

Jakub Sosin	Grafika komputerowa	Grupa 06
Problem przynależności punktu do wielokątów	15.05.2020r.	ROK II IS

1.Cel ćwiczenia:

Zapoznanie z algorytmami związanymi z problemem sprawdzania przynależności punktu do wielokąta.

2.Wykonanie ćwiczenia:

Algorytmem sprawdzający, po której stronie prostej leży punkt:

```

0 references
private int Strony_odcinek(PointF a, PointF b, PointF c)
{
    float matrixWyznacznik = a.X*b.Y+1+a.Y*1*c.X+1*b.X*c.Y-1*b.Y*c.X- a.X*1*c.Y-a.Y*b.X*1;

    if (matrixWyznacznik > 0) return 1;
    if (matrixWyznacznik < 0) return -1;
    return 0;
}

```

Metoda ta oblicza wyznacznik z macierzy (matrixWyznacznik). To czy punkt znajdzie się po lewej, czy prawej stronie zależy od znaku obliczonego wyznacznika macierzy.

```

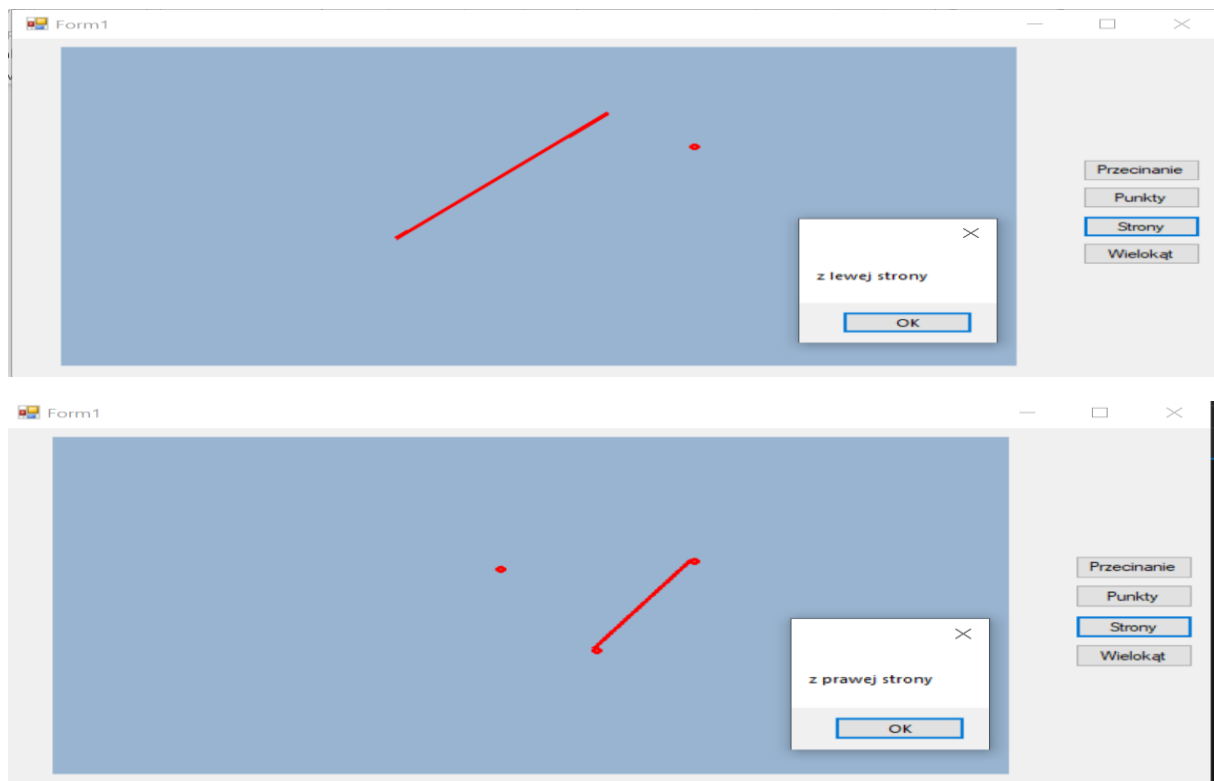
private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    switch(zadanie)
    {
        case 1:
            if (counter == 0)
                punkty = new PointF[3];

            if(counter < 3)
            {
                punkty[counter].X = e.X;
                punkty[counter].Y = e.Y;
                counter++;
                g.DrawEllipse(pen1, e.X, e.Y, 4, 4);
            }
            if(counter == 3)
            {
                g.DrawLine(pen1, punkty[0], punkty[1]);
                int wyn = Strony_odcinek(punkty[0], punkty[1], punkty[2]);
                if (wyn == -1) MessageBox.Show("z lewej strony");
                if (wyn == 1) MessageBox.Show("z prawej strony ");
                if (wyn == 0) MessageBox.Show("punkt na odcinku");
                counter = 0;
            }
            break;
    }
}

