

**МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**Петровская средняя общеобразовательная школа**

---

143395, Московская область,  
Наро-Фоминский район, с. Петровское  
тел. 8(496) 342-32-14

e-mail: [petrchool@mail.ru](mailto:petrchool@mail.ru)  
<http://nf-petrov.edumsko.ru/>

**ПРОЕКТ**  
**ИССЛЕДОВАНИЕ ИНФОРМАЦИОННЫХ МОДЕЛЕЙ КАК ОДИН ИЗ**  
**АСПЕКТОВ ПОДГОТОВКИ К ЕГЭ**

**СКОРНЯКОВА ТАТЬЯНА**  
**ЕВГЕНЬЕВНА**

## СОДЕРЖАНИЕ:

1. ВВЕДЕНИЕ
2. ИССЛЕДОВАНИЕ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ.
  - 1) РАЗБОР ЗАДАЧИ C1 ДЕМОВЕРСИИ ЕГЭ 2012
  - 2) ВИЗУАЛИЗАЦИЯ РЕШЕНИЯ ЗАДАЧИ
3. СПИСОК ЛИТЕРАТУРЫ

## ВВЕДЕНИЕ

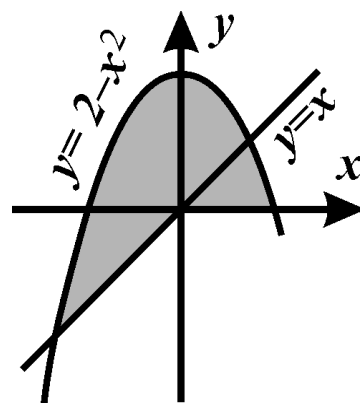
Успешная сдача ЕГЭ зависит не только от знаний и умения решать те или иные задачи, но и от того, как ученик освоил формат экзамена. Каждое задание экзаменационной работы характеризуется не только проверяемым содержанием, но и проверяемыми умениями. Кодификатор определяет две группы требований к уровню подготовки выпускников: с одной стороны, знать/понимать/уметь и, с другой стороны, использовать приобретенные знания и умения в практической деятельности и повседневной жизни.

Задание С1, предполагающее поиск и устранение ошибок в уже имеющейся программе и ее доработку, на мой взгляд в немалой степени не решается учащимися из-за незнания алгебры. На языке алгебры формальные модели записываются с помощью уравнений, точное решение которых основывается на поиске равносильных преобразований алгебраических выражений, позволяющих выразить переменную величину с помощью формулы. Точные решения существуют только для некоторых уравнений определенного вида (линейные, квадратные, тригонометрические и др.), поэтому для большинства уравнений приходится использовать методы приближенного решения с заданной точностью (графические, числовые и др.). На примере задачи С1 можно рассмотреть тему Моделирование и формализация. Если учащиеся в процессе подготовки к ЕГЭ не просто будут рассматривать различные варианты печатных картинок к заданию С1, а напишут программу, которая заштрихует заданную область, степень усвоения материала будет выше.

## ИССЛЕДОВАНИЕ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ.

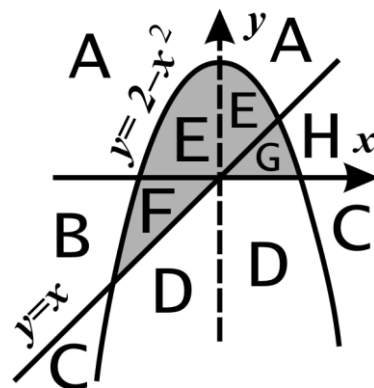
1) Демо-версия ЕГЭ 2012 года предлагает следующую задачу С1

Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости ( $x$ ,  $y$  – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы. Программист торопился и написал программу неправильно.



*ПАСКАЛЬ*

```
var x, y: real;
begin
  readln(x, y);
  if y >= x then
    if y >= 0 then
      if y <= 2 - x * x then
        write('принадлежит')
      else
        write('не принадлежит')
    end.
end.
```



Последовательно выполните следующее:

1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D, E, F и G). Точки, лежащие на границах областей, отдельно не рассматривать.

В столбцах условий укажите "да", если условие выполнится, "нет" если условие не выполнится, "—" (прочерк), если условие не будет проверяться, «не изв.» если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце "Программа выведет" укажите, что программа выведет на экран. Если

программа ничего не выводит, напишите "—" (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «не изв». В последнем столбце укажите "да" или "нет".

2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы) .

Решение: [1]

- 1) начнем заполнять таблицу, выписывая истинность каждого из трёх условий
- 2) условие  $y \geq x$  истинно выше прямой  $y=x$ , то есть в областях А, В, Е, F

Область	$y \geq x?$	$y \geq 0?$	$y \leq 2-x^2?$	вывод	верно?
А	да				
В	да				
С	нет				
Д	нет				
Е	да				
Ф	да				
Г	нет				

- 3) условие  $y \geq 0$  истинно выше прямой  $y=0$ , то есть в областях А, Е, Г, однако это условие проверяется только тогда, когда первое условие,  $y \geq x$ , истинно; поэтому для всех областей, где первое условие неверно (это области С, Д, Г), сразу в столбце второго условия ставим прочерк (условие не будет проверяться)

Область	$y \geq x?$	$y \geq 0?$	$y \leq 2-x^2?$	вывод	верно?
А	да	да			
В	да	нет			
С	нет	—			
Д	нет	—			
Е	да	да			
Ф	да	нет			

G	нет	–			
---	-----	---	--	--	--

4) третье условие выполняется для областей «внутри» параболы, то есть для E, F, G, D; однако оно проверяется только тогда, когда первые два истинны (для A и E), в остальных строках ставим прочерк:

Область	$y >= x$ ?	$y >= 0$ ?	$y <= 2 - x * x$ ?	вывод	верно?
A	да	да	нет		
B	да	нет	–		
C	нет	–	–		
D	нет	–	–		
E	да	да	да		
F	да	нет	–		
G	нет	–	–		

5) как следует из текста программы, она выведет что-то на экран только в том случае, когда выполняются первые два условия и программа выходит на третье: для области A будет выведено «не принадлежит», для области E – «принадлежит», именно в этих двух случаях программа работает правильно, в остальных – нет:

Область	$y >= x$ ?	$y >= 0$ ?	$y <= 2 - x * x$ ?	вывод	верно?
A	да	да	нет	не принадлежит	да
B	да	нет	–	–	нет
C	нет	–	–	–	нет
D	нет	–	–	–	нет
E	да	да	да	принадлежит	да
F	да	нет	–	–	нет
G	нет	–	–	–	нет

б) для того, чтобы доработать программу, проще всего составить одно сложное условие, описывающее всю заштрихованную область

7) в данном случае удобно представить данную область в виде объединения областей, первая из которых включает области E+G, а вторая – области E+F

8) область E+G соответствует условию  $(y >= 0)$  and  $(y <= 2 - x * x)$

9) область E+F соответствует условию  $(y \geq x)$  and  $(y \leq 2-x^2)$

10) объединение областей выполняется с помощью операции ИЛИ (or), так что полное условие принимает вид

$$(y \geq 0) \text{ and } (y \leq 2-x^2) \text{ or } (y \geq x) \text{ and } (y \leq 2-x^2)$$

поскольку операция И (and) имеет более высокий приоритет, чем ИЛИ (or), порядок выполнения операций тут правильный; в случае сомнений можно поставить дополнительные скобки:

$$((y \geq 0) \text{ and } (y \leq 2-x^2)) \text{ or } ((y \geq x) \text{ and } (y \leq 2-x^2))$$

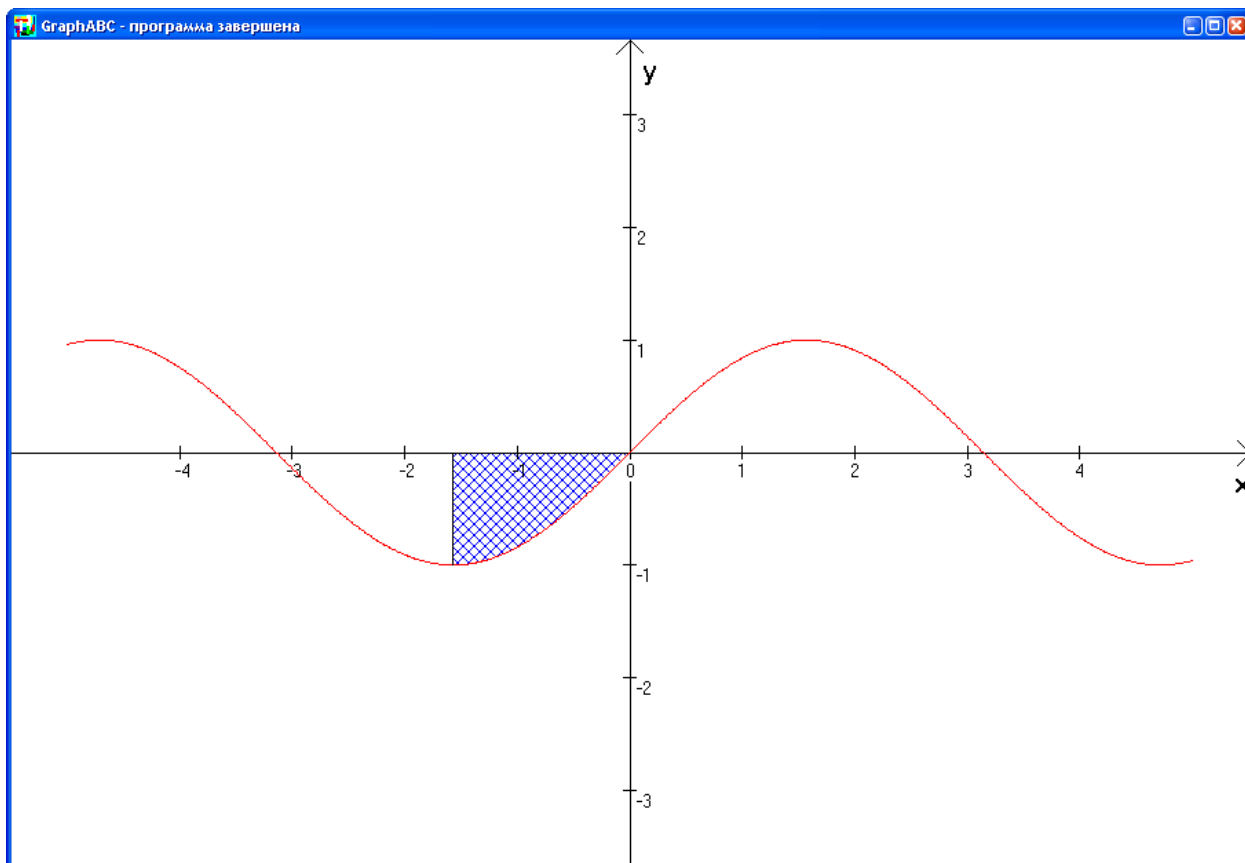
11) поскольку в обоих условиях есть условие  $y \leq 2-x^2$ , запись можно немного сократить:

$$(y \leq 2-x^2) \text{ and } ((y \geq x) \text{ or } (y \geq 0))$$

2) Работу с этой задачей можно визуализировать следующим образом:

1. Построение графиков функций;
2. Нахождение точек пересечения графиков;
3. Штриховка области, заданной по условию;
4. Определение принадлежности точки заданной закрашенной области;
5. Нахождение площади заштрихованной фигуры;

Сначала можно предложить ребятам залить указанную область при условии попадания в нее точки.



