**Графический режим в Turbo Pascal. Метод координат. Построение линий**

**Тип урока:** урок с применением опорных конспектов

**Вид урока:** комбинированный урок

**Технология:** личностно-ориентрованная

**Цели урока:**

* *Обучающая*: создать условия для формирования первичного представления о методе координат в графике Turbo Pascal.
* *Развивающая:* Развивать логическое и алгоритмическое мышление через установление причинно-следственных связей.
* *Мотивационная:* побудить интерес к изучению предмета.

**Задачи урока:**

* *Воспитательная:* развитие познавательного интереса, логического мышления.
* *Учебная:* Познакомиться с методом координат в графическом режиме Turbo Pascal и научиться строить графические рисунки из линий и точек.
* *Развивающая:* Развитие алгоритмического мышления, памяти, внимательности.

**Подготовка к уроку:**

1. Каждому ученику выдается лист опорного конспекта.
2. Разработаны разноуровневые задания по теме урока.

**План урока:**

1. Организационный момент. Мотивационное начало урока.

2. Объяснение темы урока.

3. Этап обобщения, систематизации знаний и закрепления изученного.

4. Подведение итогов. Домашнее задание.

**Ход урока:**

1. Здравствуйте, ребята. Сегодня мы продолжаем с вами говорить о среде программирования Turbo Pascal. Тема нашего сегодняшнего урока “Графический режим в Turbo Pascal. Метод координат. Построение линий”. Цель нашего сегодняшнего урока – что характеризует графический режим в языке программирования Turbo Pascal, а также научиться составлять программы, которые будут выводить на экран монитора графическое изображение, состоящее из линий.

Для достижения целей урока мы сегодня будем использовать опорные конспекты, которые лежат у вас на столах. (Приложение 1). Положите их перед собой и внимательно посмотрите на них. Дело в том, что в этих конспектах вы должны будете в процессе урока вносить данные самостоятельно, затем этот конспект вы вклеите в свою тетрадь по информатике.

Перед тем, как начать говорить о графическом режиме в Turbo Pascal, давайте вспомним последнюю тему, которую мы рассматривали с вами.

Посмотрите на доску – там находится запись программы на языке Turbo Pascal. Задача написана с ошибками, всего их 11. Давайте найдем эти ошибки, и вы скажете, что же это за программа.

|  |  |
| --- | --- |
| **Неправильный вариант программы** | **Правильный вариант программы** |
| Programma RI;  Vara x,y:realn;  Writeln (x,y);  Readln(x,y);  Begin  z = sqrt(sgr(x)+sqr(y))  end;  Writeln (‘x,y’:4:3);  Readln; | Program RI;  Var z, x, y : real;  Begin  Writeln (‘x,y’);  Readln(x,y);  z : = sqrt(sgr(x)+sqr(y));  Writeln (‘z = ’,z:4:3);  Readln;  End. |

Учащиеся находят ошибки, анализируя программу, выясняем, данная программа вычисляет z по уравнению z = x1/2+y2.

2. Теперь мы переходим к графическому режиму в Turbo Pascal. Экран дисплея компьютера – это поле, состоящее из большого количества точек. Чтобы сделать процесс графического программирования более эффективным, была разработана специализированная библиотека Graph, которая представляет из себя набор драйверов.

Для построения изображения на экране используется система координат. Отсчет она начинает от верхнего левого угла экрана, который имеет координаты (0, 0). Рис.1

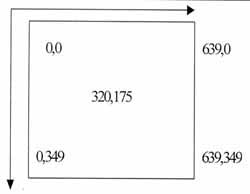


Рис.1

Ученики записывают координаты крайних точек в опорный конспект.

При работе с графическим модулем используются три основные момента:

1. Подключение модуля GRAPH - Uses Graph.

2. Установить графический режим с помощью двух переменных:

* GrDriver – устанавливает драйвер;
* GrMode – устанавливает режим работы драйвера.

3. Закрытие видео режима CloseGraph.

Какие бы изображения ни выводились на экран, все они построены из точек. Имея средство построения точки определенного цвета в нужном месте экрана, теоретически можно создать любое изображение, вплоть до картины. В библиотеке GRAPH ввод точки осуществляется процедурой:

PutPixel (x,y, color);

Где x,y – координаты точки

Color – цвет точки

Для того, чтобы построить линию используется процедура:

Line (x1,y1,x2,y2);

Где x1,y1 - координаты начала линии

x2,y2 - координаты конца линии

Для установки цвета линии применяется процедура:

SetColor (N);

Где N – номер цвета

Таким образом, для записи в программе, которая будет выводить на экран линию необходимо поместить:

SetColor (N);

Line (x1,y1,x2,y2);

Например:

SetColor (6);

Line (120,250,300,50);

Turbo Pascal позволяет вычерчивать линии различного стиля: тонкие, штриховые, пунктирные и т.д. Установка стиля производится следующей процедурой:

SetLineStyle (U, N);

Где U – название узора

N – цвет узора

Если с помощью линий создана замкнутая область, то в модуле Graph существует возможность заштриховать эту область. Для этого используется процедура:

FloodFill (x,y,N);

Где x,y - координаты точки, внутри замкнутой поверхности

N – номер цвета линии, которой обрамляется замкнутая поверхность.