```

W tym miejscu tworzę tablicę 3-elementową współrzędnych punktów. Aby funkcja zadziałała trzeba wybrać na PictureBoxie dokładnie trzy punkt(counter==3)

Działanie:



Algorytmem sprawdzający, czy dwa dane punkty leżą po tej samej stronie prostej:

```
private int check_teSameStrony(PointF a, PointF b, PointF c, PointF d)
{
    int wyn1 = Strony_odcinek(a, b, c);
    int wyn2 = Strony_odcinek(a, b, d);

    if (Math.Sign(wyn1) == Math.Sign(wyn2)) return 1;
    else return 0;
}
```

Ten fragment kodu ma za zadanie wywołać dwa razy funkcję `Strony_odcinek`, jeśli funkcje zwracają ten sam wynik otrzymuje wartość `true`, w przeciwnym wypadku wartość `false`.

```

case 2:
if (counter == 0)
    punkty = new PointF[4];

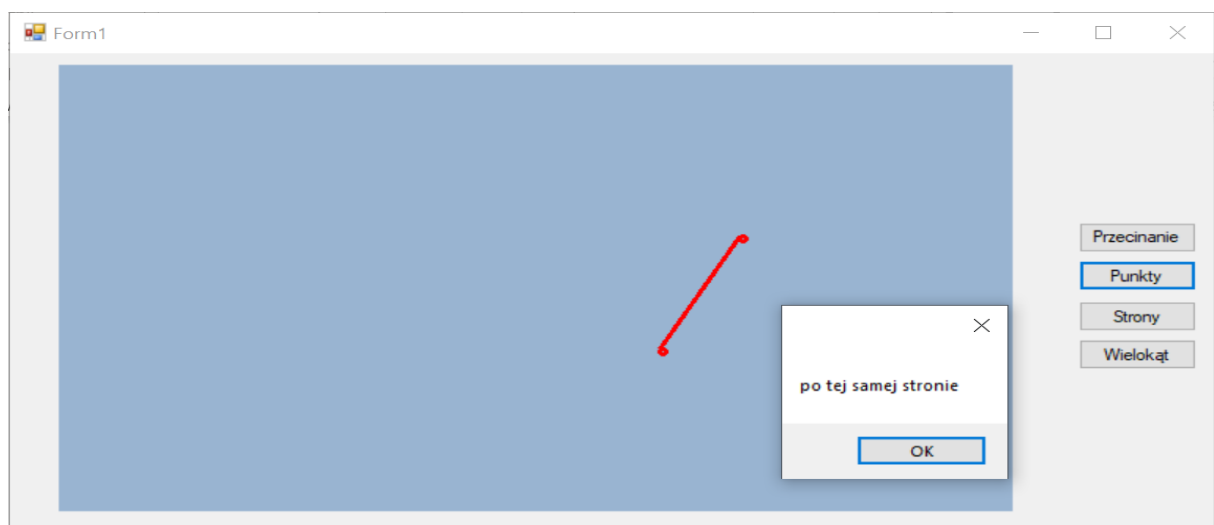
if (counter < 4)
{
    punkty[counter].X = e.X;
    punkty[counter].Y = e.Y;
    counter++;
    g.DrawEllipse(pen1, e.X,e.Y,4,4);
}
else
{
    g.DrawLine(pen1, punkty[0], punkty[1]);

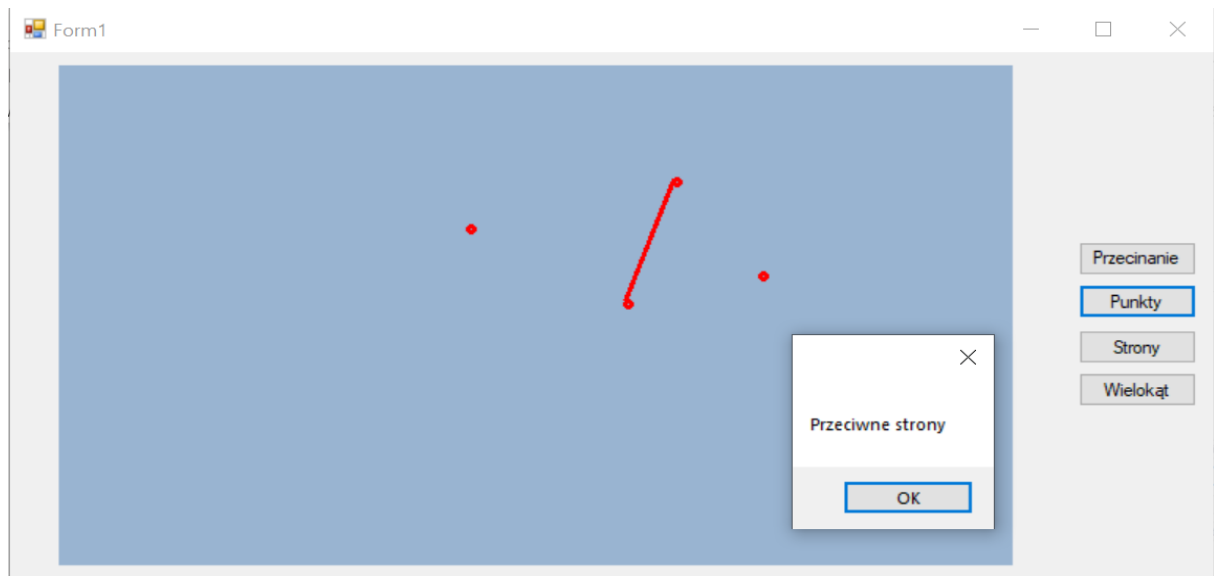
    if (CzyTeSameStrony(punkty[0], punkty[1], punkty[2], punkty[3]) == 1)
        MessageBox.Show("po tej samej stronie");
    else
        MessageBox.Show("Przeciwne strony");
}
break;

