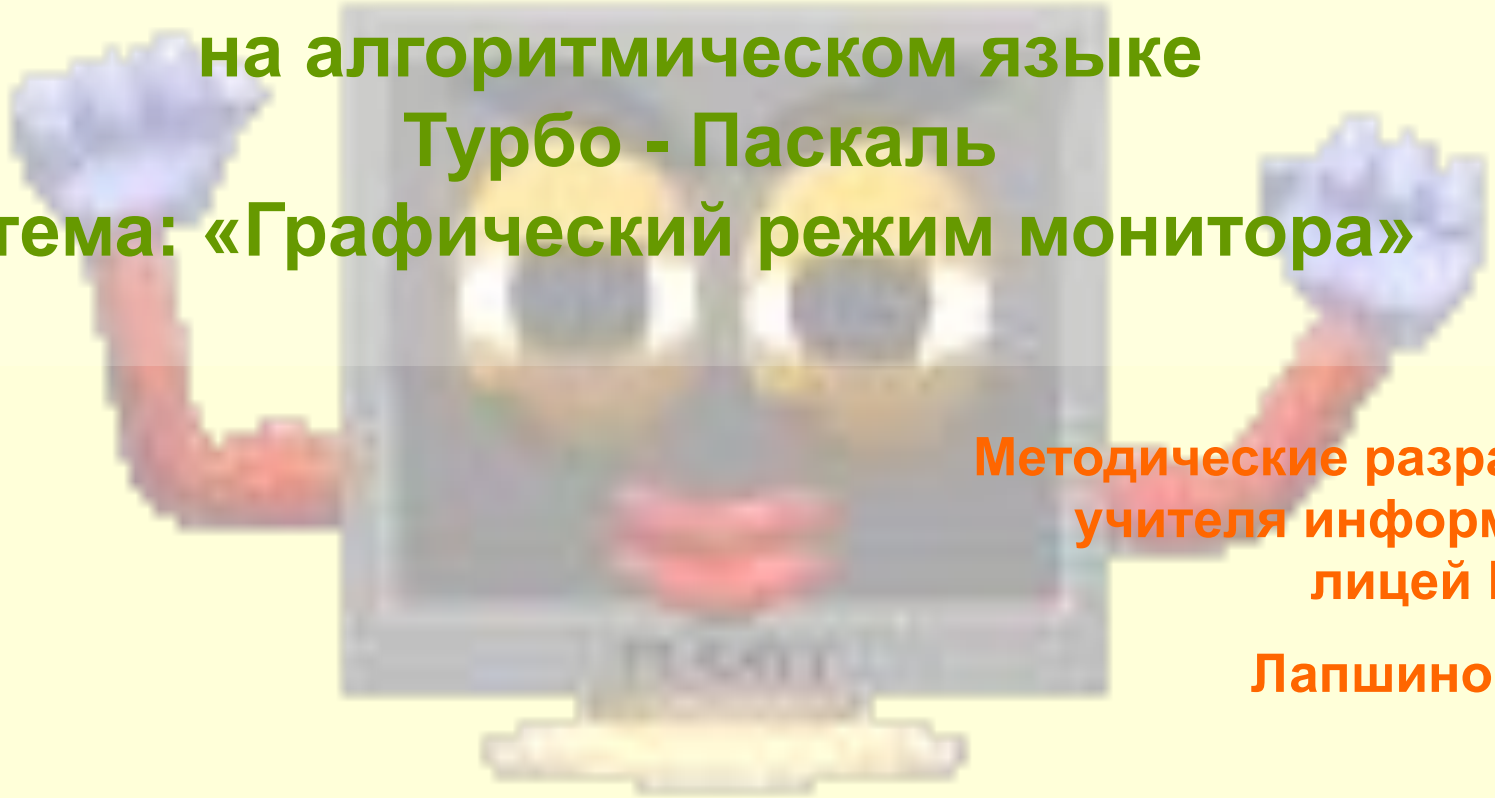
Программирование

на алгоритмическом языке Турбо - Паскаль

тема: «Графический режим монитора»

