```
// Autor: Krzysztof Zajączkowski
// www.algorytm.org

#include <windows.h>

#include <math.h>

class dPoint2D{
      double x;
      double y;
public:
       dPoint2D(double x,double y);

       dPoint2D operator -(dPoint2D p);
       dPoint2D operator +(dPoint2D p);
       dPoint2D operator *(double k);
       double operator *(dPoint2D p);
       double SickDistance2D(dPoint2D p);
       double Length();
       dPoint2D GetPointOnLine(dPoint2D &fP,dPoint2D &lP);

       double X();
       double Y();
       void X(double x);
       void Y(double y);
};

dPoint2D::dPoint2D(double x,double y){
    this->x = x;
    this->y = y;
}

double dPoint2D::X(){
     return x;
}
double dPoint2D::Y(){
     return y;
}
void dPoint2D::Y(double y){
     this->y = y;
}
void dPoint2D::X(double x){
     this->x = x;
}

dPoint2D dPoint2D::operator *(double k){
     dPoint2D temp(x * k,y * k);
     return temp;
```

```
}

double dPoint2D::operator *(dPoint2D p){
    return p.X() * x + p.Y() * y;
}

double dPoint2D::SickDistance2D(dPoint2D p){
      return pow(x - p.X(),2.0) + pow(y - p.Y(),2.0);
}

double dPoint2D::Length(){
    return sqrt(x * x + y * y);
}

dPoint2D dPoint2D::operator -(dPoint2D p){
     dPoint2D temp(this->x - p.X(),this->y - p.Y());
     return temp;
}

dPoint2D dPoint2D::operator +(dPoint2D p){
     dPoint2D temp(this->x + p.X(),this->y + p.Y());
     return temp;
}

dPoint2D dPoint2D::GetPointOnLine(dPoint2D &fP,dPoint2D &lP){
     //double u = ((x - fP.X()) * (lP.X() - fP.X()) + (y - fP.Y())
        * (lP.Y() - fP.Y())) / (lP.SickDistance2D(fP));
    double u = (((*this) - fP) * (lP - fP))/ ((fP - lP) * (fP -
       lP));
    return fP + (lP - fP) * u;
}

LRESULT CALLBACK WndProc(HWND hWnd,UINT msg,WPARAM wParam,LPARAM
   lParam){
    static dPoint2D v1(20,300); // Punkty linii L1
    static dPoint2D v2(300,20);
    static dPoint2D v3(500,500); // Punkty linii L2
    static dPoint2D v4(100,100);
    static dPoint2D v5(0,0);    // Punkt przecięcia
    static POINT MousePos;

    static dPoint2D *pt = NULL;
    switch(msg)
    {
        case WM_CREATE:
            {
                dPoint2D v6(0,0);
                v6 = v1 + v4 - v3;
```

```
                dPoint2D v7(0,0);
                dPoint2D v8(0,0);

                v7 = v6.GetPointOnLine(v1,v2);
                v8 = v4.GetPointOnLine(v1,v2);

                double v4v5_length = (v6 - v1).Length() / (v6 -
                   v7).Length() * (v8 - v4).Length();

                dPoint2D v9(0,0);

                v9 = v8.GetPointOnLine(v3,v4);

                v5 = (v9 - v4) * (v4v5_length / (v4 - v9).Length());

                double l = v5.Length();

                v5 = v5 + v4;
            }
            break;
        case WM_PAINT:
        {
            PAINTSTRUCT ps;
            HDC hdc = BeginPaint(hWnd,&ps);
            HPEN hPen = CreatePen(PS_SOLID,2,RGB(0,0,255));

            SelectObject(hdc,hPen);

            MoveToEx(hdc,(int)v1.X(),(int)v1.Y(),NULL);
            LineTo(hdc,(int)v2.X(),(int)v2.Y());

            MoveToEx(hdc,(int)v3.X(),(int)v3.Y(),NULL);
            LineTo(hdc,(int)v4.X(),(int)v4.Y());

            Ellipse(hdc,(int)(v5.X() - 5),(int)(v5.Y() -
               5),(int)(v5.X() + 5),(int)(v5.Y() + 5));

            DeleteObject(hPen);
            EndPaint(hWnd,&ps);
        }
        break;
        case WM_LBUTTONDOWN:
            {
                if( 5 > abs(MousePos.x - v1.X()) && 5 >
                   abs(MousePos.y - v1.Y())){
                    pt = &v1;
```

```
            }else if( 5 > abs(MousePos.x - v2.X()) && 5 >
               abs(MousePos.y - v2.Y())){
                pt = &v2;
            }else if( 5 > abs(MousePos.x - v3.X()) && 5 >
               abs(MousePos.y - v3.Y())){
                pt = &v3;
            }else if( 5 > abs(MousePos.x - v4.X()) && 5 >
               abs(MousePos.y - v4.Y())){
                pt = &v4;
            }
        }
        break;
    case WM_LBUTTONUP:
        {
            pt = NULL;
        }
        break;
    case WM_RBUTTONDOWN:
        {

        }
        break;
    case WM_MOUSEMOVE:
        {
            MousePos.x = LOWORD(lParam);
            MousePos.y = HIWORD(lParam);
            if(pt){
                //POINT pt;

                pt->X(MousePos.x);
                pt->Y(MousePos.y);

                dPoint2D v6(0,0);
                v6 = v1 + v4 - v3;

                dPoint2D v7(0,0);
                dPoint2D v8(0,0);

                v7 = v6.GetPointOnLine(v1,v2);
                v8 = v4.GetPointOnLine(v1,v2);

                double v4v5_length = (v6 - v1).Length() / (v6 -
                   v7).Length() * (v8 - v4).Length();

                dPoint2D v9(0,0);

                v9 = v8.GetPointOnLine(v3,v4);
```

```
                    v5 = (v9 - v4) * (v4v5_length / (v4 -
                       v9).Length());

                    double l = v5.Length();

                    v5 = v5 + v4;

                    InvalidateRect(hWnd,NULL,true);
                }
            }
            break;
        case WM_DESTROY:
        {
            PostQuitMessage(0);
        }
        break;
    }
    return DefWindowProc(hWnd,msg,wParam,lParam);
}

int WINAPI WinMain(HINSTANCE hInstance,HINSTANCE ,LPSTR ,int ){
    WNDCLASS wnd;

    wnd.cbClsExtra = NULL;
    wnd.cbWndExtra = NULL;
    wnd.hbrBackground = CreateSolidBrush(RGB(0,0,0));
    wnd.hCursor = LoadCursor(NULL,IDC_ARROW);
    wnd.hIcon = LoadIcon(NULL, IDI_APPLICATION);
    wnd.hInstance = hInstance;
    wnd.lpfnWndProc = WndProc;
    wnd.lpszClassName = "PunktPrzecieciaProstych";
    wnd.lpszMenuName = NULL;
    wnd.style = CS_VREDRAW|CS_HREDRAW;

    if(!RegisterClass(&wnd)){
        MessageBox(NULL,"Coś nie tak","Informacja",MB_OK);
        return 0;
    }

    HWND hwnd = CreateWindow("PunktPrzecieciaProstych","Autor:
       Krzysztof Zajączkowski, kontakt:
       malyszkz@wp.pl",WS_OVERLAPPEDWINDOW,0,0,800,600,NULL,NULL,hI
       nstance,NULL);

    ShowWindow(hwnd, SW_SHOWNORMAL);
    UpdateWindow(hwnd);

    MSG msg;
```

```
    while(GetMessage(&msg,NULL,0,0)){
        DispatchMessage(&msg);
        TranslateMessage(&msg);
    }

    return 0;
}
```