ГРАФИЧЕСКИЕ ОПЕРАЦИИ И ИХ ОПИСАНИЕ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ

Цикл уроков

Л. Г. Якуша

УРОК № 1

Тема. Графические операции и их описание на языке программирования

Цель: ознакомить учащихся с понятием машинной графики; рассмотреть графические системы; ознакомить с понятиями «система координат» и «текущая точка», графическими возможностями систем программирования; выяснить, как формируется RGB-цвет пикселей, как происходит инициализация графического режима; развивать у учащихся умение анализировать, сравнивать и делать выводы.

Тип урока: усвоение новых знаний.

Ход урока

I. Организационный этап

II. Мотивация учебной деятельности

— Достижения современных мультимедийных технологий активно проявляются в компьютерных играх, в рекламе на многочисленных интернет-сайтах, в научно-фантастических и приключенческих фильмах, в телевизионных роликах и т. д. И наиболее важным компонентом в этом разнообразии является компьютерная графика.

— Как вы считаете, с помощью каких программ или систем программирования создают данные мультимедийные технологии?

— Назовите, пожалуйста, эти программы.

III. Изложение нового материала

Графическая система

Монитор в сочетании с управляющей схемой, оформленной в виде отдельной платы (видеокарты) или специализированной микросхемы на материнской плате, образуют так называемую графическую систему.

Основные характеристики графической системы: размер монитора, разрешение экрана, цветовая гамма.

В графическом режиме экран напоминает миллиметровку, то есть он разбит на много мелких клеточек, которые называют пикселями (pixel — pictere's element, «элемент рисунка»). Наивысший графический режим, поддерживаемый системой программирования Pascal, соответствует стандарту VGA — 480 точек по вертикали, 640 точек по горизонтали, 16 цветов.

Система координат и текущая точка

В графических режимах начало экранных координат расположено в левом углу экрана и первый пиксель имеет координату (0; 0). Ось *X* направлена вправо, а ось *Y* — вниз (см. *рис. 1*).

Рис. 1. Система координат в графическом режиме

В графическом режиме текущей точкой иногда называют графический курсор, который не отображается на экране, чтобы не искажать выведенную на экран картинку. Но программа в любой момент может определить координаты текущей точки.

Формирование RGB-цвета пикселей

RGB — это сокращение от английских названий базовых цветов — red (красный), green (зелёный), blue (голубой). Именно эти цвета, смешанные в определённой пропорции, и руководят закрашиванием пикселей на экране. В стандарте VGA используется 4 бита, которые запоминают код цвета. По ним определяется один из 16 регистров палитры, которая участвует в формировании цвета. Это даёт возможность манипулировать

64 цветами, но выбирать из них в каждый конкретный момент можно лишь 16.

Каждый раз графическая система имеет дело с двумя ранее установленными цветами — цветом переднего плана (цвет рисования — foreground color) и цветом заднего плана (цвет фона — background color).

Графические возможности систем программирования

Каждая система программирования предусматривает базовые графические средства, с помощью которых можно установить нужный режим работы видеосистемы, управлять палитрой цветов, рисовать примитивные геометрические фигуры и раскрашивать их, подписывать рисунки поясняющими надписями.

Графические средства систем Turbo Pascal построены на базе общего подхода, который сокращённо называется *BGI – Borland Graphics Interface* (графический интерфейс фирмы Borland). Эти средства внесены в системную библиотеку, которая содержит набор процедур и функций, совпадающих с именами и аналогичным набором аргументов. Такие библиотеки содержат 83 графических программы и 60 системных констант.

В отличие от других систем программирования, например C и QBasic, в обозначениях процедур и констант Паскаль выделяет начало каждого ключевого слова большой буквой, например SetUserCharSize.

Функции с параметрами в Паскале записывают без использования пустых скобок в именах функций.

Например:

```
TC: x=getmaxx( );
TP: x:=GetMaxX;
```

В программе построения графических изображений желательно сразу после её заголовка записать оператор подключения библиотеки графических процедур **Graf** и других модулей, если их нужно использовать при выполнении программы.