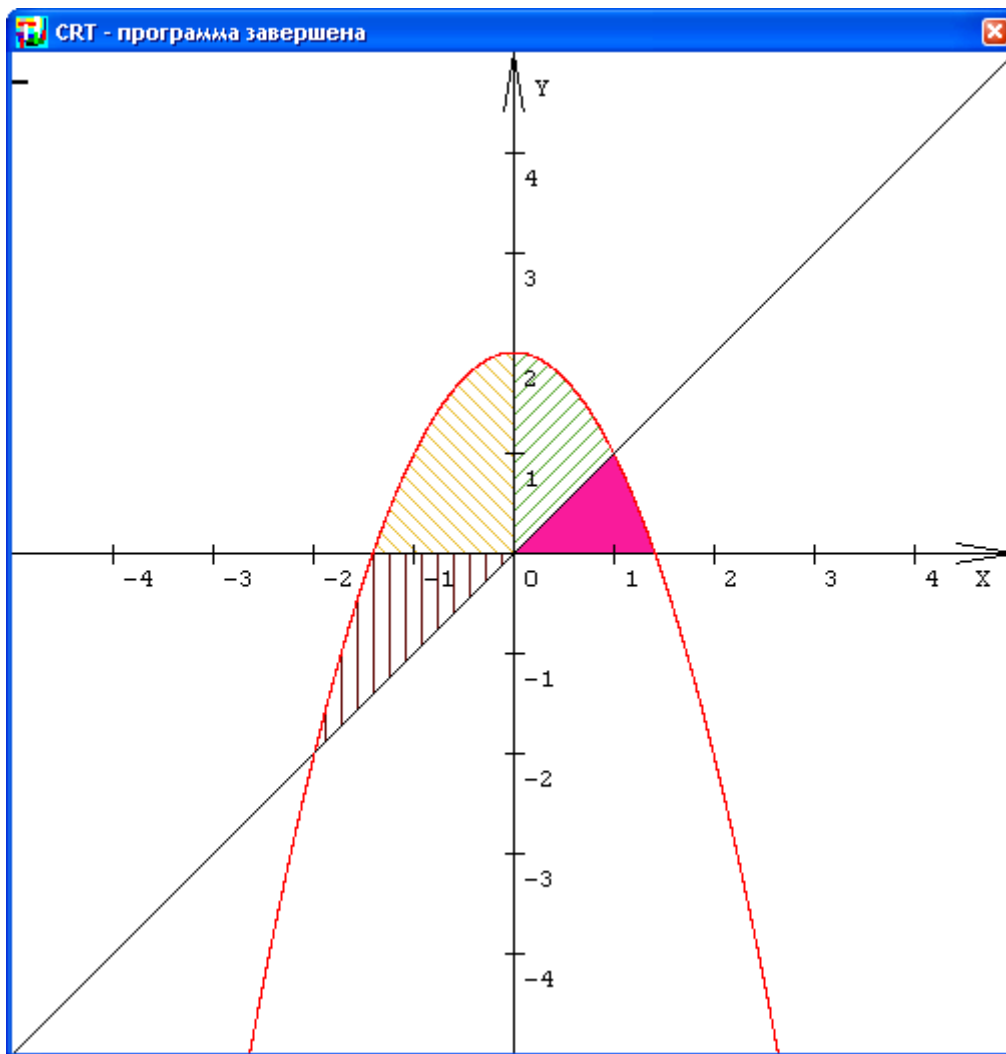
А далее при условии попадания в одну из областей А,В,С,Д,Е,Ф или G выделять ее отличающимся от остальных способом или закрашивать все 4 области случайным образом подобранными вариантами заливки.

```

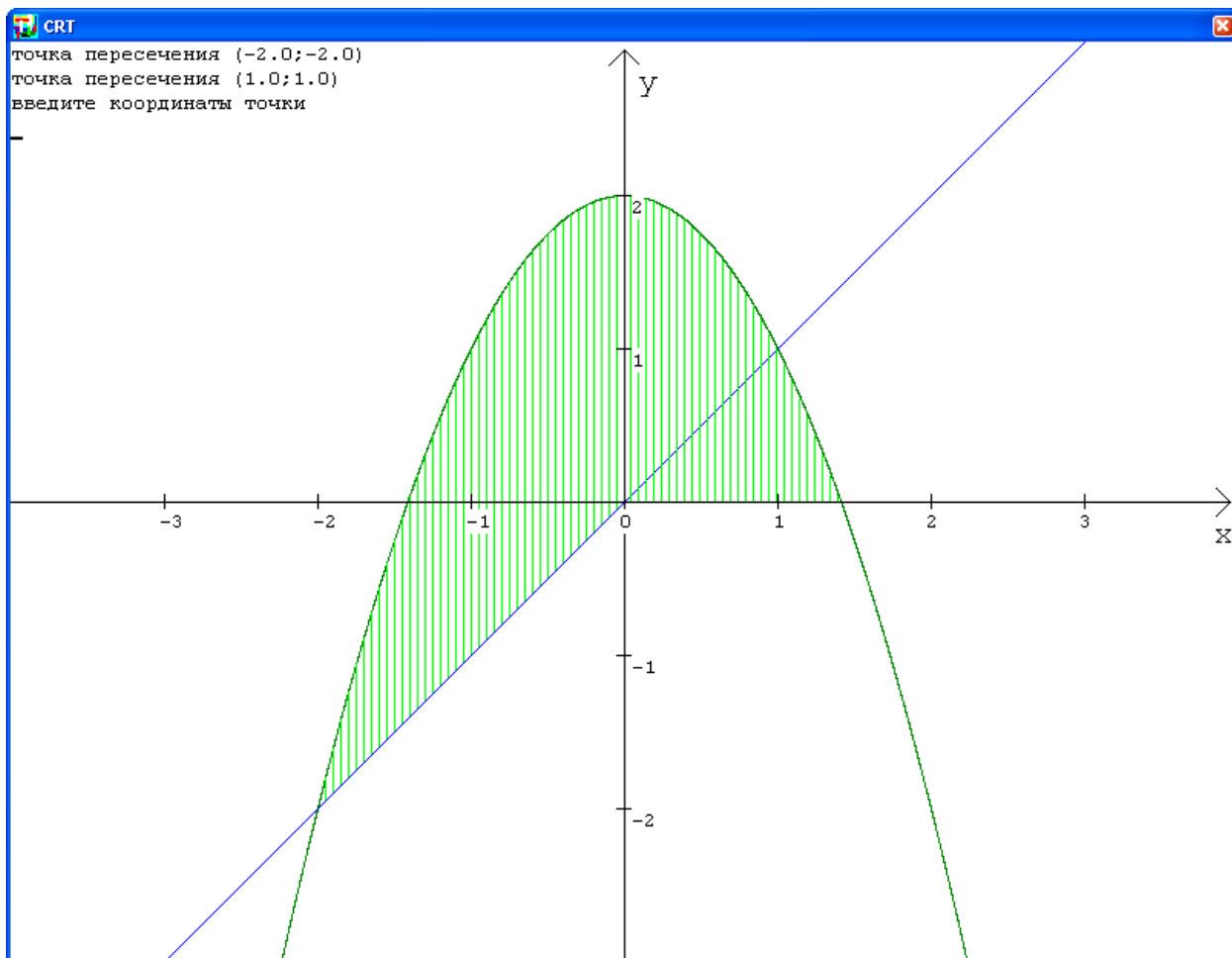
SetBrushStyle(Random(7));
floodfill(x0-
5,y0+1,rgb(random(255),random(255),random(255)));
SetBrushStyle(Random(7));
floodfill(x0+5,y0-
1,rgb(random(255),random(255),random(255)));
SetBrushStyle(Random(7));
floodfill(x0-5,y0-
1,rgb(random(255),random(255),random(255)));
SetBrushStyle(Random(7));
floodfill(x0+1,y0-
5,rgb(random(255),random(255),random(255)));