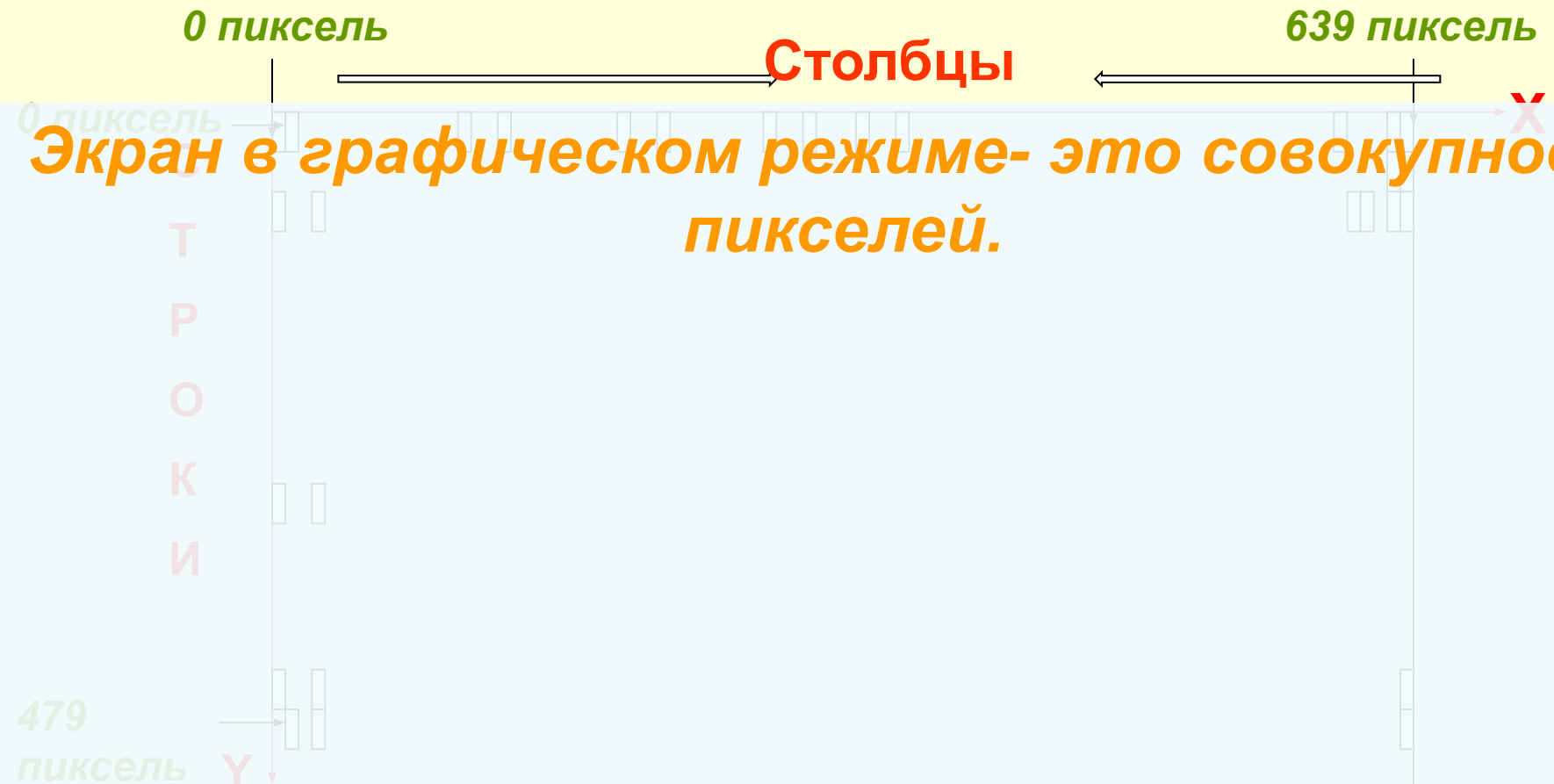


Методические разработки
учителя информатики
лицей №1581

Лапшиной О.М.

2007 г.

Экран монитора в графическом режиме.



Экран в графическом режиме- это совокупность пикселей.

СТАНДАРТНЫЙ МОДУЛЬ GRAPH

Все процедуры и функции для работы с графическим режимом монитора находятся в модуле **GRAPH**.

Поэтому после строки

Program N1;

необходимо набрать

USES GRAPH;

ИНИЦИАЦИЯ ГРАФИКИ

Предварительно в окне редактора TP в опциях **Options – Directories**,

В строке **Unit directories** прописать путь к стандартному модулю GRAPH:

D:\tp7\units

Сохранить настройку при помощи опции

Options - save

ИНИЦИАЦИЯ ГРАФИКИ

Процедура

INITGRAPH(D,M,P);

D – определяет тип графического драйвера(9 - VGA);

M – режим работы графического адаптера (максимальный - 2);

P - путь к графическому драйверу EGAVGA.BGI.