Оператор имеет такую структуру:

```
uses Graf, <другие модули>
```

Взаимодействие программы и видеосистемы в графическом режиме обеспечивают драйверы, которые имеют расширение .bgi. Каждый драйвер обеспечивает работу в нескольких графических режимах. Название драйвера и номер режима должны быть определены в процессе инициализации графического режима работы компьютера.

Инициализация графического режима

Переход компьютера в определённый графический режим в системе BGI осуществляется с помощью процедуры:

```
InitGraph(gd, gm', pach');
InitGraph(gd, gm' ');
```

Рассмотрим значение переменных в скобках:

1) *gd* — целочисленная переменная, запоминающая условный номер графического драйвера;
2) *gm* — целочисленная переменная, указывающая на номер графического режима;
3) *pach* — задаёт путь к каталогу, содержащему драйвер видеосистемы, которая реализует связь библиотечных процедур с видеокартой. Путь можно также задать через пустую строку ' '. Но если графическая система не найдёт нужного драйвера, работа программы будет завершена из-за того, что «открытие» графики не состоялось.

Закрытие графического режима осуществляется оператором **CloseGraph**. Чтобы установить цвет фона экрана, используют процедуру **SetBkColor** (номер цвета). Если данная процедура не указана, экран будет чёрным.

Таким образом, можно привести следующую упрощённую общую структуру программы построения и отображения графических объектов.

```
Program prukl; {Заголовок программы}
uses Graph; {Подключение библиотеки
  графических процедур}
var
  GraphDriver: Integer; {Переменные
  инициализации определённого графического
  режима}
  GraphMode: Integer;
begin
  GraphDriver:=0; {Автоматический выбор
  драйвера}
  InitGraph(gd, gm, ' '); {Инициализация
  графического режима}
  <операторы>{Построение и отображение
  графических объектов}
  <графические процедуры и функции>
  readln; {Ожидание нажатия клавиши}
  CloseGraph;{Закрытие графического режима}
end.
```

Итак, начиная работу в графическом режиме, следует помнить:

1) в графическом режиме экран — это совокупность точек (пикселей), каждая из которых может быть закрашена в один из 16 цветов;
2) координаты точек растут слева направо и сверху вниз; левая верхняя точка имеет координаты (0, 0), а права нижняя — (632, 479);

3) для того чтобы программа могла выводить на экран графические изображения, необходимо инициализировать графический режим.

IV. Закрепление нового материала

1. Что такое графическая система? Основные её характеристики.
2. Как записывают оператор подключения к библиотеке графических процедур?
3. Какой вид имеет система координат монитора в графическом режиме? Дать определение текущей точки.
4. С помощью какой процедуры осуществляют инициализацию графического режима?
5. Описать значение переменных, которые используют в процедуре для инициализации графического режима.

V. Домашнее задание

УРОК № 2
Тема. Основные группы процедур и функций графики. Применение в программах процедур и функций для построения простых графических изображений

Цель: ознакомить с основными процедурами и функциями для построения простых графических изображений; закрепить умение составлять и реализовывать программы на построение простых графических изображений с использованием графических функций и процедур, используя язык программирования.

Тип урока: комбинированный урок.

Ход урока

I. Организационный этап

II. Актуализация опорных знаний и проверка домашнего задания

➔ **Фронтальный опрос**

1. Что называют графической системой?
2. Какое отличие между системой координат в графическом режиме и системой координат, которой мы пользуемся в математике?
3. Что такое RGB-цвет пикселей? Как расшифровывается аббревиатура?
4. Какую процедуру используют для инициализации графического режима?

III. Сообщение темы и цели урока

На предыдущем уроке мы с вами рассмотрели графические системы и их основные составляющие. Выяснили, каким образом формируется изображение на экране, а также узнали о виде системы координат в графическом режиме и установили понятие текущей точки.

Также выяснили, что для работы в графическом режиме необходимо его инициализировать.