```





Затем нужно предложить обобщить решение для случая, если невозможно указать точно координаты точки для заливки, рассмотреть возможность организации штриховки как процедуры.



В приведенной ниже программе выдается запрос на ввод координат точки. В случае, если точка принадлежит закрашенной области, заштрихованная область изменяет цвет.

```

program grafik_C1;
uses graphABC, crt;
const W=800; // размеры экрана
      H=600;
      a=-4; // интервал для построения графиков
      b=4;
var x, y: real;
    x0, y0: integer; // начало координат
    m: real; // масштаб изображения
    xEkr, yEkr: integer; // экранные координаты точки
{-----

```

```

    F1: значение функции  $y=2-x*x$ 
-----}
function  F1(x:real):real;
begin
F1:=2-x*x;
end;
{-----}

    F2: прямая  $y=x$ 
-----}
function  F2(x:real):real;
begin
F2:=x;
end;
{-----}

    F: разность функций
-----}
function  F(x:real):real;
var y:real;
begin
F:=F1(x)-F2(x);
end;
{-----}

    Cross: точки пресечения графиков
    метод перебора
-----}
procedure Cross;
const eps=0.001;
begin
x:=a;
while (x<=b) do begin

```

```

        //найденное решение отличается от
истинного не более, чем на eps
        if F(x)*F(x+eps)<0 then
            writeln ('точка пересечения
(' ,x:3:1, ';' ,F1(x):3:1, ') ');
            x:=x+eps;
        end;
end;
{-----}
    Axis: оси координат
-----}
procedure Axis;
var i:integer;
begin
    //ось Ox
    line (0,y0,W-5,y0);
    //стрелки
    lineto(W-15,y0-10);
    line(W-5,y0,W-15,y0+10);
    //ось Oy и стрелки
    line (x0,5,x0,H);line (x0-10,15,x0,5);line
(x0,5,x0+10,15);
    //подписываем оси
    SetFontSize(15);
    TextOut(x0+10,15,'y');
    TextOut(W-15,y0+10,'x');
    //разметка осей единичными отрезками
    for i:=a+1 to b-1 do
        begin
            setfontsize(10);

```

```

line(trunc(i*m+x0),y0-5,trunc(i*m+x0),y0+5);
textout(trunc(i*m+x0-3),y0+5,inttostr(i));
if (i*m+y0>0) and (i*m+y0<H)and (i*m+y0<>y0)
then
    begin
        line(x0-
5,trunc(i*m+y0),x0+5,trunc(i*m+y0));
        textout(x0+5,trunc(i*m+y0),inttostr(-i));
    end;
end;
end;
{-----}
Plot: графики функций
-----}
procedure Plot;
begin
x:=a;
while x<=b do
    begin
        xEkr:=round(x*M+x0);
        yEkr:=round(-F1(x)*M+y0); //парабола
        setpixel(xEkr,yEkr,clgreen);
        yEkr:=round(-F2(x)*M+y0); //прямая
        setpixel(xEkr,yEkr,clblue);
        x:=x+0.001
    end;
end;
{-----}
Hatch: штриховка области
-----}

```

```

procedure Hatch;
begin
x:=a;
while x<=b do
    begin
        if (f(x)>=0) then line(x0+trunc(x*M),y0-
trunc(f1(x)*m),x0+trunc(x*M),y0-trunc(f2(x)*m));
        if (f(x)>=0)and (x>0)and (x<sqrt(2)) then
            line(x0+trunc(x*M),y0-
trunc(f1(x)*m),x0+trunc(x*M),y0);
        if (f(x)<=0)and (x>0)and (x<sqrt(2)) then
            line(x0+trunc(x*M),y0-
trunc(f1(x)*m),x0+trunc(x*M),y0);
        x:=x+0.05;
    end;
end;
{-----}
    основная программа
-----}
begin
setwindowsize(W,H);
//установим масштаб графика
M:=W/(b-a);
// начало координат
y0:=H div 2;
x0:=W div 2;
Axis; { оси координат }
Plot; { графики функций }
setpencolor(cllime);
Hatch;{штриховка}

