

Лабораторная работа №12 (3-й семестр)

Теоретические сведения

Ходить по воде и разрабатывать программы, следуя спецификации, очень просто... если они заморожены.

— Edward V Berard

Диаграмма (греч. *Διάγραμμα* (diagramma) — изображение, рисунок, чертёж) — графическое представление данных, позволяющее быстро оценить соотношение нескольких величин. Представляет собой геометрическое символическое изображение информации с применением различных приёмов техники визуализации. На практике чрезвычайно важным является построение диаграмм (графиков) различных функций, процессов и т. д.

Пример: разработать кроссплатформенную программу, которая позволяет строить график функции $y(x)=5*\sin(x)/x$

Файл *wxChart.h*

```
#ifndef WXCHART_H
#define WXCHART_H
#include <wx/wx.h>
#include <wx/panel.h>
#include <wx/dcclient.h>
#include <wx/msgdlg.h>
#include <vector>
class wxChart: public wxPanel
{
    DECLARE_DYNAMIC_CLASS(wxChart)
protected:
    DECLARE_EVENT_TABLE();
    int _x(double x);
    int _y(double y);
    void calcScale(double xm_min, double xm_max, double
ym_min, double ym_max);
    double m_a, m_b, m_c, m_d;
    std::vector<double> m_X;
    std::vector<double> m_Y;
    int m_width, m_height;
    double m_xMin, m_xMax, m_yMin, m_yMax;
public:
    wxChart();
    wxChart(wxWindow* parent, const long& id, const wxString&
label, const wxPoint& pos = wxDefaultPosition, const wxSize& size
= wxDefaultSize): wxPanel(parent, id, pos, size)
    {
        m_width = size.GetWidth();
        m_height = size.GetHeight();
    }
    void render(wxDC &dc);
    void setData(double* x, double* y, int size);
    void OnPaint(wxPaintEvent& event);
    void plot();
};
#endif // WXCHART_H
```

Файл *wxChart.cpp*

```
#include "wxChart.h"
#include <wx/dcclient.h>
#include <vector>
#include <algorithm>
IMPLEMENT_DYNAMIC_CLASS(wxChart, wxPanel);
wxChart::wxChart() { }
    int wxChart::_x(double x)
{
    return m_a*x + m_b;
}
    int wxChart::_y(double y)
{
    return m_c*y + m_d;
}
void wxChart::calcScale(double xm_max, double xm_min, double
ym_max, double ym_min)
{
    int xc_min = 10;
    int xc_max = m_width - 10;
    int yc_min = 10;
    int yc_max = m_height - 10;

    m_a = (xc_max - xc_min) / (xm_max - xm_min);
    m_b = xc_min - m_a * xm_min;
    m_c = (yc_min - yc_max) / (ym_max - ym_min);
    m_d = yc_min - m_c * ym_max;
}
void wxChart::OnPaint(wxPaintEvent& event)
{
    wxPaintDC dc(this);
    render(dc);
}
void wxChart::plot()
{
    Refresh();
}
void wxChart::setData(double* x, double* y, int size)
{
    m_X.clear();
    m_Y.clear();
    m_X.assign(x, x+size);
    m_Y.assign(y, y+size);
    m_xMin = wxRound(*std::min_element(m_X.begin(), m_X.end()));
    m_xMax = wxRound(*std::max_element(m_X.begin(), m_X.end()));
    m_yMin = wxRound(*std::min_element(m_Y.begin(), m_Y.end()));
    m_yMax = wxRound(*std::max_element(m_Y.begin(), m_Y.end()));
    calcScale(m_xMax, m_xMin, m_yMax, m_yMin);
}
void wxChart::render(wxDC &dc)
{
    dc.DrawRectangle(10, 10, m_width-20, m_height-20);
}
```

```

if (m_X.size() > 0)
{
    dc.SetPen( wxPen( *wxBLACK, 2, wxSOLID));
    dc.DrawLine( _x(m_xMin), _y(0), _x(m_xMax), _y(0));
    dc.DrawLine( _x(0), _y(m_yMin), _x(0), _y(m_yMax));
    dc.SetPen( wxPen( *wxLIGHT_GREY, 1, wxDOT));
    int num_steps = 10;
    double step_x = wxRound((m_xMax - m_xMin)/num_steps);
    double step_y = wxRound((m_yMax - m_yMin)/num_steps);
    for (int i = 1; i < num_steps; i++)
    {
        double xs = m_xMin+step_x*i;
        double ys = m_yMin+step_y*i;
        dc.DrawLine( _x(xs), _y(m_yMin), _x(xs), _y(m_yMax));
        dc.DrawLine( _x(m_xMin), _y(ys), _x(m_xMax), _y(ys));
        wxString x_coord, y_coord;
        x_coord << xs;
        y_coord << ys;
        int x_off = dc.GetTextExtent(x_coord).GetWidth() / 2;
        int y_off = dc.GetTextExtent(y_coord).GetHeight() / 2;
        dc.DrawText(x_coord, _x(xs)-x_off, _y(0));
        dc.DrawText(y_coord, _x(0), _y(ys)-y_off);
    }
    dc.SetPen(wxPen(*wxBLUE,2));
    for (int i=0; i<m_X.size()-1; i++)
    {
        int x0 = _x(m_X[i]);
        int x1 = _x(m_X[i+1]);
        int y0 = _y(m_Y[i]);
        int y1 = _y(m_Y[i+1]);
        dc.DrawLine(x0,y0,x1,y1);
    }
}
}

BEGIN_EVENT_TABLE(wxChart, wxPanel)
    EVT_PAINT(wxChart::OnPaint)
END_EVENT_TABLE();