На этом уроке вы выучите основные группы процедур и функций, которые необходимы для работы в графическом режиме, а также научитесь составлять программы с использованием этих функций и процедур для создания простых графических изображений.

IV. Изложение нового материала

Итак, рассмотрим основные группы процедур и функций, которые используют для работы в графическом режиме.

Описание процедуры или функции	Производимое действие
`PutPixel(x, y, n)`	Осуществляет построение точки, где x, y — координаты точки, n — номер цвета. Координаты точки должны быть целыми числами. Если в качестве координаты используют действительные выражения, то с помощью функции **trunc** координату округляется до ближайшего целого числа
`Line(x1, x2, y1, y2)`	Осуществляет построение отрезка текущим цветом, отрезок, где x_1, y_1 — координаты начала, x_2, y_2 — координаты конца отрезка
`LineTo(x, y)`	Проводит линию от текущей точки к координатам (x,y)
`LineRel(dx, dy)`	Проводит линию от текущей точки с приращением dx, dy
`MoveTo(x, y)`	Изменяет значение текущих координат
`Rectangle(x1, y1, x2, y2)`	Выводит прямоугольник с координатами левого верхнего и правого нижнего углов

Описание процедуры или функции	Производимое действие
Bar(x1, y1, x2, y2)	Выводит прямоугольник, закрашенный текущим стилем
Circle(x, y, r)	Осуществляет построение круга радиуса *r* с центром в точке *x*,*y*
Arc(x, y, a, b, r)	Осуществляет построение дуги радиуса *r* с центром в точке *x*, *y*;*a*— значение начального угла, *b* — значение конечного угла
OutTextXY(x, y, 'текст')	Вводит текст, начиная с координаты *x*,*y*
Процедуры для установки цвета и штриховки	
SetColor(цвет)	Устанавливает цвет для вывода символов и линий
SetBKColor(цвет)	Устанавливает цвет фона
SetFillStyle(код, цвет)	Устанавливает вид штриховки в соответствии с кодом, который может приобретать несколько значений
SolidFill=1	Сплошная штриховка
LineFill=2	Штриховка линиями
HatcheFill=7	Штриховка символом «+»
FloodFill(x, в, цвет предела)	Штрихует любую замкнутую область, *x*, *y* — координаты точки внутри области
CloseDotFill=11	Густая штриховка точками
ClearDevise	Очищает весь экран

Цвета, которые используются в графическом режиме

Цвет	Номер цвета	Обозначение
black	0	чёрный
blue	1	синий
green	2	зелёный
gyan	3	голубой
red	4	красный
magenta	5	фиолетовый
brown	6	коричневый
lightgray	7	светло-серый
darkgray	8	тёмно-серый
lightblue	9	ярко-синий
lightgreen	10	ярко-зелёный
lightgyan	11	ярко-голубой
lightred	12	розовый
lightmagtnta	13	малиновый
yellow	14	жёлтый
white	15	белый

➔ Работа учащихся с таблицами

Учитель называет процедуру или функцию графического режима, а ученик должен найти её запись на языке программирования.)

V. Составление программ, которые используют процедуры и функции для создания простых графических изображений

Задача 1. Составить программу для построения квадрата, вписанного в круг.

```
Program Kvadrat;
uses graph; {Подключение библиотеки
  графических процедур}
var
  gd, gm: integer; {Переменные установки
  графического режима}
  X1, y1, r: integer; {Переменные координат
  квадрата, радиус круга}
begin
  writeln ('Введите начальные координаты
  квадрата');
  readln(x1, y1); {Начальные координаты
  квадрата}
  Gd:=0; {Автоматический выбор драйвера}
  InitGrahp(gd, gm' '); {Инициализация
  графического режима}
  SetBKColor(15); {Белый цвет фона}
  SetColor(1); {Синий цвет изображения}
  Rectangle(x1, y1, x1+200, y1+200);
  {Построение квадрата}
  R:=trunc(sqrt(100*100+100*100));
  {Вычисление радиуса круга}
  Circle(x1+100, y1+100, r); {Построение
  круга}
  OutTextXY(190,350, 'рис.1'); {Вывод
  текста}
  readln; {Ожидание нажатия клавиши}
  CloseGrahp; {Закрытие графического режима}
end.
```

Если ввести с клавиатуры координаты x_1=100 и y_1=100, то в результате выполнения программы на белом фоне экрана синим цветом будет построен квадрат со стороной 200, вписанный в круг с координатами центра *x*=200, *y*=200. Под рисунком

будет надпись — рис. 1. Для возвращения в программу нужно нажать любую клавишу.