```

```

Cross; {точки пресечения графиков}
writeln('введите координаты точки ');
readln(x,y);
if (y <=2-x*x) and ((y>=x) or (y>=0)) then
  begin
    writeln ('принадлежит');
    setpencolor(clred);
    Hatch; {штриховка}
  end
else begin setpencolor(clred);setbrushcolor(clred);
circle(round(x*M+x0),round(-y*M+y0),4);
setpencolor(clwhite);
Hatch; {штриховка}writeln ('не принадлежит');
end;
end.

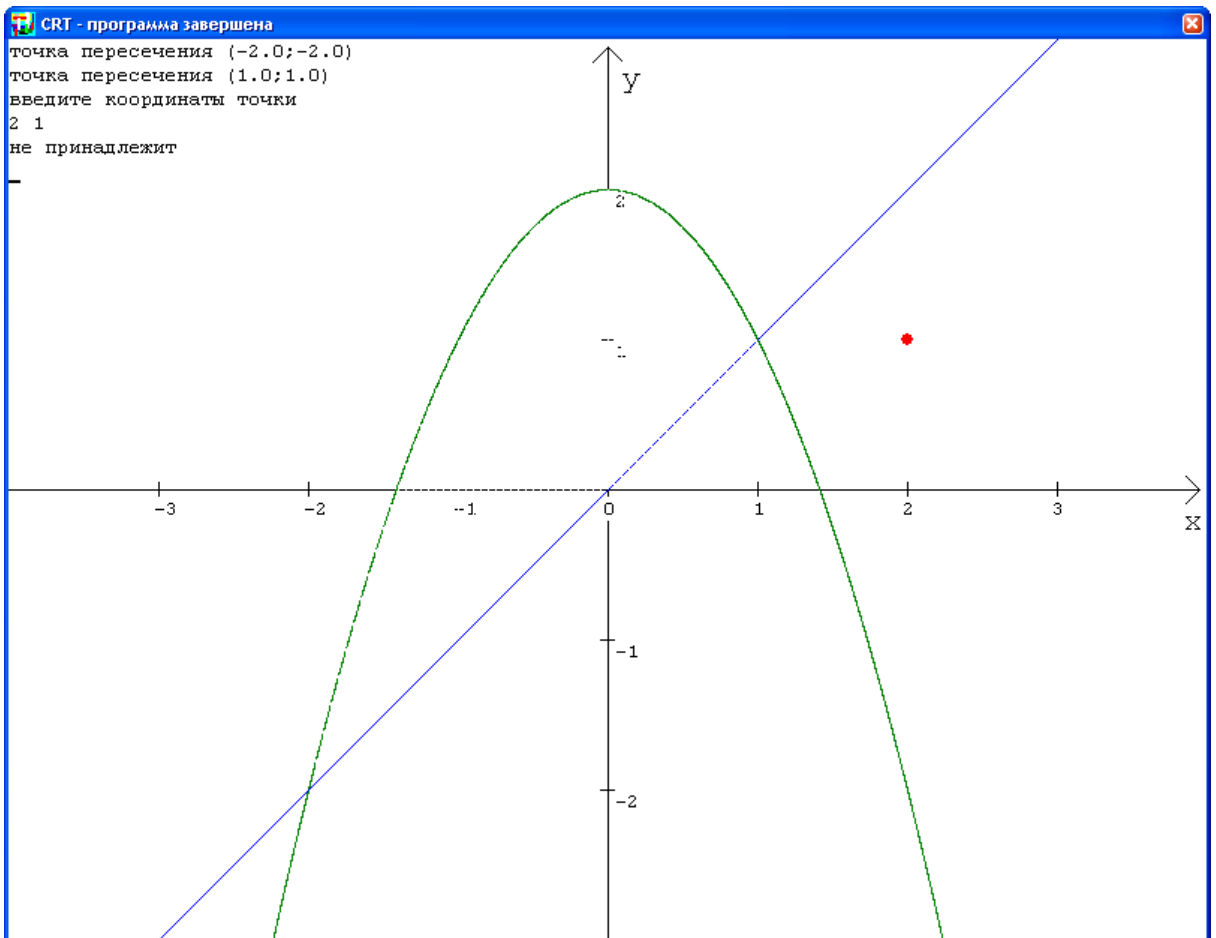
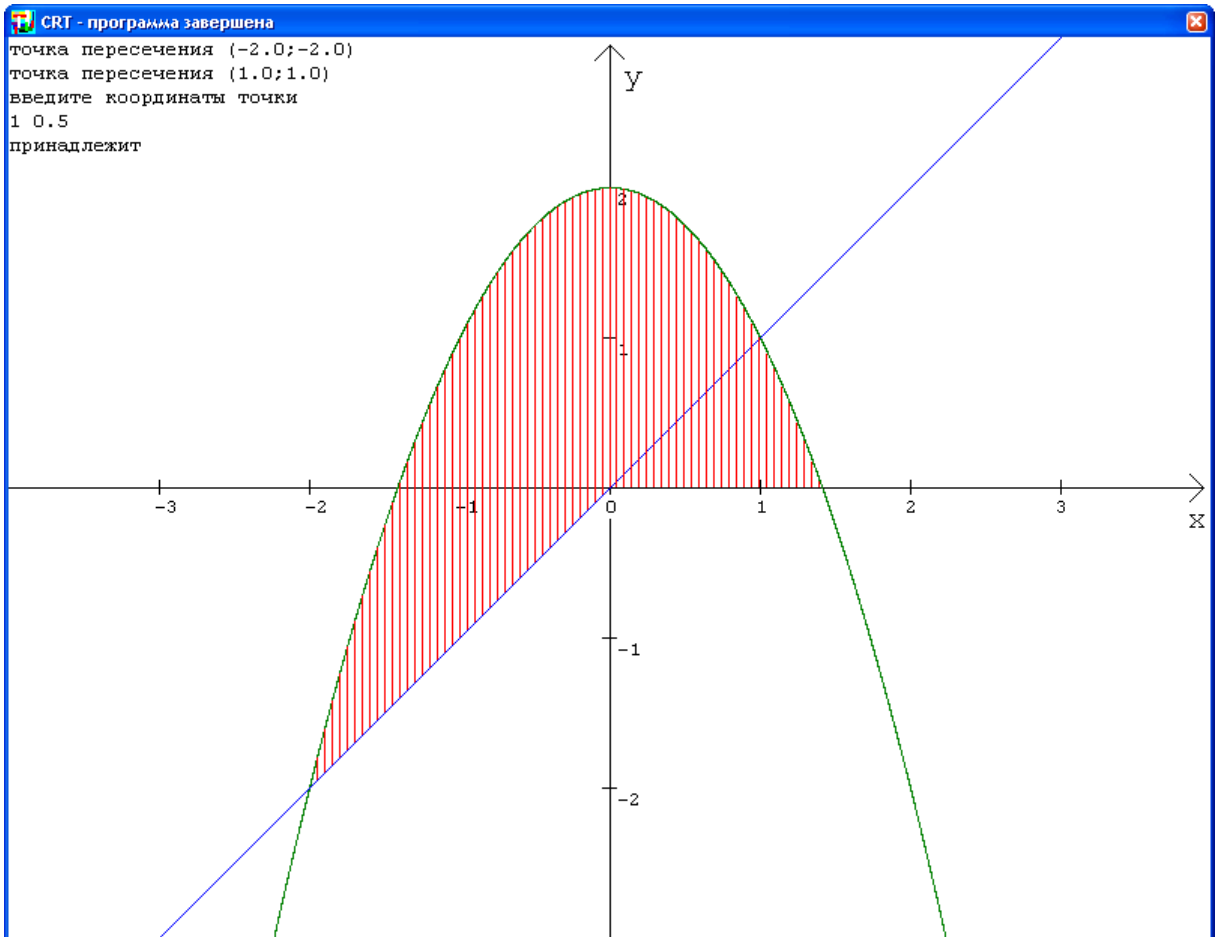
```