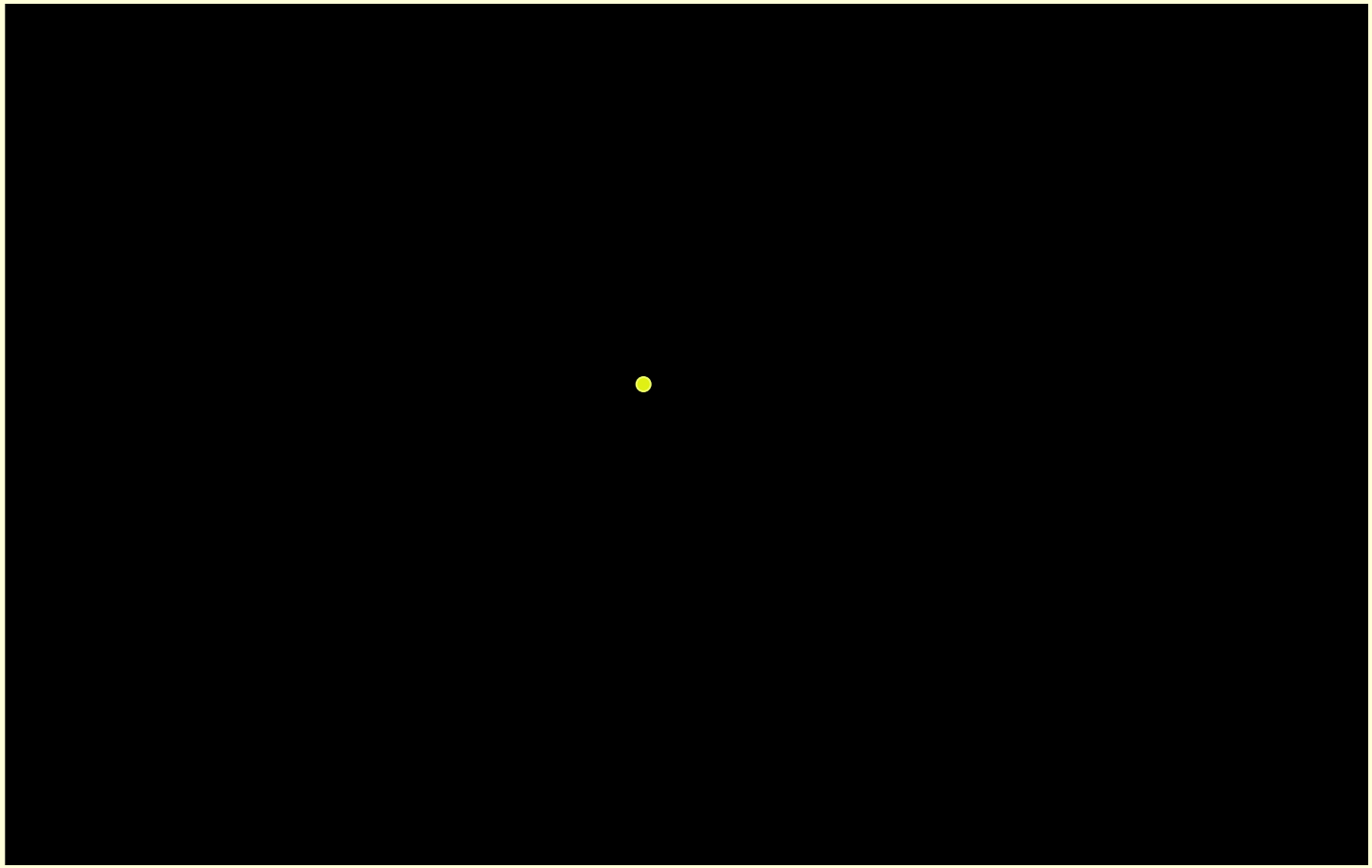
ИНИЦИАЦИЯ ГРАФИКИ

D,M - переменные типа INTEGER;

P - переменная типа STRING.

```
Var
d,m:integer;
p:string;
begin     d:=detect; m:=2;
INITGRAPH(D,M,'D:\TP7\BGI');
Putpixel(320,240,14);
ReadIn
End.
```

Вывод точки на экран



Процедуры изменения цвета

Назначение: *для изменения цвета объекта и/или фона.*

Общий вид:

1.SETCOLOR(C);

2.SETBKcolor(C);

Где: С-цвет объекта или цвет фона

Количество цветов =16 (0-15).

Работа: ***setcolor(c)*** *- изменяет цвет объекта.*

SETBKcolor(C) *- изменяет цвет фона.*

◄

Процедуры работы с отрезком

Стиль отрезка

SETLINESTYLE(T,P,Th);

T – стиль,

Th – толщина в пикселях.

Пример: ***program otp;***

Uses graph;

Var d,m:integer;

Begin d:=detect; m:=2;

Initgraph(d,m,'d:\tp\bgi');

setcolor(13); setlinestyle(2,0,3);

Line(10,10,630,10); readln end.

