

Комитет по образованию Администрации г. Подольска
Муниципальное образовательное учреждение «Лицей № 26»

«Алгоритм и программа
построения графика функции»
Методическая разработка урока

Кривко-Красько Сергей Васильевич
учитель информатики

г. Подольск, 2008 г.

Пояснительная записка

Уроки по построению графиков функции проводятся на заключительных этапах изучения программирования и позволяют в концентрированной форме использовать и отработать полученные навыки программирования. Поэтому, хотя компьютерная графика и не входит в программу курсов, я привожу разработку именно таких уроков.

Их разработка рассчитана как на возможное использование языка программирования Basic (система программирования Qbasic или QuickBasic), так и языка Паскаль (система программирования TurboPascal, PascalABC). Предварительно не только изучены основные алгоритмические структуры (ветвление, цикл), но и программирование графики. На Бэйсике это операторы графики, в Паскале – процедуры и функции модуля Graph.

Методика без проблем модифицируется на использование систем визуального программирования Visual Basic, Borland Delphi, позволяя использовать дополнительные элементы интерфейса (меню, полосы прокрутки и т.п.).

Данная методика неоднократно использовалась в классах различного профиля, с использованием всех вышеназванных систем программирования, как в 10-х 11-х классах, так в текущем учебном году в 9-ом классе.

Обычно при построении графика функции различными авторами (например, Культиным Н.) предлагается задавать в качестве исходных данных границы изменения аргумента и значения функции. Я предпочитаю алгоритм строить так, как он строится на уроках математики в школе и к чему привыкли ученики, т.е. сначала надо нарисовать оси (а значит надо выбрать место на экране для точки – начала координат), затем выбрать масштаб, и потом уже построить график.

При объяснении материала, в зависимости от уровня класса, можно ограничиться изложением методики и общей схемой программы, можно разобрать с участием учеников алгоритм основной части программы – построения самого графика.

Следует отметить, что использования языка Паскаль накладывает ряд специфических проблем, связанных с правильным выбором типов переменных и их преобразования.

Тема урока: Алгоритм и программа построения графика функции.

Тип урока: изучение нового материала в форме сочетания беседы и практической работы с первичным контролем (двухчасовое занятие).

Обучающая цель: изучение способа построения графика функции, с разработкой алгоритма и программы.

Развивающая цель: развитие логического и алгоритмического мышления учащихся, дальнейшее развитие умений и навыков в составлении программ; развитие творческих способностей учащихся.

Воспитательная цель: активизация познавательного интереса к математике и информатике.

| Этапы урока | Тип работы | Работа учащихся |
|--|--|--|
| 1.Объяснение цели урока | выступление учителя | запись темы урока |
| 2.Повторение а) операторы цикла и процедуры графики б) повторение понятий "функция" и "график" | фронтальный опрос беседа | устные ответы учащихся устные ответы |
| 3.Объяснение нового материала б) анализ проблемы изображения графика функции на экране в) метод пересчета координат г) общая схема программы | беседа рассказ учителя, слайд презентации рассказ | анализ запись в тетради (консп.) составление плана |
| д) алгоритм построения графика функции е) анализ возникающих проблем | рассказ учителя, слайд презентации беседа | построение блок-схемы анализ |
| 4.Практическая работа а) постановка обязательного задания "построить графики различных функций" б) постановка дополнительных заданий в) выполнение практической работы за компьютерами г) анализ практической работы д) завершение практической работы | рассказ рассказ учителя контроль и помощь учителя беседа контроль и помощь учителя | запись в тетради запись в тетради самостоятельная работа за компьютерами вопросы и ответы уч-ся самостоятельная работа за компьютерами |
| 5.Задание на дом | | запись в тетради |

Методика разработки алгоритма и программы построения графика функции

График функции $f(x)$ – это множество точек на координатной плоскости $\{ (x, f(x)) \}$, где x принадлежит области определения функции.