The image shows a screenshot of a Pascal ABC IDE window. The title bar reads "Pascal ABC". The menu bar includes "Файл", "Правка", "Вид", "Программа", "Сервис", and "Помощь". The toolbar contains icons for file operations and execution. The main text area shows the following code:

```
{Требовалось написать программу, при выполнении которой  
с клавиатуры считываются координаты точки  
на плоскости (x, y - действительные числа) и  
определяется принадлежность этой точки заданной  
закрашенной области (включая границы)}  
program grafik_C1;  
uses graphABC, crt;  
const W=800; // размеры экрана  
      H=600;  
      a=-4; // интервал для построения графиков  
      b=4;  
var x, y: real;  
    x0, y0: integer; // начало координат  
    m: real; // масштаб изображения  
    xEkr, yEkr: integer; // экранные координаты точки  
  
-----  
F1: значение функции  $y=2-x*x$   
-----  
function F1(x: real): real;  
begin  
F1:=2-x*x;  
end;  
  
-----  
F2: прямая  $y=x$   
-----  
function F2(x: real): real;  
begin  
F2:=x;  
end;  
  
-----  
F: разность  
-----  
function F(x: real): real;  
var y: real;  
begin  
F:=F1(x)-F2(x);  
end;
```

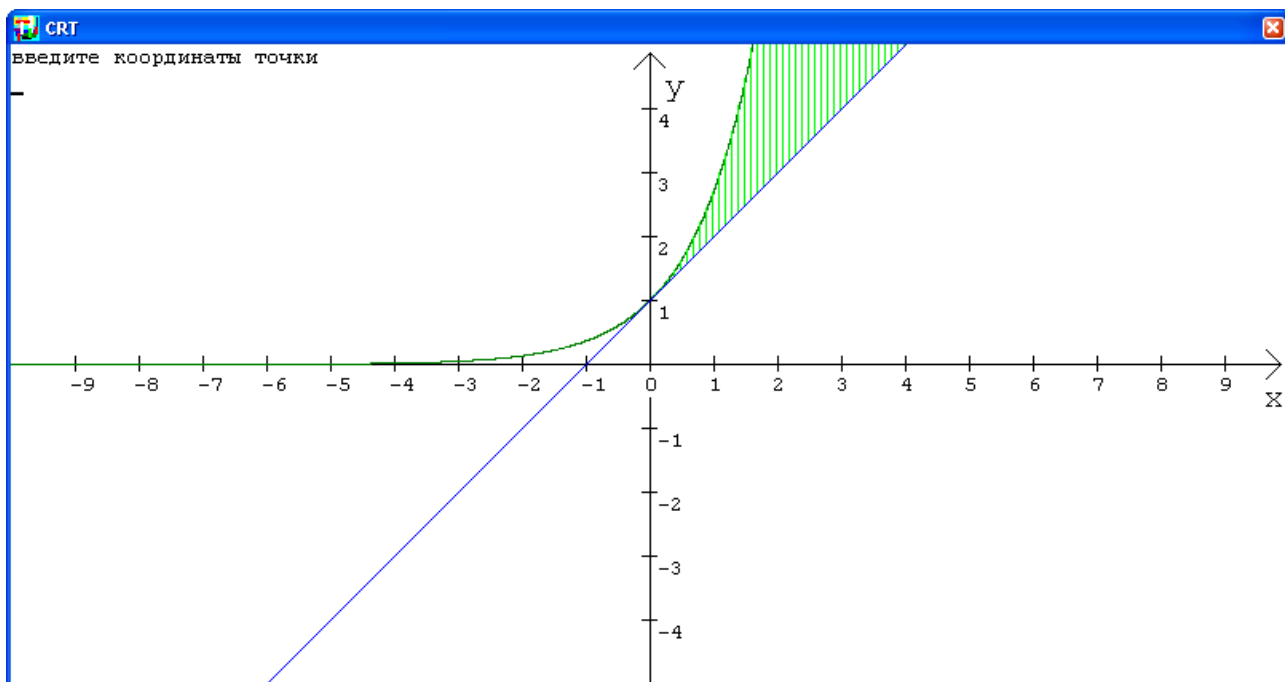
At the bottom of the window, the status bar shows "Строка: 4" and "Столбец: 1".