```

W tym wypadku tworzę tablice współrzędnych 4 punktów-mam sprawdzić po której stronie odcinka leżą dwa punkty. W zależności od wyniku zwróconej funkcji CzyTeSameStrony messagebox wyświetla komunikat „po tej samej stronie” bądź „przeciwne strony”.

Działanie





Algorytm sprawdzający, czy dwa odcinki się przecinają

```
2 references
private int przecinanie_odcinkow(PointF a, PointF b, PointF c, PointF d)
{
    int warunek1 = check_teSameStrony(a, b, c, d);

    int warunek2 = check_teSameStrony(c,d,a,b);

    if (warunek1 == 0 && warunek2 == 0)
        return 1;
    else
        return 0;
}
```

Wywołuje funkcję check_teSameStrony, przy pierwszym wywołaniu sprawdzam czy punkty c i d leżą po tej samej stronie odcinka ab, przy drugim po której stronie leżą punktu a i b.

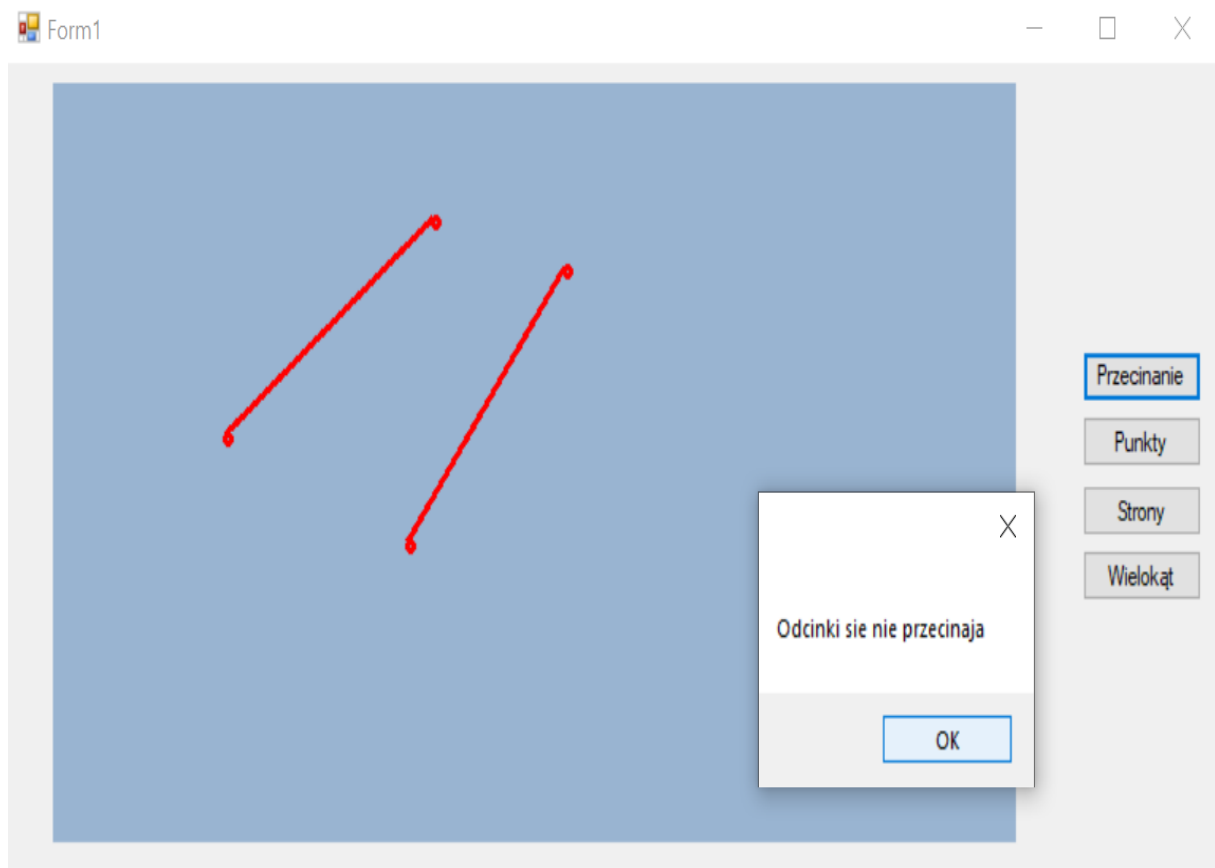
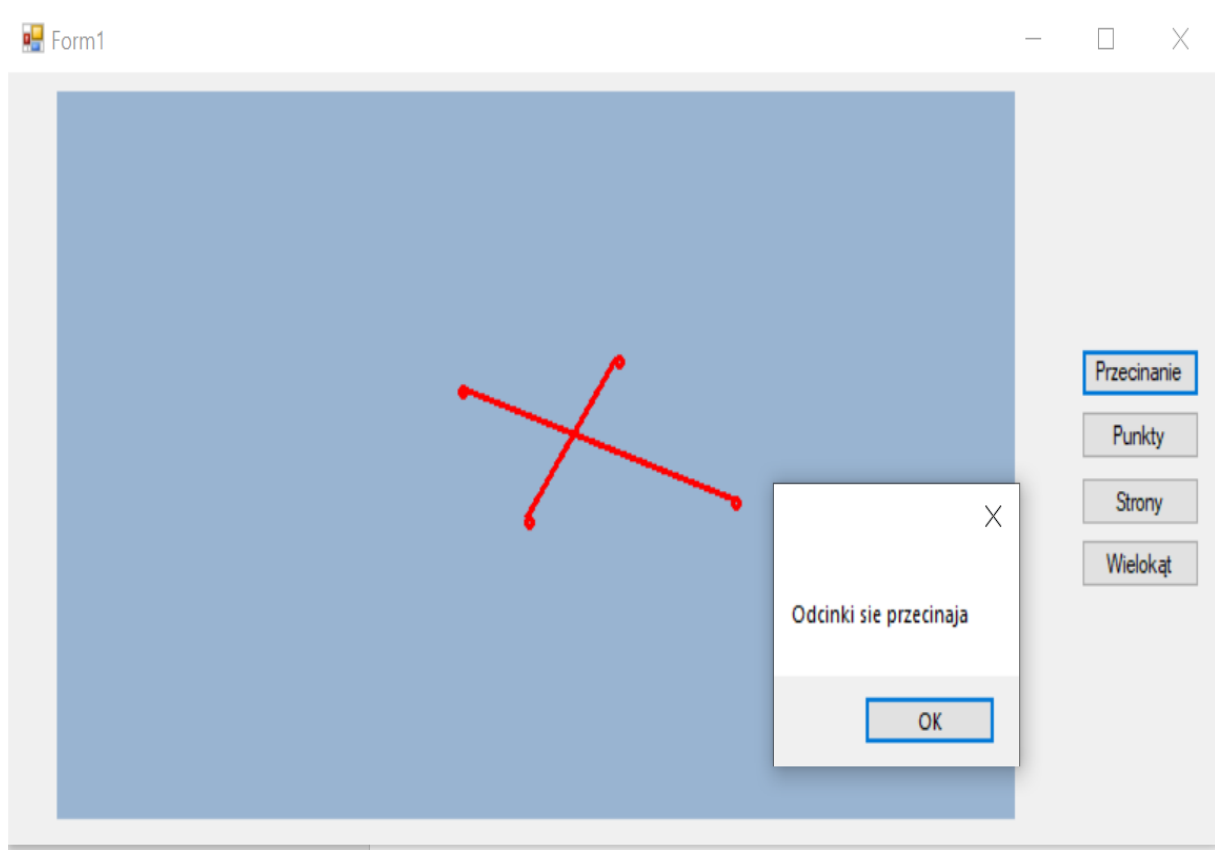
```
case 3:
    if (counter == 0)
        punkty = new PointF[4];
    if (counter < 4)
    {
        punkty[counter].X = e.X;
        punkty[counter].Y = e.Y;
        counter++;
        g.DrawEllipse(pen1, e.X, e.Y, 4, 4);
    }
    else
    {
        g.DrawLine(pen1, punkty[0], punkty[1]);

        g.DrawLine(pen1, punkty[2], punkty[3]);

        if (przecinanie_odcinkow(punkty[0], punkty[1], punkty[2], punkty[3]) == 1)
            MessageBox.Show("Odcinki sie przecinaja");
        else
            MessageBox.Show("Odcinki sie nie przecinaja");
    }
    break;
```