Для того чтобы построить на экране монитора изображение графика функции, можно было бы перебрать различные значения аргумента x в некотором интервале с определенным шагом, вычислить значения функции и вывести соответствующие точки на экран. Но, быстрее всего, в этом случае мы бы на экране ничего не увидели или увидели не то, что надо. Например, если $f(x)=\sin(x)$, то мы бы увидели прямую линию по верхнему краю экрана (т.к. значения $f(x)$ изменялись бы от -1 до 1).

Ясно, что прежде чем рисовать график функции на экране, как и при его изображении на бумаге, надо выбрать местоположение точки – начала координат (X_0, Y_0) и масштабы, т.е. размеры единичных отрезков по осям (назовем их M_x и M_y), а для правильного изображения графика надо координаты точек графика, в дальнейшем называемые – математические координаты, (x, y) пересчитать в координаты соответствующей точки на экране – "экранные" координаты $(X_э, Y_э)$ (см. рис.1).

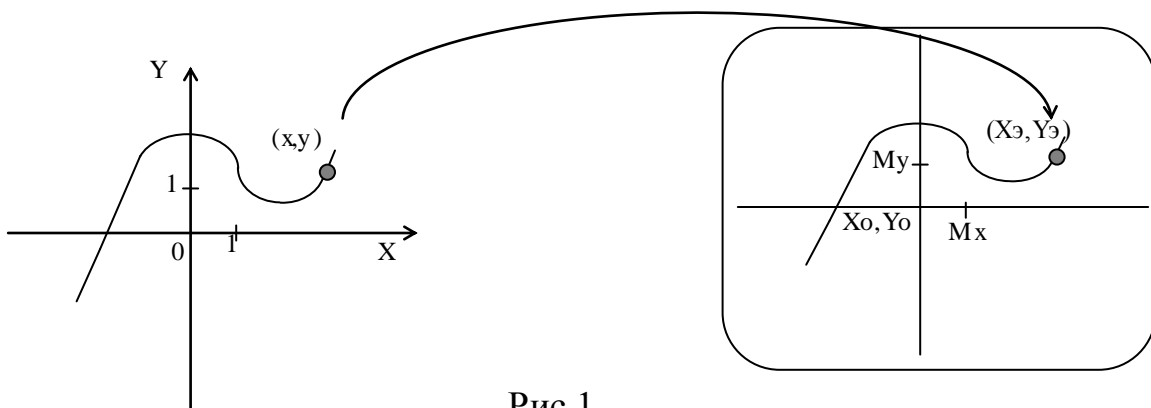


Рис.1

Несложно получить формулы для этого пересчета:

$$X_э = X_0 + M_x \cdot X, \quad (1)$$

$$Y_э = Y_0 - M_y \cdot Y. \quad (2)$$

Точка $(0,0)$ при этом перейдет в экранную точку (X_0, Y_0) , точка $(1,0)$ в точку $(X_0 + M_x, Y_0)$, точка $(0,1)$ – в точку $(X_0, Y_0 - M_y)$.

Для вывода всего графика на экран удобнее организовать цикл по значениям экранной координаты $X_э$, т.к. легче определить границы для изменения этой величины.