Задача 2. Центры кругов имеют одни и те же координаты. Разработать программу построения 5 кругов разных цветов, если известны координаты центра и радиус наименьшего круга. Радиус следующего увеличивается на 20 пикселей.

Решение

Переменные x и y — координаты кругов, переменная — сначала радиус наименьшего круга, потом радиус текущего круга. Координаты и радиус круга равны 1, а значение цвета каждого следующего круга увеличивается на 1.

```
Program Krug;{}
uses Graph; {Подключение библиотеки
  графических процедур}
var
  gd, gm: Integer; {Переменные установки
  графического режима}
  x, y, r: Integer; {Координаты и радиус
  наименьшего круга}
begin
  writeln ('Введите координаты и радиус
  круга');
  readln(x, y, r);
  Gd:=0; {Автоматический выбор драйвера}
  InitGrahp(gd, gm' '); {Инициализация
  графического режима}
  SetBKColor(15); {Белый цвет фона}
  SetColor(7); {Цвет точки центра круга}
  PutPixel(x, y, 1); {Точка центра круга}
  for i=1 to 5 do begin
  SetColor(i); {Цвет круга}
  Circle(x, y, 1){Построение круга}
  r:=r+20 end; {Увеличение радиуса}
  readln; {Ожидание нажатия клавиши}
  CloseGrahp; {Закрытие графического режима}
end.
```

Задача 3. Составить программу, которая строит квадрат, делит его на 4 равных квадрата и в каждый вписывает круг.

Решение

В программе сначала построить квадрат с координатами (50, 50, 450, 450). Потом через центр провести вертикальную и горизонтальную прямые. В небольшие квадраты вписать круги.

```
Program Kvadrat;
uses Graph; {Подключение библиотеки
  графических процедур}
var gd, gm: Integer; {Переменные установки
  графического режима}
begin
  Gd:=0; {Автоматический выбор драйвера}
  InitGrahp(gd, gm' '); {Инициализация
  графического режима}
  SetBKColor(15); {Белый цвет фона}
  SetColor(1); {Синий цвет изображения}
  Rectangle(50,50,450,450); {Построение
  квадрата}
  Line (50,250,450,250){Горизонтальная
  линия}
  Line(250,50,250,450){Вертикальная линия}
  Circle(150,150,100); {Построение круга}
  Circle(150,350,100);
  Circle(350,150,100);
  Circle(350,350,100);
  OutTextXY(470,300, 'рис.2'); {Вывод
  текста}
  readln; {Ожидание нажатия клавиши}
  CloseGrahp; {Закрытие графического режима}
end.
```

VI. Закрепление нового материала

➔ Тесты (работа в парах)

1. Какой из указателей инициализации графического режима записан правильно?
   А. InitGraph('gd', gm, « «).
   Б. InitGraph(gd, «gm», « «).
   В. InitGraph(gd, gm, ' ').
   Г. InitGraph('gd', 'gm « «).
2. Указать соответствие. Какая из нижеперечисленных графических процедур:

1. рисует на экране точку;	А. SetBKColor;
2. рисует на экране отрезок прямой;	Б. Arc;
3. рисует на экране круг;	В. PutPixel;
4. рисует на экране дугу;	Г. Circle S;
5. устанавливает цвет фона на экране;	Д. Line?