◄

Вывод отрезка на экран

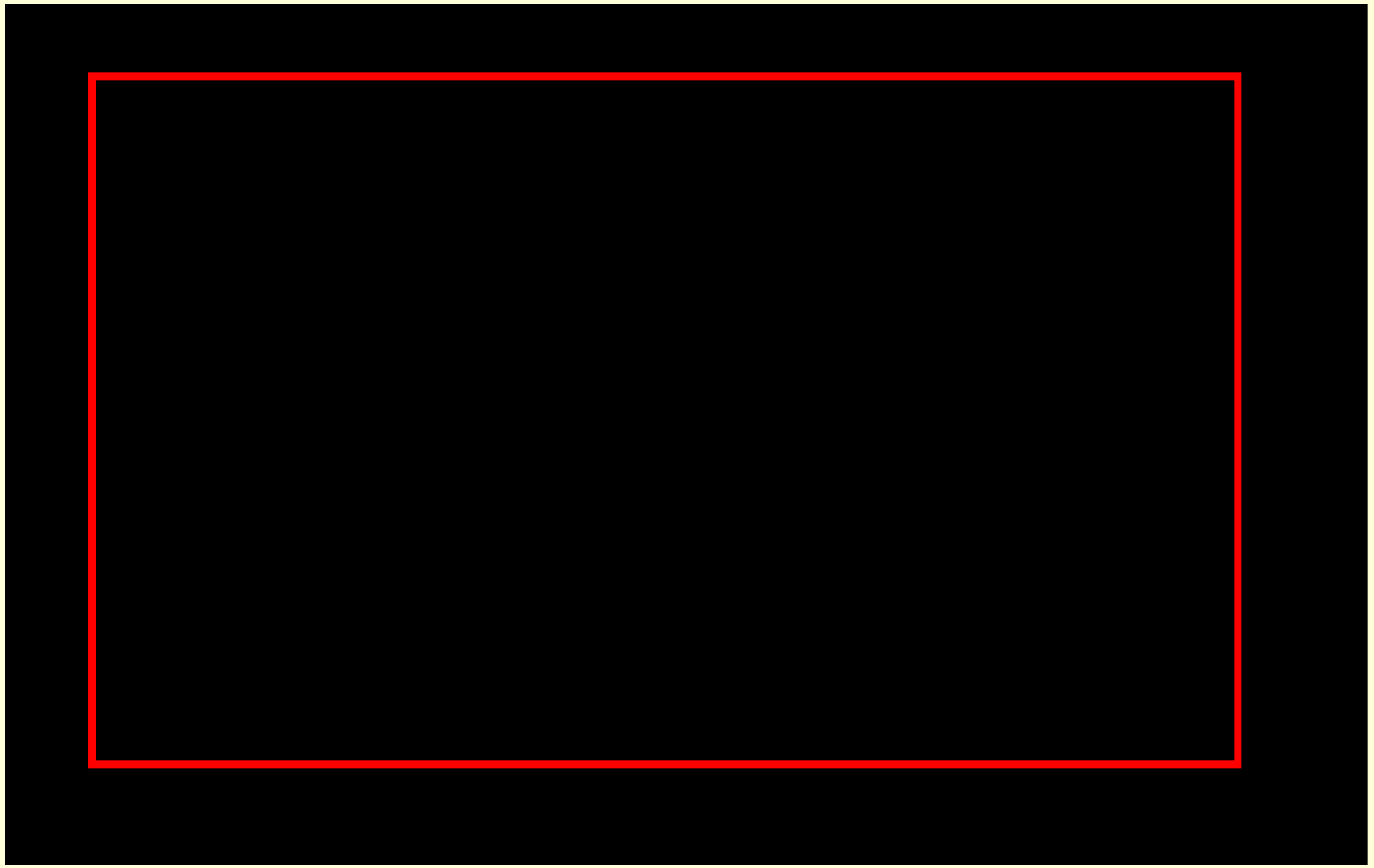
Процедура изображения прямоугольника

Rectangle(X1,Y1,X2,Y2);

Пример:

```
program pr;
Uses graph;
Var d,m:integer;
Begin d:=detect; m:=2;
Initgraph(d,m,'d:\tp\bgi');
setcolor(4); setlinestyle(1,0,3);
rectangle(10,10,630,470); readln end.
```

Вывод прямоугольника на экран



Процедуры заливки

setfillstyle(f,c);

F - стиль заливки (1 – 11); c – цвет.

Floodfill(x,y,b);

X,y – координаты любой точки внутри замкнутой области, b – цвет границы, до которой производится заливка.