Case 3-wywołuje funkcję przecinanie_odcinkow i w zależności od wyniku tej funkcji wyświetlam odpowiedni komunikat

Działanie

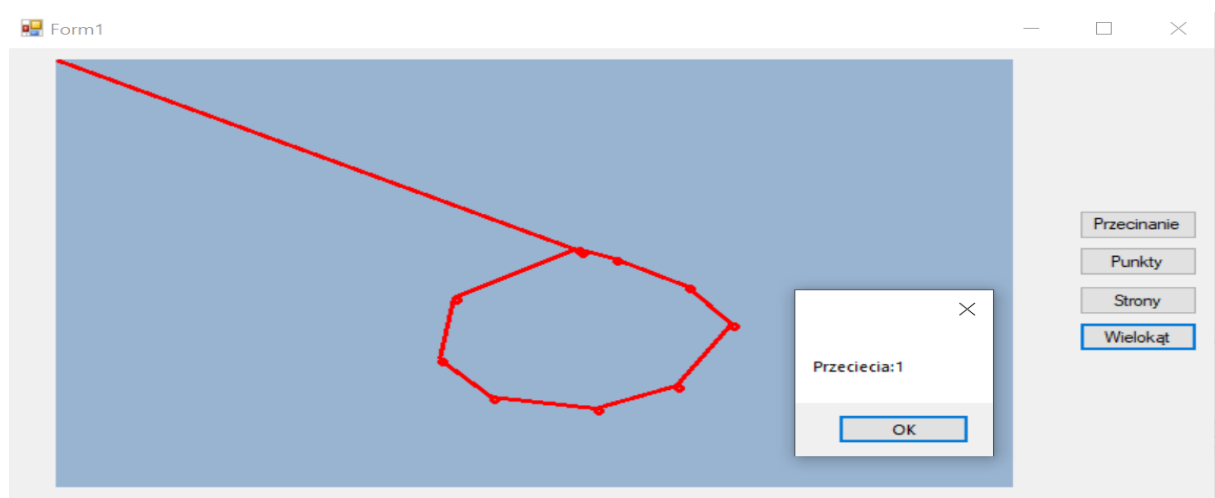


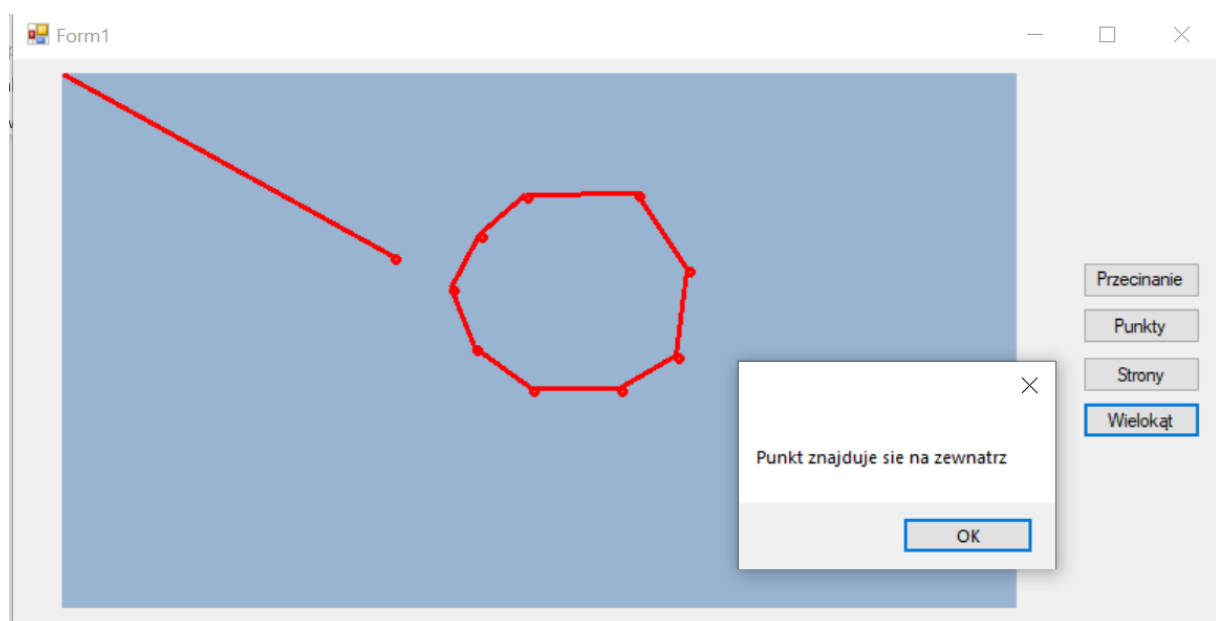
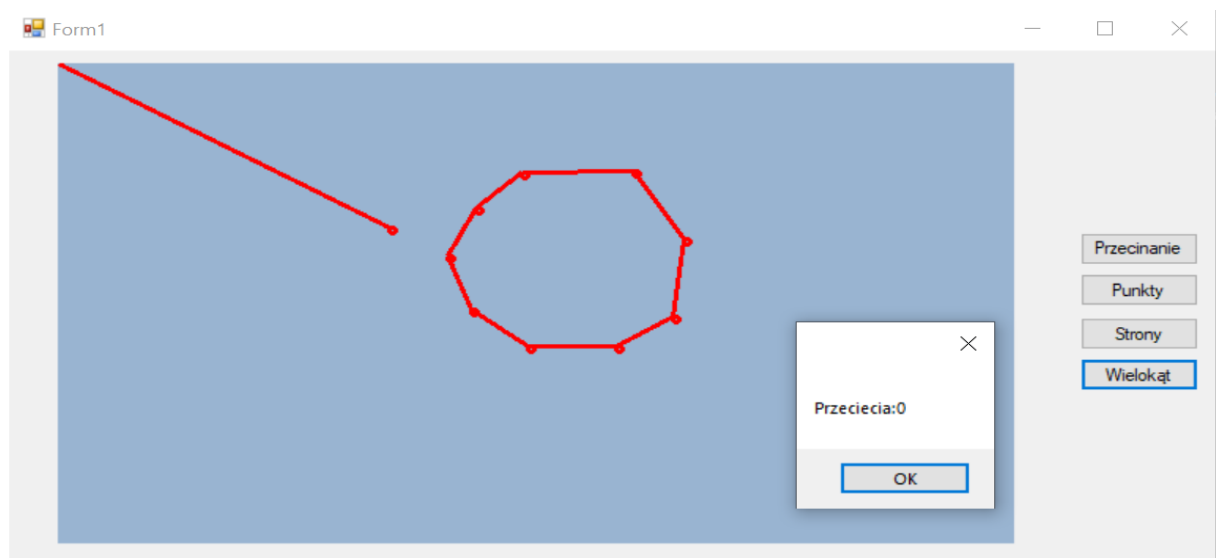
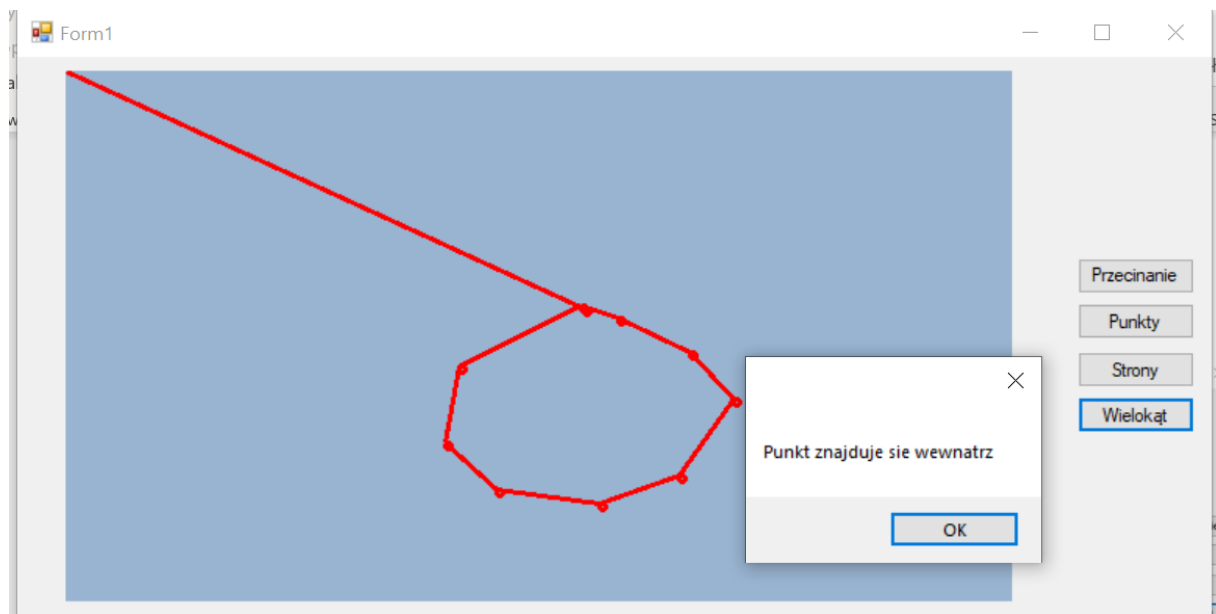
Problem przynależności punktu do wielokąta:

```
case 4:
    if (counter == 0)
        punkty = new PointF[10];
    if (counter < 10)
    {
        punkty[counter].X = e.X;
        punkty[counter].Y = e.Y;
        counter++;
        g.DrawEllipse(pen1, e.X, e.Y, 4, 4);
    }
    else
    {
        for (int i = 0; i < 8; i++)
            g.DrawLine(pen1, punkty[i], punkty[i + 1]);
        g.DrawLine(pen1, punkty[0], punkty[8]);
        PointF test = new PointF(0, 0);
        g.DrawLine(pen1, test, punkty[9]);
        int przeciecia = 0;
        for(int i = 0; i < 8; i++)
        {
            if (przecinanie_odcinkow(punkty[i], punkty[i + 1], test, punkty[9]) == 1)
            {
                przeciecia++;
            }
        }
        MessageBox.Show("Przeciecia:"+przeciecia);
        if (przeciecia%2 == 0)
            MessageBox.Show("Punkt znajduje sie na zewnatrz");
        else
            MessageBox.Show("Punkt znajduje sie wewnatrz");
    }
    break;
}
```