3. Для работы в графическом режиме необходимо подключить модуль:
   А. Dos.
   Б. Ctr.
   В. Graph.
   Г. Windows.
4. Цвет изображения точек и линий можно установить с помощью процедуры:
   А. SetColor (цвет).
   Б. CloseDotFill=11.
   В. SetBKColor (цвет).
   Г. ClearDevise.
5. Вид штриховки можно установить с помощью процедуры:
   А. SetFillStyle (код, цвет).
   Б. CloseDotFill=11.
   В. SetBKColor (цвет).
   Г. ClearDevise.
6. Установить соответствие между процедурами и результатами.

1. ClearDevise	А. Вывод текста.
2. CloseGrahp	Б. Очистка экрана.
3. OutTextXY	В. Закрытие графического режима.

VII. Домашнее задание

1. Подготовиться к практической работе.
2. Выучить материалы конспекта.

УРОК № 3

Тема. Создание простых графических изображений с использованием процедур и функций графического режима

Цель: научить переходить в графический режим работы и создавать простые графические изображения.

Тип урока: проверка знаний, умений и навыков.

Ход урока

I. Задания для практической работы

1. Знать теоретический материал по теме:
   а) переход к графическому режиму;
   б) вид координатной сетки экрана монитора;
   в) запись процедур и функций изображения линий и фигур;
   г) запись процедур и функций установки цвета фона и штриховки фигур.
2. Составить программу создания простых графических изображений.
3. Выполнить программу и вывести изображение на экран.
4. Оформить отчёт о выполнении практической работы (листинг программы) и ответить на контрольные вопросы.

Задания

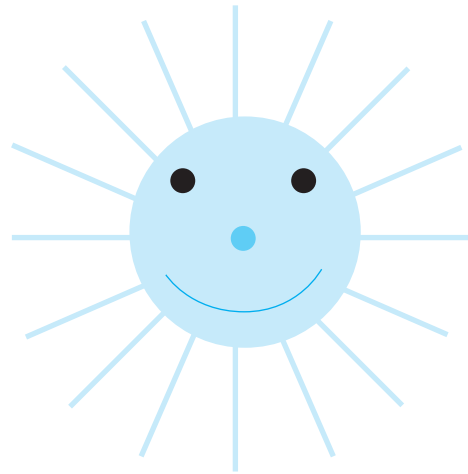
Уровень А

1. Построить треугольник с вершинами в точках (100, 100), (150, 100), (80, 70). Цвет фона — серый, цвет линий — красный.
2. Построить прямоугольник с вершинами (80, 80), (170, 180), (170, 150), (80, 150). Закрасить его жёлтым цветом. Цвет фона — малиновый.

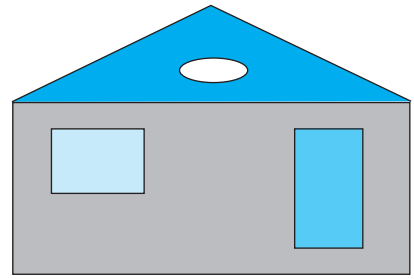
Уровень Б

1. Нарисовать желтое солнышко, у которого черные глазки, оранжевый нос, улыбка красного цвета и желтые лучики.
2. Нарисовать дом, в котором зелёная крыша, жёлтое окно, малиновые двери.

Например, вид солнышка к задаче 1 (уровень Б)



Вид дома к задаче 2 (уровень Б):



Контрольные вопросы

1. Как осуществляется инициализация графического режима?
2. Как выглядит экранная система координат монитора в графическом режиме?
3. Что должен помнить пользователь перед началом работы в графическом режиме?
4. С помощью каких функций и процедур можно изобразить в графическом режиме точку, отрезок прямой, прямоугольник, круг?
5. С помощью какого указателя можно определить цвет точек и линий?
6. С помощью какого указателя можно изменять цвет фона?
7. С помощью какого указателя можно устанавливать вид штриховки?
8. Как осуществляется очистка экрана?
9. С помощью какого указателя осуществляется закрытие графического режима?

II. Анализ и оценивание работы учащихся на уроке

III. Домашнее задание

Творческое задание: составить программу для построения простых графических изображений с использованием основных процедур и функций графического режима.