Пример: *program pr;*

Uses graph;

Var d,m:integer;

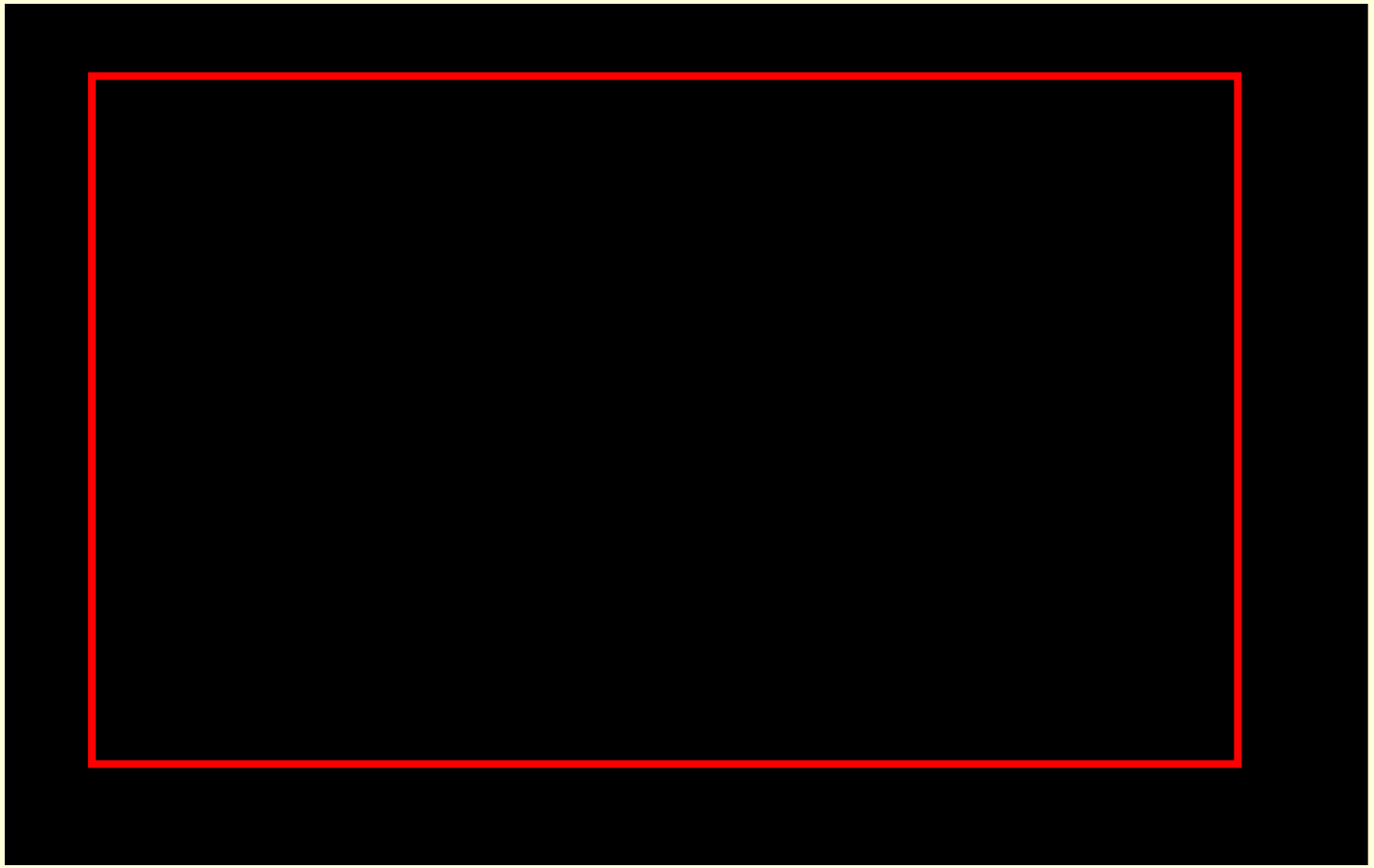
Begin d:=detect; m:=2;

Initgraph(d,m,'d:\tp\bgi');

setcolor(4); setfillstyle(4,13);

rectangle(10,10,630,470); Floodfill(20,20,4); readln end.