Если в уже написанной программе изменить исходные функции, заштриховать другую область – получается задача-тренажер.

```
{-----F1: экспонента-----}  
function F1(x:real):real;  
begin  
F1:=exp(x);  
end;  
{-----F2: прямая y=x+1-----}  
function F2(x:real):real;  
begin  
F2:=x+1;  
end;
```



Приведенная на рисунке область штрихуется следующим образом:

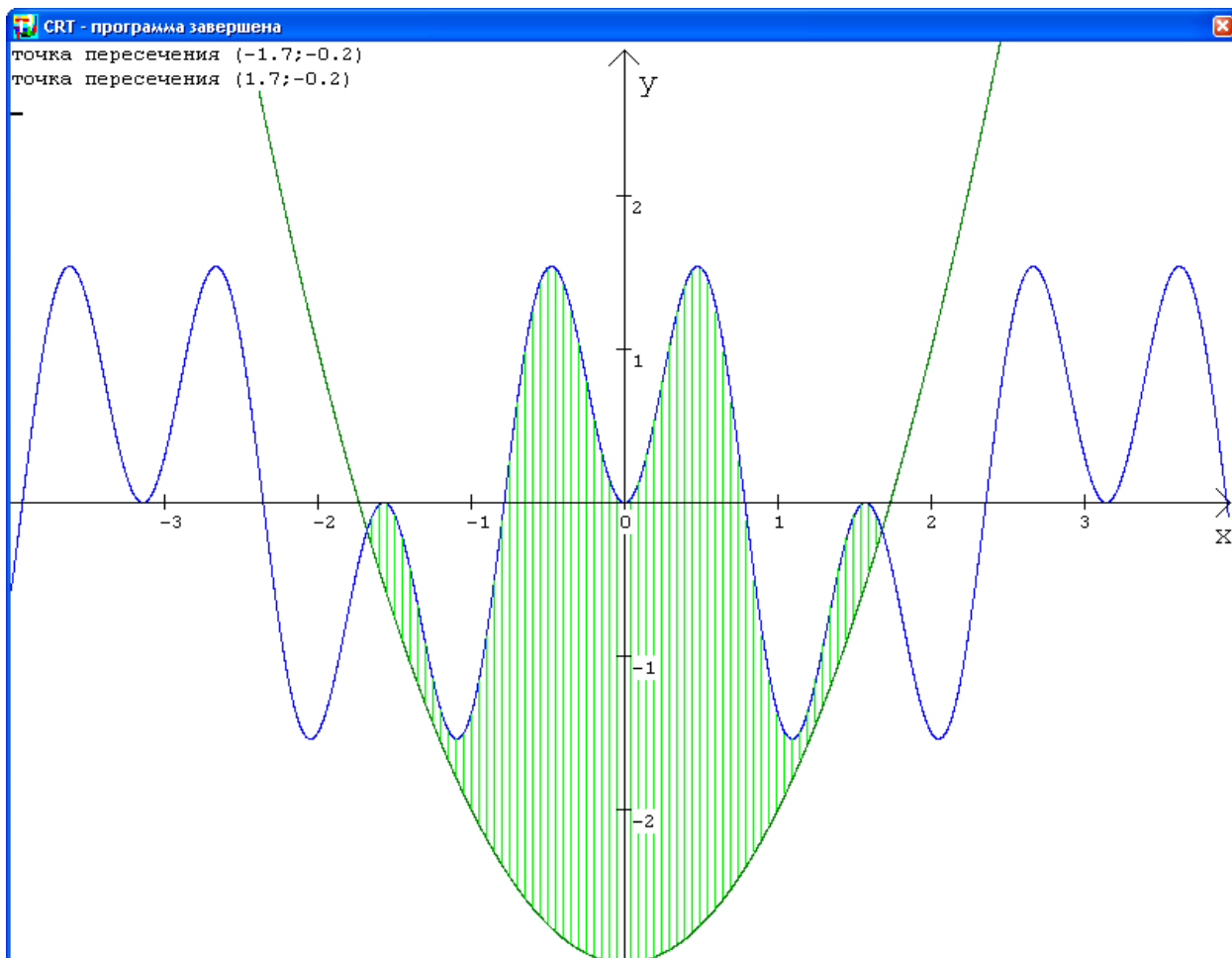
```
while x<=b do begin  
    if (f1(x)-f2(x)>=0) and (x>0) then  
line(x0+trunc(x*M),y0-trunc(f1(x)*m),x0+trunc(x*M),y0-  
trunc(f2(x)*m));
```

```

x:=x+0.1;
end;

```

Использование для построения функций, графики которых выглядят непривычно, которые сложно построить на листе бумаги в клетку с использованием карандаша и линейки, поможет сделать работу привлекательнее и для слабых учащихся.



