняя грань параллелепипеда;

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

Fillellipse(X,Y:integer;XR,YR:word); - Рисует защтрихованный эллипс, используя X,Y как центр и XR,YR как горизонтальный и вертикальный радиусы.

RestoreCRTMode; - Восстанавливает текстовый режим работы экрана;

OutText(Text:string); - Выводит текстовую строку на экран.

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

SetTextStyle(Font, Direction; CharSize: Word);

Устанавливает вид шрифта, стиль(0 - горизонтальный, 1 - вертикальный) и размер шрифта (0..10)

0 - матричный (по умолчанию)
1 - полужирный
2 - светлый (тонкий)
3 - книжный (рубленный)
4 - готический

```
program graphik;                          { y=2+x*x          файл grafik.pas}
uses graph;   {подлючение модуля GRAPF}
var a,b,x,y:integer;  {a,b-переменные, определяющие тип драйвера и
                              видеорежим соответственно}
begin
      detectgraph(a,b);  {автоопределение типа драйвера и видеорежима}
        initgraph(a,b,'c:\ProgDos\BP\BGI'); {инициализация графического режима}
      setgraphmode(2);   {установка нового режима (2), очистка эрана}
      setBKcolor(0); {основной цвет фона – черный}
      setcolor(4);     {основной цвет рисования – красный}
      moveTO(100,100);  {смещает курсор}
      lineRel(50,50); {вычерчивание линии по приращению от текущего
                                          указателя}
            readln; {пауза до нажатия клавиши <ввод>}
setcolor(2);      {основной цвет рисования – зеленый}
line(100,100,100,50); рисует линию от т.X1,У2 до т.X2,У2
            readln; {пауза до нажатия <ввод>}
setcolor(5);    {основной цвет рисования – фиолетовый}
putPixel(0,0,12);   {выводит точку цветом 12 с координатой 0,0}
            readln; {пауза до нажатия <ввод>}
moveTO(0,2); {смещает курсор к точке с координатой 0,2}
```

```
 for x:=0    to   20   do
            begin
                        y:=2+ x*x;
                        lineTO(x,y); {рисует линию от текущего указателя к
точке с координатой X,У}
            end;
      readln; {пауза до нажатия ввода}
 Settextstyle(1,0,5);
 OutTextXY(50,400,'график функции y=2+x*x'); {выводит текст в заданное место
                                              экрана}
     readln; {пауза до нажатия <ввод>}
  putPixel(639,479,14); {выводит точку цветом 14 с координатой 639,479}
     readln; {пауза до нажатия <ввод>}
  closegraph; {закрывает графический режим и восстанавливает текстовый}
 end.
```

Для КР в ауд. 1-266 каталог V1_03k\

Вопросы к экзаменам (1 семестр) в файле vopr05_1

Если X и Y вещественные , то их нужно преобразовать к целым координатам:

LineTo(trunc(x),trunc(y));

При необходимости применяют масштабирование и смещение.

В метод.указаниях к лаб.раб. 2005 приведена программа построения графика циклоиды, в которой рассчитываются $x = a\,t - b \sin t$, $y = a - b \cos t$.

Фрагмент программы для расчета и построения графика

```
moveTO(10,240);  {смещение курсора к точке с координатой (10,240)}
              tt:=0.0;
for t:=1 to 200 do            {расчет и вычерчивание циклоиды}
               begin
                  x:= 0.4*tt - 0.5*sin(tt);
                  y:= 0.4 - 0.5*cos(tt);
 {линия}        lineTO(trunc(100*x)+10,trunc(100*y)+240);
                  tt:=tt+Pi/32
               end;
```

```
{Пример программы, рисующей  переливающийся круг, пока не нажата
                              любая клавища}
PROGRAM I_13;
Uses Graph, crt;
Var     a,b,x:integer;
begin
 detectgraph(a,b);
 initgraph(a,b,'e:\tp\bgi');
 setgraphmode(2);
   clearDevice;   {очистка экрана}
 setBKcolor(0);     {цвет фона}
 repeat
 for x:=0 to 100 do
  begin
   if keypressed then break;
   setcolor(x);          {цвет рисования}
   circle(320,240,30);
   delay(50);
   end;
   Until Keypressed;
   closegraph;
end.
```

Задание для РГР:

разработать схему алгоритма, написать и отладить программу

1) для расчета и построения графиков двух функций (результаты расчетов должны храниться в виде массивов и распечатываться в виде таблицы), цветом необходимо выделить наибольшее и наименьшее значения для каждой из функций.