Вывод прямоугольника на экран



Процедура залитого прямоугольника

bar(x1,y1,x2,y2);

Пример: ***program pr;***

Uses graph;

Var d,m:integer;

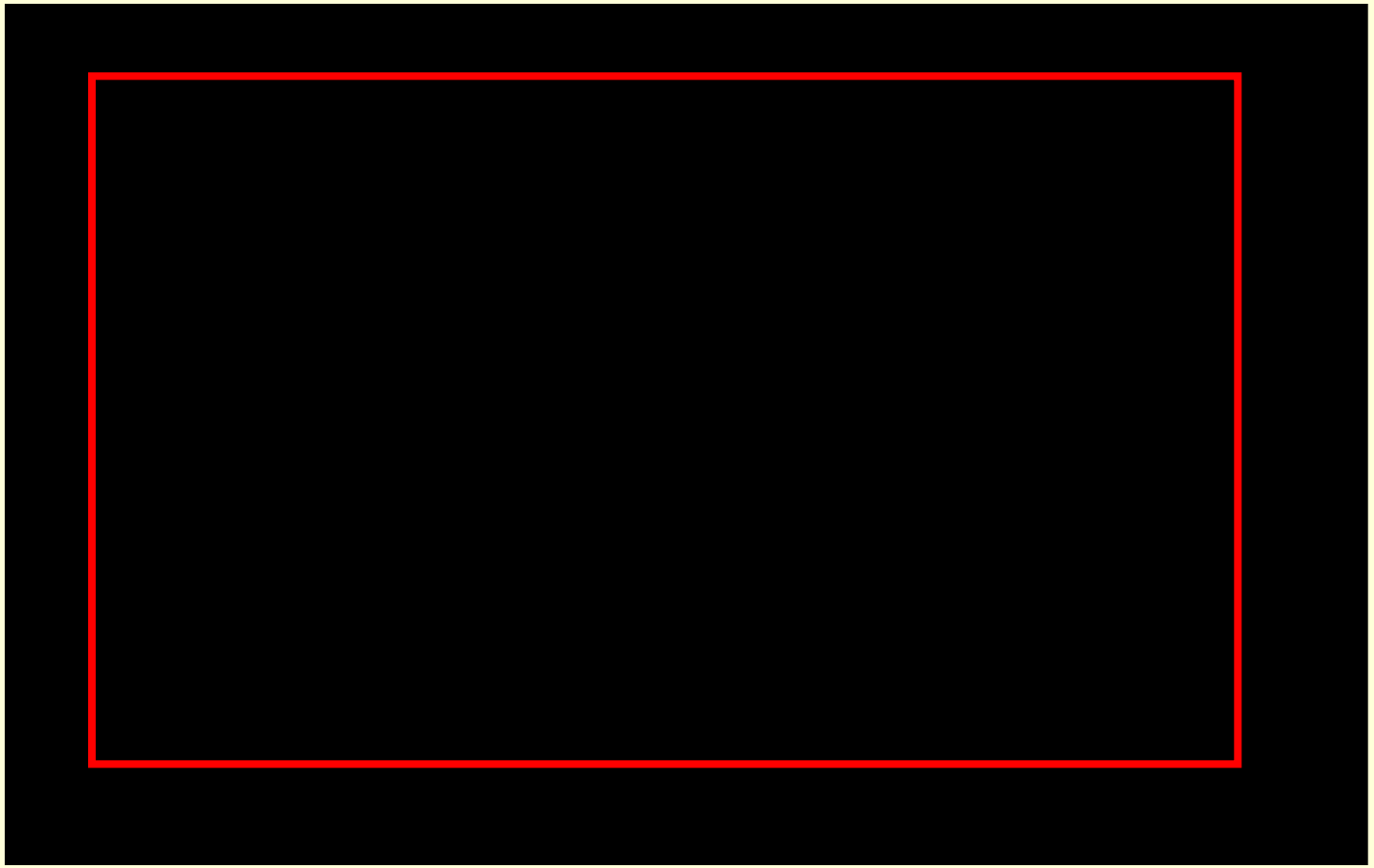
Begin d:=detect; m:=2;

Initgraph(d,m,'d:\tp\bgi');

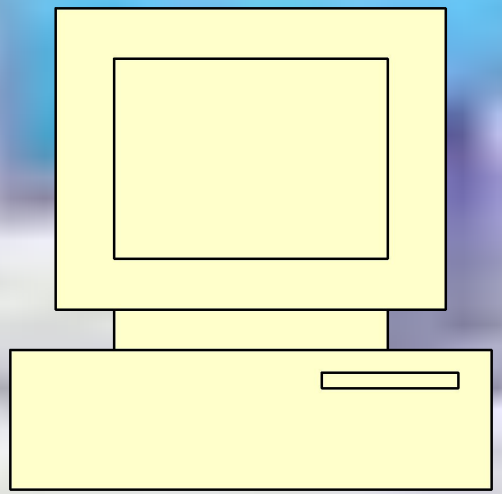
setcolor(4); setfillstyle(4,13);

bar(10,10,630,470); readln end.

Вывод прямоугольника на экран



Задания
на закрепление пройденного.



