

Написать программу для построения графика функции

$$y = ax^3 + bx^2 + cx + d$$

с использованием графического интерфейса.

### ТЕКСТ УПРАВЛЯЮЩЕГО МОДУЛЯ ПРОЕКТА (ФАЙЛ GRAPHPLOT.CPP)

```
//-----  
#include <vcl.h>  
#pragma hdrstop  
USERES("GraphPlot.res");  
USEFORM("parabgr.cpp", Form1);  
//-----  
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)  
{  
    try  
    {  
        Application->Initialize();  
        Application->CreateForm(__classid(TForm1), &Form1);  
        Application->Run();  
    }  
    catch (Exception &exception)  
    {  
        Application->ShowException(&exception);  
    }  
    return 0;  
}  
//-----
```

### ТЕКСТ МОДУЛЯ ГЛАВНОЙ ФОРМЫ

#### *Файл parabgr.h*

```
//-----  
#ifndef parabgrH  
#define parabgrH  
//-----  
#include <Classes.hpp>  
#include <Controls.hpp>  
#include <StdCtrls.hpp>  
#include <Forms.hpp>  
#include "PERFGRAP.h"  
#include <ExtCtrls.hpp>  
//-----  
class TForm1 : public TForm  
{  
    __published:        // IDE-managed Components  
        TEdit *aEdit;  
        TEdit *bEdit;  
        TEdit *cEdit;  
        TEdit *dEdit;  
        TPaintBox *pb;  
        TEdit *leftEdit;  
        TEdit *rightEdit;  
        TLabel *Label1;  
        TLabel *Label2;  
        TLabel *Label3;  
};
```

```

        TLabel *Label4;
        TLabel *Label5;
        TLabel *Label6;
        TButton *RefreshButton;
        void __fastcall pbPaint(TObject *Sender);

        void __fastcall RefreshButtonClick(TObject *Sender);
private:    // User declarations
public:    // User declarations
        __fastcall TForm1(TComponent* Owner);
};
//-----
extern PACKAGE TForm1 *Form1;
//-----
#endif

```

### *Файл parabgr.cpp*

```

//-----

#include <vcl.h>
#pragma hdrstop

#include "parabgr.h"
//-----
#pragma package(smart_init)
#pragma link "PERFGRAP"
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
        : TForm(Owner)
{
}
float f(float x, float a, float b, float c, float d)
{
return (d + x * (c + x * (b + x * a)));
}
//-----
void __fastcall TForm1::pbPaint(TObject *Sender)
{
TPaintBox *pPB = (TPaintBox *)Sender;
POINT points[601];
pPB->Canvas->Brush->Color = clWhite;

pPB->Canvas->FillRect(Rect(0,0,600,400));

pPB->Canvas->Pen->Color = clBlack;
float a = StrToFloat(aEdit->Text);
float b = StrToFloat(bEdit->Text);
float c = StrToFloat(cEdit->Text);
float d = StrToFloat(dEdit->Text);
float xl = StrToFloat(leftEdit->Text);
float xr = StrToFloat(rightEdit->Text);
float h = (xr - xl) / 600;
float y[601];
y[0] = f(xl, a, b, c, d);
float min = y[0];
float max = min;
int k;
for (k = 1; k < 601; k++) {

```

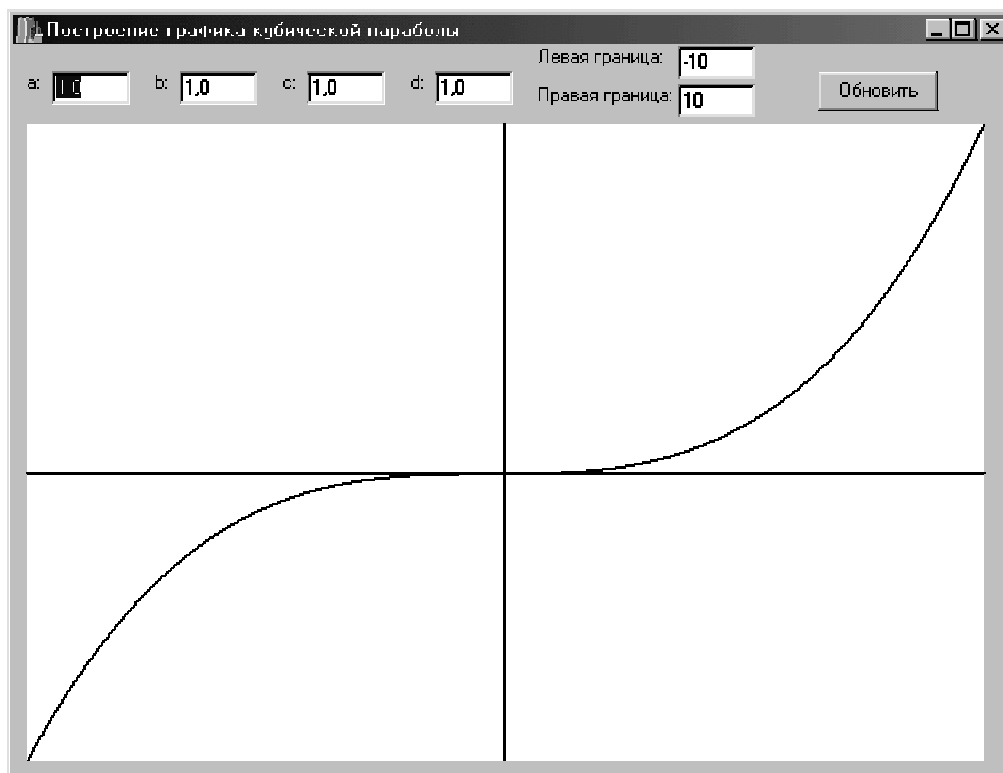
```

        y[k] = f(xl + k * h, a, b, c, d);
        if (y[k] > max) max = y[k];
        else if (y[k] < min) min = y[k];
    }
    float my = (max - min) / 400;
    pPB->Canvas->MoveTo((int) (-xl)/h, 0);
    pPB->Canvas->LineTo((int) (-xl)/h, 400);
    pPB->Canvas->MoveTo(0, (int) (max)/my);
    pPB->Canvas->LineTo(600, (int) max/my);
    for (k = 0; k < 601; k++) {
        points[k].x = k;
        points[k].y = (int) ((max - y[k]) / my);
    }
    pPB->Canvas->Polyline((TPoint*) points, 600);
}
//-----

void __fastcall TForm1::RefreshButtonClick(TObject *Sender)
{
    pbPaint(pb);
}
//-----

```

### ОКНО ПРОГРАММЫ ПОСЛЕ ЗАПУСКА



Программа позволяет задать в соответствующих полях коэффициенты параболы и границы для построения графика. После этого следует нажать кнопку «Обновить» вывода графика в окно приложения.