

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Draw.src.Model

{

class ExamFigure4 : Shape

{

#region Constructor

public ExamFigure4(RectangleF rect) : base(rect)

{

}

public ExamFigure4(RectangleShape rectangle) : base(rectangle)

{

}

#endregion

/// <summary>

/// Проверка за принадлежност на точка point към правоъгълника.

/// В случая на правоъгълник този метод може да не бъде пренаписван, защото

/// Реализацията съвпада с тази на абстрактния клас Shape, който проверява

/// дали точката е в обхващащия правоъгълник на елемента (а той съвпада с

/// елемента в този случай).

/// </summary>

public override bool Contains(PointF point)

{

if (base.Contains(point))

// Проверка дали е в обекта само, ако точката е в обхващащия правоъгълник.

// В случая на правоъгълник - директно връщаме true

return true;

else

// Ако не е в обхващащия правоъгълник, то неможе да е в обекта и => false

return false;

}

/// <summary>

/// Частта, визуализираща конкретния примитив.

/// </summary>

public override void DrawSelf(Graphics grfx)

{

base.DrawSelf(grfx);

base.RotateShape(grfx);

grfx.FillRectangle(new SolidBrush(Color.FromArgb(Opacity, FillColor)), Rectangle.X, Rectangle.Y, Rectangle.Width, Rectangle.Height);

grfx.DrawRectangle(new Pen(BorderColor, BorderWidth), Rectangle.X, Rectangle.Y, Rectangle.Width, Rectangle.Height);

//From top left to bottom right

Point p1 = new Point((int)Rectangle.X, (int)Rectangle.Y);

Point p2 = new Point((int)(Rectangle.X + Rectangle.Width), (int)(Rectangle.Y + Rectangle.Height));

grfx.DrawLine(new Pen(BorderColor, BorderWidth), p1, p2);

//From bottom left to top right

Point p3 = new Point((int)(Rectangle.X), (int)(Rectangle.Y + Rectangle.Height));

Point p4 = new Point((int)(Rectangle.X + Rectangle.Width), (int)Rectangle.Y);

grfx.DrawLine(new Pen(BorderColor, BorderWidth), p3, p4);

grfx.ResetTransform();

}

}

}

—-----------------------------------------DIALOG—---------------------------------------------------------

public void AddRandomExamFigure4()

{

Random rnd = new Random();

int x = rnd.Next(100, 1000);

int y = rnd.Next(100, 600);

ExamFigure4 examFigure4 = new ExamFigure4(new Rectangle(x, y, 200, 100));

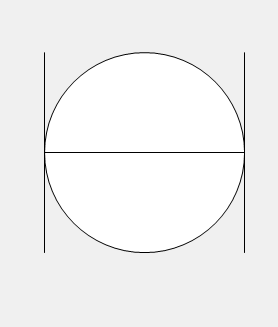
examFigure4.FillColor = Color.White;

examFigure4.BorderColor = Color.Black;

examFigure4.Opacity = 255;

ShapeList.Add(examFigure4);

}



using System;

using System.Collections.Generic;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Draw.src.Model

{

class ExamFigure1 : Shape

{

#region Constructor

public ExamFigure1(RectangleF rect) : base(rect)

{

}

public ExamFigure1(RectangleShape rectangle) : base(rectangle)

{

}

#endregion

/// <summary>

/// Проверка за принадлежност на точка point към правоъгълника.

/// В случая на правоъгълник този метод може да не бъде пренаписван, защото

/// Реализацията съвпада с тази на абстрактния клас Shape, който проверява

/// дали точката е в обхващащия правоъгълник на елемента (а той съвпада с

/// елемента в този случай).

/// </summary>

public override bool Contains(PointF point)

{

if (base.Contains(point))

// Проверка дали е в обекта само, ако точката е в обхващащия правоъгълник.

// В случая на правоъгълник - директно връщаме true

return true;

else

// Ако не е в обхващащия правоъгълник, то неможе да е в обекта и => false

return false;

}

/// <summary>

/// Частта, визуализираща конкретния примитив.

/// </summary>

public override void DrawSelf(Graphics grfx)

{

base.DrawSelf(grfx);

base.RotateShape(grfx);

grfx.FillEllipse(new SolidBrush(Color.FromArgb(Opacity, FillColor)), Rectangle.X, Rectangle.Y, Rectangle.Width, Rectangle.Height);

grfx.DrawEllipse(new Pen(BorderColor, BorderWidth), Rectangle.X, Rectangle.Y, Rectangle.Width, Rectangle.Height);

//Left line

Point p1 = new Point((int)Rectangle.X, (int)Rectangle.Y);

Point p2 = new Point((int)(Rectangle.X), (int)(Rectangle.Y + Rectangle.Height));

grfx.DrawLine(new Pen(BorderColor, BorderWidth), p1, p2);

//Right line

Point p3 = new Point((int)(Rectangle.X + Rectangle.Width), (int)Rectangle.Y);