Гораздо сложнее организовать штриховку области в круге, где выше или ниже прямой могут располагаться сразу две точки окружности. Необходимо рассматривать не уравнение окружности, а две функциональные зависимости для верхней дуги и нижней дуги отдельно.

```

{-----F3: окружность верхняя дуга-----}
function F3(x:real):real;
begin
  F3:=sqrt(r*r-(x-xC)*(x-xC))+yC;

```

```

end;
{-----F4: окружность нижняя дуга-----}
function F4(x:real):real;
begin
F4:=-sqrt(r*r-(x-xC)*(x-xC))+yC;
end;

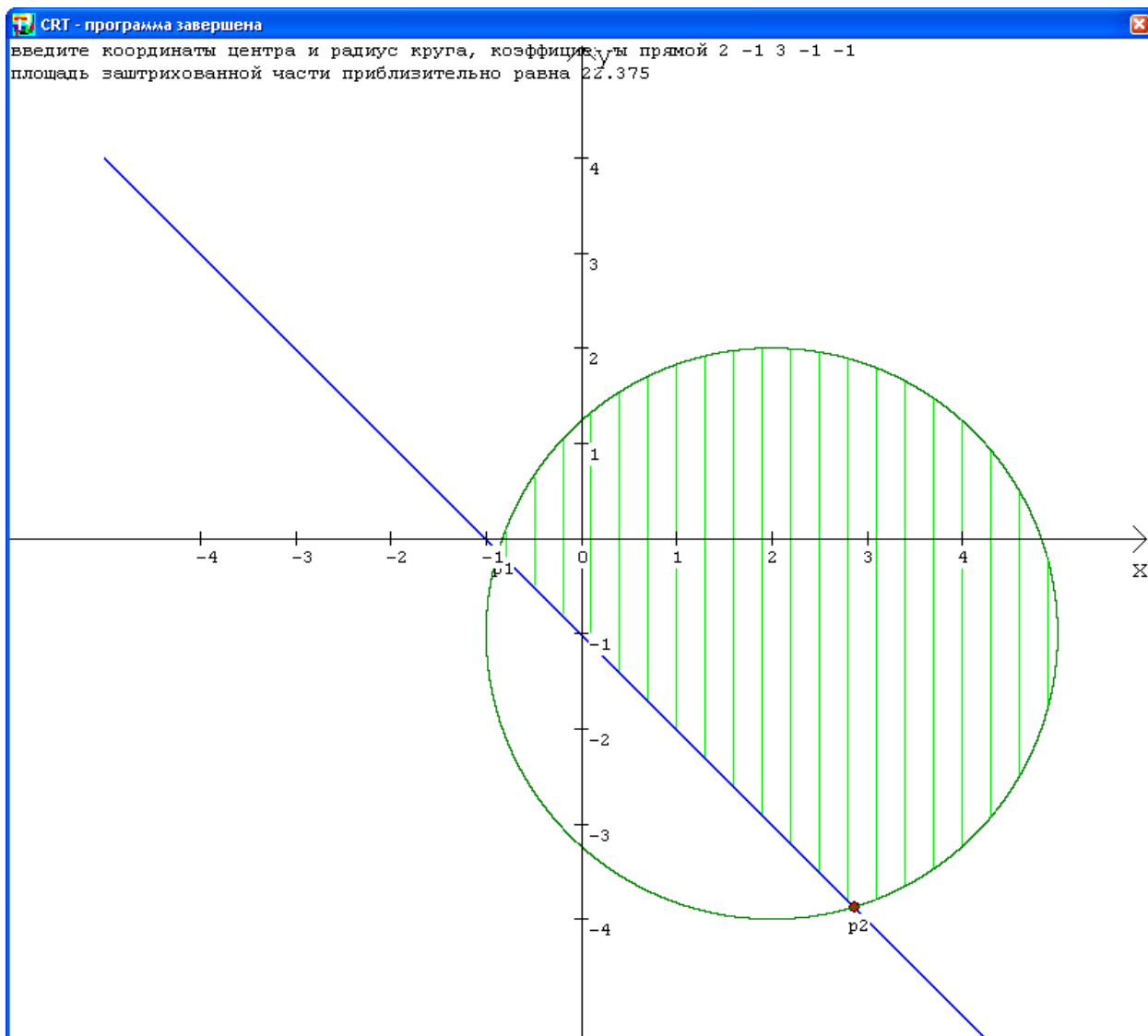
```

**Фрагмент программы штриховки области выглядит так:**

```

begin if (x>=xC-r) and (x<=xC+r)
      then begin
            if (F3(x)>F2(x)) and (F4(x)>F2(x))
            then line(x0+trunc(x*M),y0-
trunc(f3(x)*m),x0+trunc(x*M),y0-trunc(f4(x)*m));
            if (F3(x)>F2(x)) and (F4(x)<F2(x))
            then line(x0+trunc(x*M),y0-
trunc(f3(x)*m),x0+trunc(x*M),y0-trunc(f2(x)*m));
            end;
      x:=x+sh;
end;

```



Как правило, в задаче C1 требуется использовать знания из курса математики (решение уравнений, графики функций, область допустимых значений, составление уравнений прямой по приведенному графику), нужно уметь, прежде всего, разбираться в серии вложенных условных операторов в полной и неполной форме. Возможность увидеть результаты своей работы на экране монитора будет способствовать более успешной подготовке к решению задач C1.

## СПИСОК ЛИТЕРАТУРЫ

1. <http://kpolyakov.narod.ru/school/ege.htm>
2. Угринович Н.Д. информатика и информационные технологии. Учебник для 10-11 классов.- М. Бинوم. Лаборатория знаний, 2005.
3. Путимцева Ю.С. Информатика. Диагностические работы в формате ЕГЭ 2011.- М. : МНЦМО, 2011