Рассмотрим пример программы, которая выводит изображение линий на мониторе.

Program Liniya;

Uses Graph;

Var GrDriver, grMode: integer;

Begin;

GrDriver := Detect;

InitGraph (GrDriver, grMode, ‘ ‘);

SetColor (5);

Line (50,100,600,300);

Readln;

Close Graph;

End.

3. Для закрепления полученных знаний учащиеся получают карточки-задания, по которым нужно написать программу, которая выводит на экран заданный рисунок. Рис. 2, Рис.3, Рис. 4. Используя Приложение 2, закрасьте элементы рисунка.

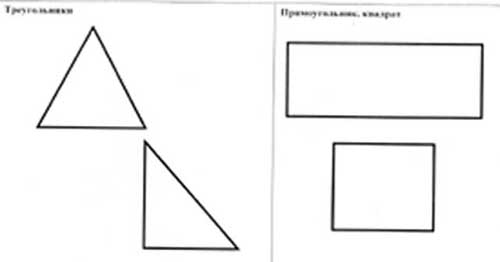


Рис. 2

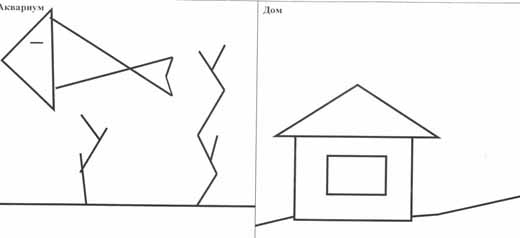


Рис.3

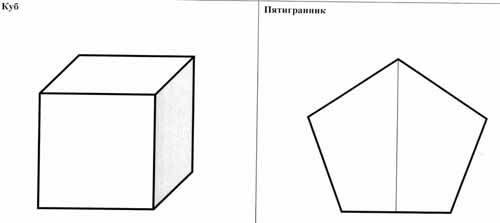


Рис.3

4. Работы ребят проверяются, оцениваются.

Ученики получают на дом задание – на клетчатом листе бумаги, построить рисунок из линий (не менее 12), записать координаты точек рисунка.

**Приложение1.**

Лист А4 занимают Рис.5, Рис.6, Рис.1, Рис. 7.

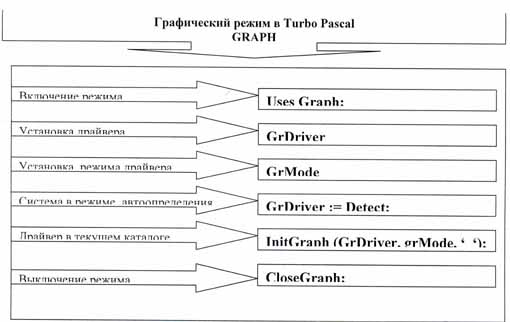
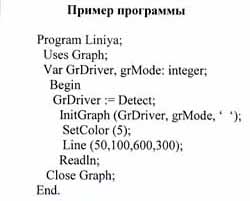


Рис.5



Рис.6



**Приложение 2.**

|  |
| --- |
| Константы цвета  Цвет (русский)  Цвет (английский)  Черный  Black  0  Синий  Blue  1  Зеленый  Green  2  Голубой  Cyan  3  Красный  Red  4  Фиолетовый  Magenta  5  Коричневый  Brown  6  Светло серый  LightGrey  7  Темно серый  DarkGrey  8  Ярко синий  LightBlue  9  Ярко зеленый  LightGreen  10  Ярко голубой  LightCyan  11  Розовый  LightRed  12  Малиновый  LightMagenta  13  Желтый  Yellow  14  Белый  White  15  **Константы типов и толщины линий**  Тип (русск)  Тип (англ)  № типа  Сплошная  SolldLn  0  Точечная  DottedLn  1  Штрихпунктирная  CenterLn  2  Пунктирная  DashedLn  3  Тип определяется пользователем  UserBitLn  4  Нормальная толщина  NormWidth  1  Тройная толщина  ThickWidth  3  **Константы типов и толщины линий**  Константа  Описание узора  № узора  EmptyFill  Сплошной цветом фона  0  SolidFill  Сплошной текущим цветом  1  LineFill  Тип ---  2  LtSlashFill  Тип // нормальной толщины  3  SlashFill  Тип // удвоенной толщины  4  BkSlashFill  Тип \\ нормальной толщины  5  LtBkSlashFill  Тип \\ удвоенной толщины  6  HatchFill  Заполнение клеткой  7  XHatchFill  Заполнение косой редкой клеткой  8  InterLeaverFill  Заполнение косой частой клеткой  9  WideDotFill  Заполнение редкими точками  10  CloseDotFill  Заполняется частыми точками  11  UserFill  Определяется пользователем  12 |