

# Лабораторная работа №9 (2-й семестр)

## Теоретические сведения

**Пример.** Написать программу, в которой определен класс, описывающий прямоугольник, который может отображаться на холсте компонента TPaintBox и имеет возможность изменять размер и угол наклона (рис. 1).

Файл: *TRotatedRectangle.h*

```
#ifndef TROTATEDRECTANGLE_H_INCLUDED
#define TROTATEDRECTANGLE_H_INCLUDED

#include <wx/dcclient.h>
const int N = 4;
class TRotatedRectangle
{
private:
    wxClientDC* paintBox;
    int fCenterX, fCenterY, fWidth, fHeight;
    double fAlpha;
    wxPoint fVertices[N];
    void CalculateVertices();
public:
    TRotatedRectangle(wxClientDC* , int centerX, int centerY,
int width, int height, double alpha);
    void Rotate(double dAlpha);
    void Draw();
    void ChangeSize(int width, int height);
};
#endif // TROTATEDRECTANGLE_H_INCLUDED
```

Файл: *TRotatedRectangle.cpp*

```
#include <math.h>
#include "TRotatedRectangle.h"

void TRotatedRectangle::CalculateVertices()
{
    fVertices[0].x = fWidth/2;
    fVertices[0].y = -fHeight/2;
    fVertices[1].x = -fWidth/2;
    fVertices[1].y = -fHeight/2;
    fVertices[2].x = -fWidth/2;
    fVertices[2].y = fHeight/2;
    fVertices[3].x = fWidth/2;
    fVertices[3].y = fHeight/2;
    for (int i = 0; i < N ; i++)
    {
        int x = fVertices[i].x;
        int y = fVertices[i].y;
        fVertices[i].x = x * cos(fAlpha) - y * sin(fAlpha) + fCenterX;
        fVertices[i].y = x * sin(fAlpha) + y * cos(fAlpha) + fCenterY;
    }
}
```

Измерять продуктивность программирования подсчетом строк кода — это так же, как оценивать постройку самолета по его весу.

— Bill Gates

```

        fVertices[i].y = x * sin(fAlpha) + y * cos(fAlpha) + fCenterY;
    }
}

void TRotatedRectangle::ChangeSize(int width, int height)
{
    fWidth = width;
    fHeight = height;
    CalculateVertices();
    paintBox->Clear();
    this->Draw();
}

TRotatedRectangle::TRotatedRectangle(wxClientDC* PaintBox, int
centerX, int centerY, int width, int height, double alpha)
{
    paintBox = PaintBox;
    fCenterX = centerX;
    fCenterY = centerY;
    fWidth = width;
    fHeight = height;
    fAlpha = alpha;
    CalculateVertices();
    paintBox->Clear();
}

void TRotatedRectangle::Draw()
{
    paintBox->SetBrush(wxBrush(*wxWHITE));
    paintBox->DrawPolygon(N, fVertices);
}

void TRotatedRectangle::Rotate(double dAlpha)
{
    fAlpha += dAlpha;
    CalculateVertices();
    paintBox->Clear();
}

```

Файл: *Unit1.cpp*

```

#include <wx/dcclient.h>
#include "TrotatedRectangle.h"
...
TRotatedRectangle *rect;
wxClientDC *dc;
wxplot2Frame::wxplot2Frame(wxWindow* parent, wxWindowID id)
{
    ...
    dc = new wxClientDC(Panell1);
    int w, h;
    dc->GetSize(&w, &h);
    rect = new TRotatedRectangle(dc, w/2, h/2, 200, 100, 0.0);
}

```

```

wxplot2Frame::~wxplot2Frame()
{
    delete rect;
    delete dc;
}

void wxplot2Frame::OnButton1Click(wxCommandEvent& event)
{
    rect->Draw();
}

void wxplot2Frame::OnButton2Click(wxCommandEvent& event)
{
    rect->Rotate(M_PI/10.0);
    rect->Draw();
}

void wxplot2Frame::OnButton3Click(wxCommandEvent& event)
{
    rect->Rotate(-M_PI/10.0);
    rect->Draw();
}

void wxplot2Frame::OnButton4Click(wxCommandEvent& event)
{
    long int w, h;
    TextCtrl1->GetValue().ToLong(&w);
    TextCtrl2->GetValue().ToLong(&h);
    rect->ChangeSize(w, h);
}

```

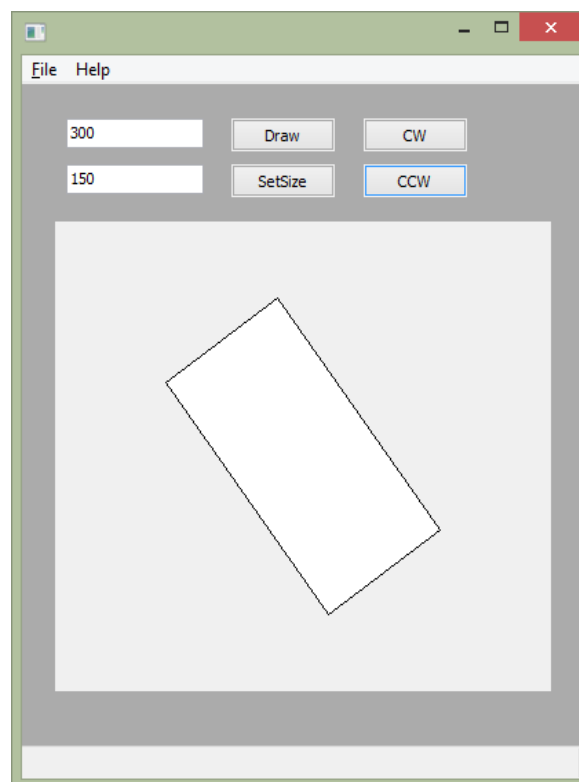


Рис. 1 — пример работы программы