Процедура изображения прямоугольного параллелепипед

bar3d(x1,y1,x2,y2, d, T);

D - толщина в пикселях;

T – наличие или отсутствие верхне крышки (если T=true, то верхняя крышка есть, если T= false, то нет).

Процедуры изображения окружности, эллипса, дуг, секторов

Окружность
CIRCLE(X,Y,R);

Дуга окружности
ARC(X,Y,NU,CU,R);

Сектор окружности
PIESLICE(X,Y,NU,CU,R);

Эллипс и дуга эллипса
ELLIPSE(X,Y,NU,CU,RX,RY);

Сектор эллипса
SECTOR(X,Y,NU,CU,RX,RY);

R,RX,RY - Радиус в пикселях;

NU,CU - начальный угол и конечный угол в градусах против часовой стрелки;

X, Y - Координаты центра.

Процедуры вывода текстовой константы

1. *Settextstyle(f,d,s);*

   F – код шрифта (0 – точечный),

   D – направление(0 – горизонтально, 1 – снизу вверх),

   S – размер (1 – 10).

2. *outtext(Текстовая константа);*

3. *outtextxy(x,y,Текстовая константа);*

Процедура очистки экрана

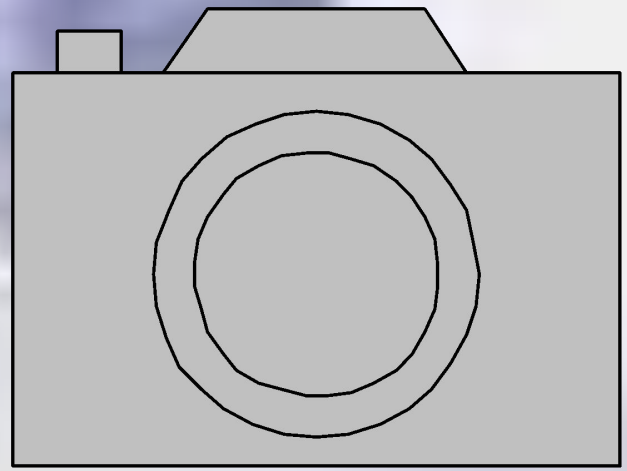
Назначение: ***предназначена для очистки экрана.***

Общий вид: ***CLeaRdevise;***

Работа:

- ***очищает экран;***
- ***устанавливает курсор в позицию (0,0);***

Задания
на закрепление пройденного.



Графический режим в Турбо Паскале

- ✔В графическом режиме экран представляет собой совокупность точек, каждая из которых может быть окрашена в один из 16 цветов;
- ✔ Координаты точек возрастают слева направо и сверху вниз; левая верхняя точка имеет координаты (0,0), а правая нижняя – (639,479);
- ✔Для того, чтобы программа могла выводить на экран графические примитивы (линии, окружности, прямоугольники), необходимо **инициализировать** графический режим.

Шаблон графической программы выглядит следующим образом:

```
Uses Graph;
Var  grDriver: Integer;
     grMode: Integer;
     ErrCode: Integer;
Begin
     grDriver:=detect;
     InitGraph(grDriver, grMode, 'c:\BP\bgi');
     ErrCode:=GraphResult;
```

```
If  ErrCode=grOk then
  begin
     {сюда следует поместить основные инструкции}
  end;
     readln;
      CloseGraph;
End.
```

Процедуры и функции графического режима

PutPixel(x,y,color); - окрашивает пиксел, точку с координатами (x,y), цветом color. В качестве параметра color обычно используют именованную константу. {x,y: integer}

Line(x1,y1,x2,y2); - вычерчивает линию между двумя точками экрана, координаты которых указаны при вызове процедуры. Линия вычерчивается стилем, установленным процедурой **SetLineStyle**. Цвет линии можно задать при помощи процедуры **SetColor**.

Rectangle(x1,y1,x2,y2); - вычерчивает прямоугольник. Параметры x1 и y1 задают положение верхнего угла прямоугольника, x2 и y2 – правого нижнего угла прямоугольника. Вид контура прямоугольника можно задать при помощи процедуры **SetLineStyle,** а цвет – при помощи процедуры **SetColor**.

SetLineStyle(ТипЛинии, Образец, Толщина); - устанавливает стиль вычерчиваемых контуров и линий.

Параметр **ТипЛинии**, в качестве которого обычно используется одна из именованных констант определяет вид линии.

Параметр **Толщина** опредяляет толщину линии. Линия может быть обычной толщины (константа NormWidth) или утолщённая (константа ThickWidth).

Параметр **Образец** используется в том случае, если процедура **SetLineStyle** устанавливает тип линии, определяемый программистом. Значением параметра **Образец** должна быть четырёхразрядная шестнадцатиричная константа, кодирующая отрезок линии длиной в 16 пикселов.