Tutaj inaczej niż poprzednio, tworzę tylko case-a, a w nim tablice w której znajdują się punkty wielokąta o 10 wierzchołkach. Tworzę również punkt (jest nim początek układu współrzędnych), za pomocą którego sprawdzam czy połączy się z ostatnim punktem tego 10-kąta. Funkcja przecinanie_odcinkow wykonuje się w pętli i zlicza ilość przecięć. W przypadku gdy ilość przecięć będzie parzysta-odcinek leży na zewnątrz wielokąta, w przeciwnym wewnątrz.

Działanie





3.Kod źródłowy:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        private System.Drawing.Graphics g; private System.Drawing.Pen pen1 = new
System.Drawing.Pen(Color.Red, 3);
        public Form1()
        {
            InitializeComponent();
            g = pictureBox1.CreateGraphics();
        }
        private int counter = 0;
        private int zadanie = 0;
        private PointF[] punkty;

        private int Strony_odcinek(PointF a, PointF b, PointF c)
        {
            float matrixWyznaczik = a.X*b.Y*1+a.Y*1*c.X+1*b.X*c.Y-1*b.Y*c.X-
a.X*1*c.Y-a.Y*b.X*1;

            if (matrixWyznaczik > 0) return 1;
            if (matrixWyznaczik < 0) return -1;
            return 0;
        }

        private int check_teSameStrony(PointF a, PointF b, PointF c, PointF d)
        {
            int wyn1 = Strony_odcinek(a, b, c);
            int wyn2 = Strony_odcinek(a, b, d);

            if (Math.Sign(wyn1) == Math.Sign(wyn2)) return 1;
            else return 0;
        }

        private int przecinanie_odcinkow(PointF a, PointF b, PointF c,PointF d)
        {
            int warunek1 = check_teSameStrony(a, b, c, d);

            int warunek2 = check_teSameStrony(c,d,a,b);

            if (warunek1 == 0 && warunek2 == 0)
                return 1;
            else
                return 0;
        }
    }
}
```