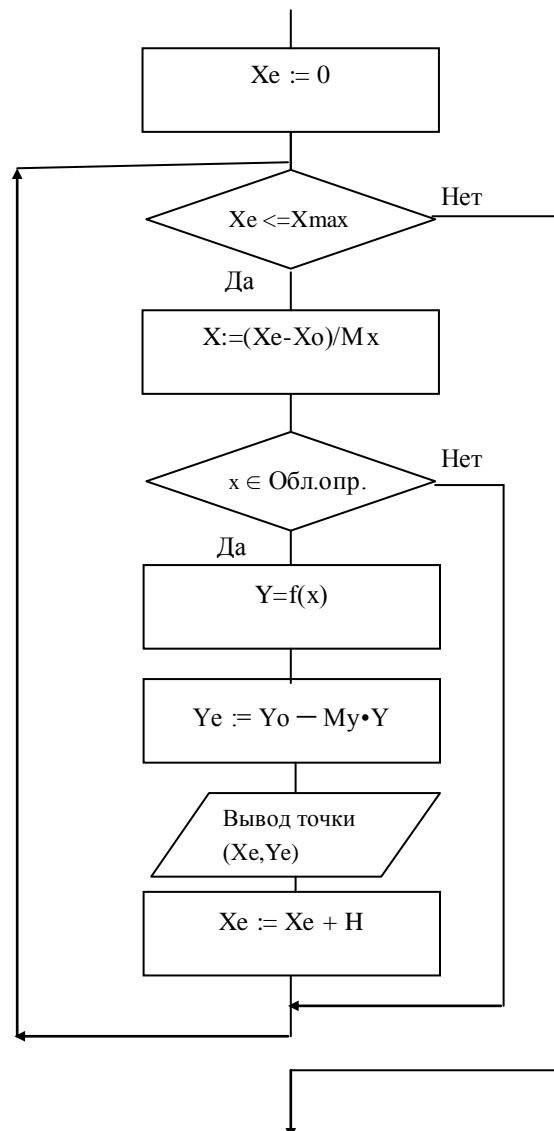
Порядок расчета координат получается следующий:

$$X_э \longrightarrow X = (X_э - X_0)/M_x \longrightarrow Y = f(x) \longrightarrow Y_э = Y_0 - M_y \cdot Y. \longrightarrow \text{Точка } (X_э, Y_э)$$

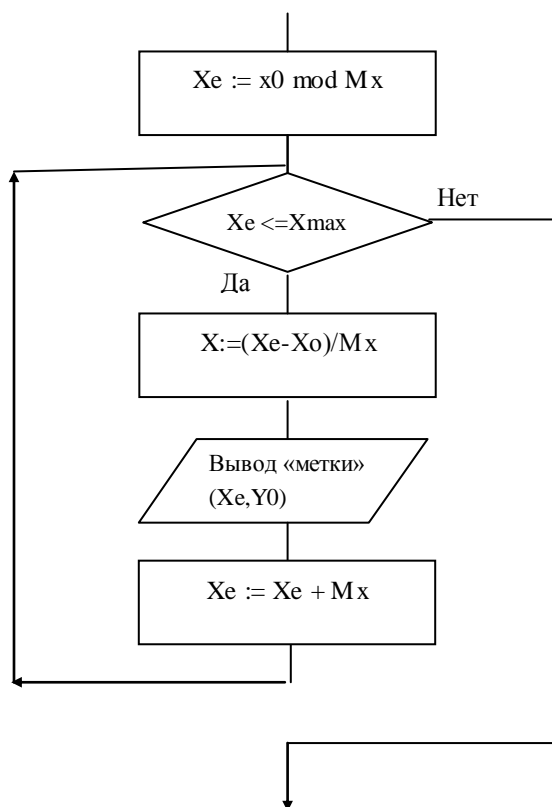
Общая схема построения программы

1. Ввод исходных данных (X_0 , Y_0 , M_x , M_y).
2. Изображение осей.
3. Изображение разметки осей (или же координатной сетки).
4. Изображение графика функции.

Алгоритм построения графика функции (блок-схема)



Дополнительно может быть рассмотрен алгоритм разметки осей (на примере одной оси X):



Пояснения к алгоритмам и анализ возникающих проблем.

- X_{\max} - максимальное значение экранной координаты, зависит от используемого экранного режима;
- H – шаг изменения экранной координаты, может быть равен 1, но тогда график становится недостаточно «плотным», лучше использовать шаг более мелкий;
- Условие проверки, принадлежит ли значение x области определения функции, зависит от самой функции и не может быть унифицировано;
- «вывод точки (X_e, Y_e) » может быть организован как вывод окружности небольшого радиуса, как вывод линии от предыдущей точки к текущей (в этом случае возникает проблема 1-ой линии, путь решения – или сделать ее невидимой, или же до цикла просчитать координату 1-ой точки);
- «вывод метки» на оси можно организовать также как вывод окружности или же небольшой линии – штриха, возможен также вывод линии координатной сетки от $(X_e, 0)$ до (X_e, Y_{\max}) .