Константа	**Тип линии**
SolidLn	Сплошная, непрерывная
DottedLn	Пунктирная, с постоянной длиной штрихов

CenterLn	Штрих-пунктирная линия
DashedLn	Пунктирная, длина штрихов чуть больше, чем у линии типа DottedLn
UserBitLn	Определённый программистом тип линии

SetColor(Цвет) – задает цвет для вывода текста(процедуры OutTextXY и OutText), вычерчивания линий и фигур (процедуры Line, Circle, Rectangle и др.). В качестве параметра Цвет обычно используют именованные константы.

Circle(x,y,r) – вычерчивает окружность радиуса r с центром в точке с координатами (x,y). Цвет окружности можно задать при помощи процедуры **SetColor.**

Ellipse(x, y, УголНачала, УголКонца, РадиусX, РадиусY); - вычерчивает эллипс или дугу эллипса с центром в точке с координатами (x,y). Параметры **УголНачала** и **УголКонца** задают угловые координаты начальной и конечной точек линии эллипса, которая вычерчивается против часовой стрелки от начальной точки к конечной. Угловые координаты задаются в градусах, их значения возрастают против часовой стрелки. Параметры **РадиусX** и **РадиусY** определяют горизонтальный и вертикальный радиусы эллипса. Линия эллипса или дуги вычерчиваются в соответствии с установками процедуры **SetColor.**

Sector(x, y, Угол1, Угол2, РадиусX, РадиусY); - вычерчивает эллиптический (РадиусX≠ РадиусY) или круговой (РадиусX=РадиусY) сектор. Параметры x, y задают положение центра сектора, Угол1 и Угол2 – углы прямых, ограничивающих сектор, а РадиусX и РадиусY – радиусы эллипса по осям X и Y, из которого «вырезается» отображаемый сектор. Нулевому углу соответствует горизонтальный отрезок, проведённый из точки (x,y) в сторону возрастания координаты x.

PieSlice(x,y,УголНачала, УголКонца, Радиус); - вычерчивает сектор радиуса **Радиус** с центром в точке с координатами **(x,y)**. Параметры **УголНачала** и **УголКонца** задают угловые координаты начальной и конечной точек линии окружности (в градусах), которая вычерчивается против часовой стрелки от начальной к конечной точке. Значение угловой координаты возрастает против часовой стрелки. Нулевому углу соответствует горизонтальный отрезок, проведенный из точки **(x,y)** в сторону возрастания координаты **x**.Если **УголНачала** равен 0^o,

а **УголКонца** равен 360º, процедура вычерчивает круг.

Сектор закрашивается в соответствии со стилем, заданным процедурой **SetFillStyle**, линия и границы вычерчиваются цветом, установленным процедурой **SetColor**.

Bar(x1,y1,x2,y2) – вычерчивает закрашенный прямоугольник. Параметры **x1** и **y1** задают положение левого верхнего угла прямоугольника, **x2** и **y2** - правого нижнего. Используемый стиль и цвет заливки задается процедурой **SetFillStyle**.

Bar3D(x1,y1,x2,y2, Глубина, Граница) – вычерчивает параллелепипед. Параметры **x1** и **y1** задают положение левого верхнего, а **x2** и **y2** - правого нижнего угла ближней грани параллелепипеда. Параметр **Глубина** задаёт расстояние между передней и задней гранями. Параметр **Граница** определяет, нужно ли вычерчивать верхнюю границу задней грани параллелепипеда.

Цвет и стиль закраски ближней грани параллелепипеда можно задать при помощи процедуры **SetFillStyle** , цвет линий границы – процедурой **SetColor** .

SetFillStyle(Стиль, Цвет) – устанавливает стиль и цвет заливки (закрашивания). В качестве параметра **Стиль** обычно используют одну из именованных констант, список которых приведён ниже. Параметр **Цвет** также задаётся именованной константой.

Константа	**Стиль заполнения области**
EmptyFill	Без заливки (сплошная заливка цветом фона)
SolidFill	Сплошная заливка текущим цветом
LineFill	Горизонтальная штриховка
LtSlashFill	Штриховка под углом 45° влево тонкими линиями

SlashFill Штриховка под углом 45° влево

BkSlashFill Штриховка под углом 45° вправо тонкими линиями

LtBkSlashFill Штриховка под углом 45° вправо

HatchFill Штриховка клеткой

XhatchFill Штриховка под углом 45° редкой косой клеткой

InterleaveFill Штриховка под углом 45° частой косой клеткой

WideDotFill Заполнение редкими точками

CloseDotFill Заполнение частыми точками

UserFill Тип заполнения определяется программистом