```

Файл формы

```

// Объявить в классе формы
wxChart * myChart;
// -----
...
// Где-то в конструкторе формы
myChart = new wxChart(this, wxNewId(), wxString(_("test")),
wxDefaultPosition, wxSize(400,400));
// -----
...
void wxChart2Frame::OnButton1Click(wxCommandEvent& event)
{
    #define N 100
    double x[N];
    double y[N];

```

```
for (int i=0;i<N;i++)  
{  
    x[i] = i*0.1-1.0;  
    y[i] = 5.0*sin(x[i])/x[i];  
}  
myChart->setData(x,y,N);  
myChart->plot();  
myChart->Refresh();  
}
```

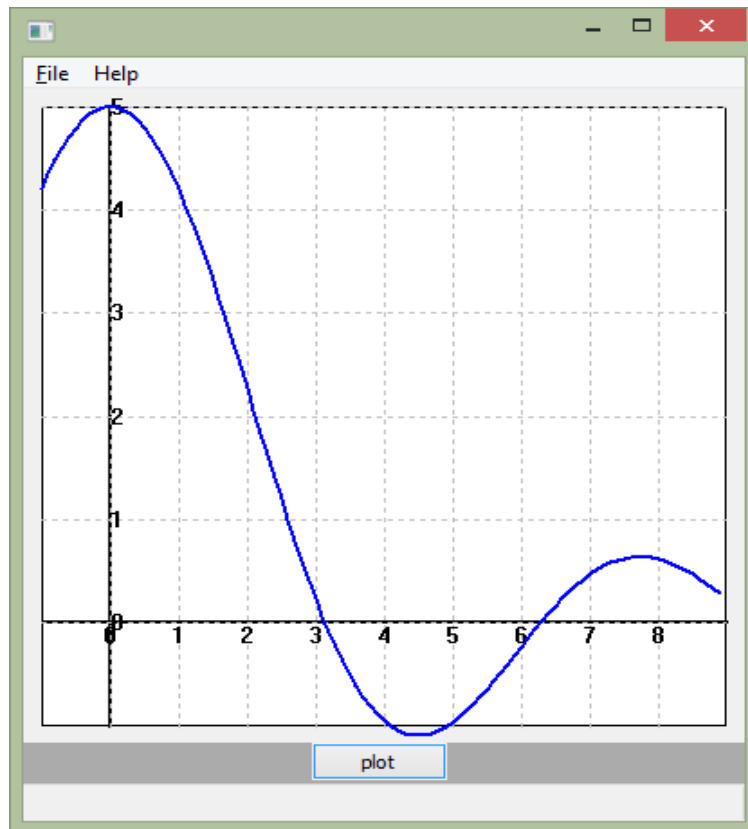


Рис. 2 - пример работы приложения