```

private void pictureBox1_Click(object sender, EventArgs e)
{

}

private void button1_Click(object sender, EventArgs e)
{
    pictureBox1.Refresh();
    zadanie = 1;

}

private void button5_Click(object sender, EventArgs e)
{

}

private void button7_Click(object sender, EventArgs e)
{

}

private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    switch(zadanie)
    {
        case 1:
            if (counter == 0)
                punkty = new PointF[3];

            if(counter < 3)
            {
                punkty[counter].X = e.X;
                punkty[counter].Y = e.Y;
                counter++;
                g.DrawEllipse(pen1, e.X, e.Y, 4, 4);
            }
            if(counter == 3)
            {
                g.DrawLine(pen1, punkty[0], punkty[1]);
                int wyn = Strony_odcinek(punkty[0], punkty[1], punkty[2]);
                if (wyn == -1) MessageBox.Show("z lewej strony");
                if (wyn == 1) MessageBox.Show("z prawej strony ");
                if (wyn == 0) MessageBox.Show("punkt na odcinku");
                counter = 0;
            }
            break;

        case 2:
            if (counter == 0)
                punkty = new PointF[4];

            if (counter < 4)
            {
                punkty[counter].X = e.X;
                punkty[counter].Y = e.Y;
                counter++;
                g.DrawEllipse(pen1, e.X,e.Y,4,4);
            }
            else
            {
                g.DrawLine(pen1, punkty[0], punkty[1]);
            }
        }
    }
}

```

```

        if (check_teSameStrony(punkty[0], punkty[1], punkty[2],
punkty[3]) == 1)
            MessageBox.Show("po tej samej stronie");
        else
            MessageBox.Show("Przeciwne strony");
    }
    break;
case 3:
    if (counter == 0)
        punkty = new PointF[4];
    if (counter < 4)
    {
        punkty[counter].X = e.X;
        punkty[counter].Y = e.Y;
        counter++;
        g.DrawEllipse(pen1, e.X, e.Y, 4, 4);
    }
    else
    {
        g.DrawLine(pen1, punkty[0], punkty[1]);

        g.DrawLine(pen1, punkty[2], punkty[3]);

        if (przecinanie_odcinkow(punkty[0], punkty[1], punkty[2],
punkty[3]) == 1)
            MessageBox.Show("Odcinki sie przecinaja");
        else
            MessageBox.Show("Odcinki sie nie przecinaja");
    }
    break;
case 4:
    if (counter == 0)
        punkty = new PointF[10];
    if (counter < 10)
    {
        punkty[counter].X = e.X;
        punkty[counter].Y = e.Y;
        counter++;
        g.DrawEllipse(pen1, e.X, e.Y, 4, 4);
    }
    else
    {
        for (int i = 0; i < 8; i++)
            g.DrawLine(pen1, punkty[i], punkty[i + 1]);
        g.DrawLine(pen1, punkty[0], punkty[8]);
        PointF test = new PointF(0, 0);
        g.DrawLine(pen1, test, punkty[9]);
        int przeciecia = 0;
        for(int i = 0; i < 8; i++)
        {
            if (przecinanie_odcinkow(punkty[i], punkty[i + 1], test,
punkty[9]) == 1)
            {
                przeciecia++;
            }
        }
        MessageBox.Show("Przeciecia:" + przeciecia);
        if (przeciecia % 2 == 0)
            MessageBox.Show("Punkt znajduje sie na zewnatrz");
    }
}

```

```

        else
            MessageBox.Show("Punkt znajduje sie wewnatrz");
    }
    break;
}
}

private void pictureBox1_Resize(object sender, EventArgs e)
{
    g = pictureBox1.CreateGraphics();
}

private void button2_Click(object sender, EventArgs e)
{
    pictureBox1.Refresh();
    zadanie = 2;
    counter = 0;
}

private void button3_Click(object sender, EventArgs e)
{
    pictureBox1.Refresh();
    zadanie = 3;
    counter = 0;
}

private void button4_Click(object sender, EventArgs e)
{
    pictureBox1.Refresh();
    counter = 0;
    zadanie = 4;
}

private void Form1_Load(object sender, EventArgs e)
{
}

}
}

```