Необходим разбор проблемы типов данных. Переменные X, Y очевидно, типа real. Если используется шаг цикла N меньший 1, то переменная Xе тоже должна быть типа real. Но при выводе точки значения координат должны быть преобразованы в integer путем округления (round или trunc).

Вычисление функции желательно оформить в виде подпрограммы Function. Возможно также в этой же подпрограмме определять логическую переменную, соответствующую условию « $x \in \text{Обл.опр.}$ »

Дополнительно в алгоритме следует предусмотреть проверку ограничений $Y_e \geq 0$, $Y_e \leq Y_{\max}$, иначе возможен аварийный останов программы.

Для построения графика предлагается использовать функции:

$y = \sin(x)$, $y = x * \sin(x)$, $y = \text{tg}(x)$, $y = (x-1)*(x+2)/(x^2+1)$, $y = (x-1)*(x+2)/(x*(x-2))$,

$y = \sqrt{x^2 - 1} / x$ и др.

Смена графика производится перепрограммированием подпрограммы вычисления функции.

Дополнительные задания (они же домашние задания):

1. Выделить график в областях возрастания и убывания разным цветом.
2. Выделить на графике точки локальных максимумов и локальных минимумов.
3. Разработать алгоритм и программу построения графика функции, заданной в полярных координатах (R, φ) $R = R(\varphi)$. Например $R = R_0 * (a * \sin(c * \varphi) + b)$.

Возможные пути решения:

В 1-ой задаче можно сравнить значение функции со значением в предшествующей точке и в зависимости от результата сравнения выбрать цвет.

Аналогично во 2-ой задаче можно сравнить значение функции с ее значениями в предшествующей и последующей точках.

Для 3-ей задачи необходимо дать ученикам формулы пересчета координат:

$x = x_0 + R * \cos(\varphi)$; $y = y_0 - R * \sin(\varphi)$.

Возможный вариант программы: функция $y = \sqrt{x^2 - 1}/x$

```
program graphic;
  uses graph;
  var x,y,xe,ye,h:real; fl:boolean;
      DriveVar, ModeVar, xmax, ymax, Mx, My, x0, y0: integer;

  procedure F(x:real; var y:real; var fl:boolean); 'вычисление функции
  begin
    fl:=true; y:=0;
    if x*x-1>=0 then y:=sqrt(x*x-1)/x else fl:=false;
  end;

begin
  Xmax:=640; Ymax:=480; h:=0.01;
  X0:= Xmax mod 2; Y0:=Ymax mod 2; 'начало координат в центре экрана
  writeln('введите масштабы Mx, My); Readln(Mx,My);
  DriverVar:=Detect;
  InitGraph(DriverVar, ModeVar, '\TP\BGI'); 'инициализация графики
  SetLineStyle(0,0,2);

  line(x0,0,x0,ymax); line(0,y0,xmax,y0); 'оси
  SetLineStyle(1,0,1);
  xe:=x0 mod mx; 'разметка осей - координатная сетка
  while xe<=Mx do
    begin
      line(round(xe),0,round(xe),ymax);
      xe:=xe+Mx;
    end;
  ye:=y0 mod my;
  while ye<=My do
    begin
      line(0,round(ye), xmax, round(xe));
      ye:=ye+Mx;
    end;

  SetLineStyle(0,0,1); SetColor(Red);
  xe:=0; 'построение графика
  while xe<=xmax do
    begin
      x:=(xe-x0)/mx;
      F(x,y,fl);
      if fl then
        begin
          ye:=y0-mY*y;
          if (ye>=0) and (ye<=ymax) then
            circle(round(xe),round(ye),2);
          end;
          xe:=xe+h;
        end;
    end;

  readln; closegraph;
end.
```