2) для решения нелинейного уравнения или вычисления приближенного значения определенного интеграла по заданию преподавателя.

Отчет по РГР

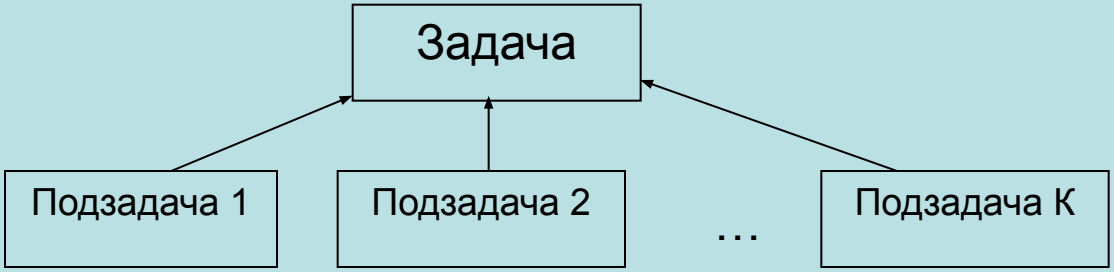
печатается в текстовом редакторе Word содержит помимо титульного листа:

1. Задание.
2. Математическая формулировка задачи.
3. Детальная схема общего алгоритма.
4. Тексты программ на Турбо Паскале.
5. Таблица результатов с выделением макс. и мин. значений для каждой функции, выполненная с помощью табличного процессора Excel. Расчеты коэффициентов для масштабирования функций.
6. Графики функций, напечатанные через Excel.
7. Заданное нелинейное уравнение, схему алгоритма, текст программы и результат его решения.
8. Интеграл, метод решения, алгоритм, программа и результат.

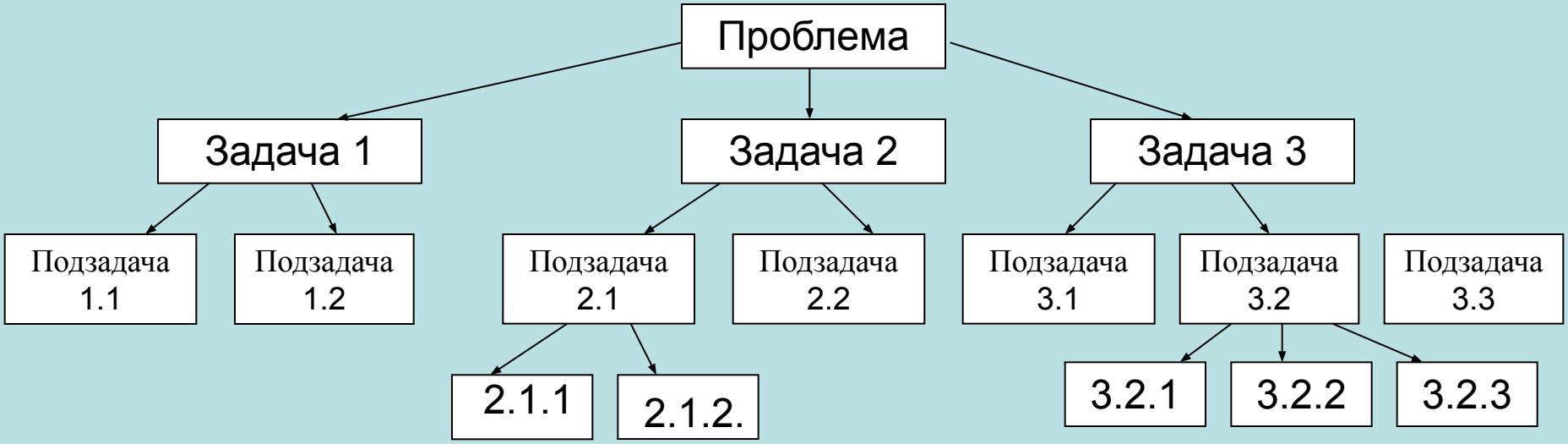
Программные средства демонстрируются с дискеты

Структуризация

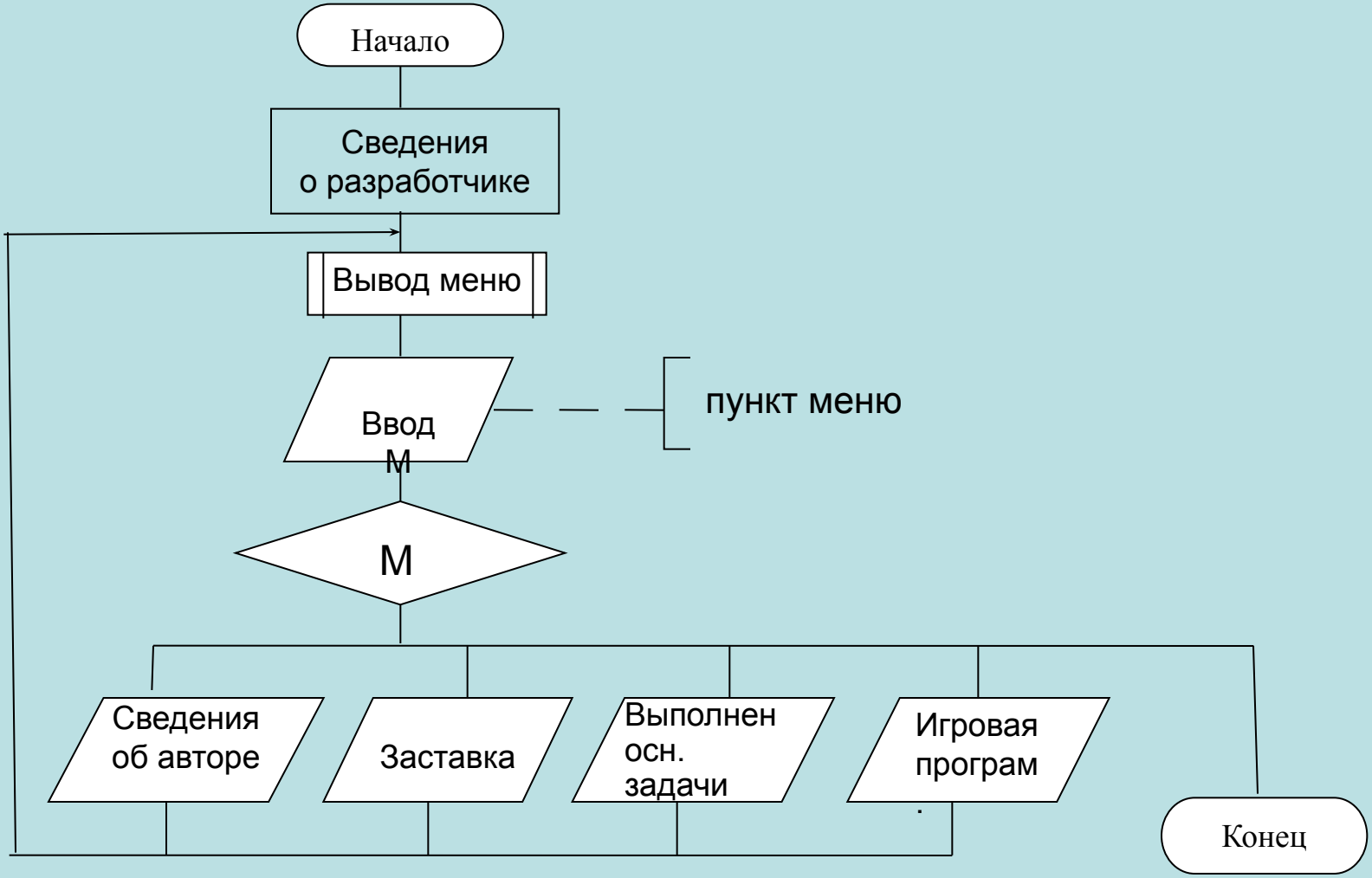
Восходящее проектирование алгоритмов и программ



Нисходящее программирование («сверху - вниз)



Методы проектирования программ // Шафеева О.П. 2004. -32с.
(Метод. указания к курсовой работе и РГР)



Укрупненная схема алгоритма для курсовой работы