Point p4 = new Point((int)(Rectangle.X + Rectangle.Width), (int)(Rectangle.Y + Rectangle.Height));

grfx.DrawLine(new Pen(BorderColor, BorderWidth), p3, p4);

//Middle line

Point p5 = new Point((int)((Rectangle.X + Rectangle.X) / 2), (int)((Rectangle.Y + (Rectangle.Y + Rectangle.Height)) / 2));

Point p6 = new Point((int)((Rectangle.X + Rectangle.Width + Rectangle.X + Rectangle.Width) / 2), (int)((Rectangle.Y + (Rectangle.Y + Rectangle.Height)) / 2));

grfx.DrawLine(new Pen(BorderColor, BorderWidth), p5, p6);

grfx.ResetTransform();

}

}

}

—-----------------------------------------DIALOG—---------------------------------------------------------

public void AddRandomExamFigure1()

{

Random rnd = new Random();

int x = rnd.Next(100, 1000);

int y = rnd.Next(100, 600);

ExamFigure1 examFigure1 = new ExamFigure1(new Rectangle(x, y, 200, 200));

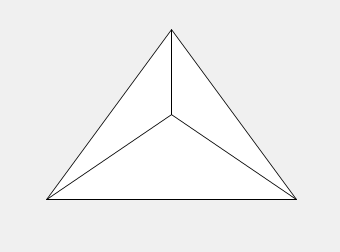
examFigure1.FillColor = Color.White;

examFigure1.BorderColor = Color.Black;

examFigure1.Opacity = 255;

ShapeList.Add(examFigure1);

}



using System;

using System.Collections.Generic;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Draw.src.Model

{

class ExamFigure0 : Shape

{

#region Constructor

public ExamFigure0(RectangleF tri) : base(tri)

{

}

public ExamFigure0(TriangleShape triangle) : base(triangle)

{

}

#endregion

/// <summary>

/// Проверка за принадлежност на точка point към правоъгълника.

/// В случая на правоъгълник този метод може да не бъде пренаписван, защото

/// Реализацията съвпада с тази на абстрактния клас Shape, който проверява

/// дали точката е в обхващащия правоъгълник на елемента (а той съвпада с

/// елемента в този случай).

/// </summary>

public override bool Contains(PointF point)

{

if (base.Contains(point))

// Проверка дали е в обекта само, ако точката е в обхващащия правоъгълник.

// В случая на правоъгълник - директно връщаме true

return true;

else

// Ако не е в обхващащия правоъгълник, то неможе да е в обекта и => false

return false;

}

/// <summary>

/// Частта, визуализираща конкретния примитив.

/// </summary>

public override void DrawSelf(Graphics grfx)

{

base.DrawSelf(grfx);

base.RotateShape(grfx);

Point[] p = { new Point((int)Rectangle.X + ((int)Rectangle.Width / 2), (int)Rectangle.Y),

new Point((int)Rectangle.X, (int)(Rectangle.Y + Rectangle.Height)),

new Point((int)(Rectangle.X + Rectangle.Width), (int)(Rectangle.Y + Rectangle.Height)) };

grfx.FillPolygon(new SolidBrush(Color.FromArgb(Opacity, FillColor)), p);

grfx.DrawPolygon(new Pen(BorderColor, BorderWidth), p);

//Center point

Point center = new Point((int)(Rectangle.X + Rectangle.Width / 2), (int)(Rectangle.Y + Rectangle.Height / 2));

//From top to center

Point p1 = new Point((int)Rectangle.X + ((int)Rectangle.Width / 2), (int)Rectangle.Y);

grfx.DrawLine(new Pen(BorderColor, BorderWidth), p1, center);

//From left to center

Point p2 = new Point((int)Rectangle.X, (int)(Rectangle.Y + Rectangle.Height));

grfx.DrawLine(new Pen(BorderColor, BorderWidth), p2, center);

//From right to center

Point p3 = new Point((int)(Rectangle.X + Rectangle.Width), (int)(Rectangle.Y + Rectangle.Height));

grfx.DrawLine(new Pen(BorderColor, BorderWidth), p3, center);

grfx.ResetTransform();

}

}

}

—-----------------------------------------DIALOG—---------------------------------------------------------

public void AddRandomExamFigure0()

{

Random rnd = new Random();

int x = rnd.Next(100, 1000);

int y = rnd.Next(100, 600);

ExamFigure0 examFigure0 = new ExamFigure0(new Rectangle(x, y, 250, 170));

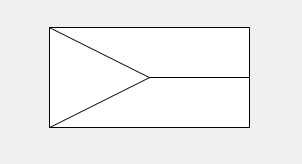
examFigure0.FillColor = Color.White;

examFigure0.BorderColor = Color.Black;

examFigure0.Opacity = 255;

ShapeList.Add(examFigure0);

}



using System;

using System.Collections.Generic;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Draw.src.Model

{

class ExamFigure2 : Shape

{

#region Constructor

public ExamFigure2(RectangleF rect) : base(rect)

{

}

public ExamFigure2(RectangleShape rectangle) : base(rectangle)

{

}

#endregion

/// <summary>

/// Проверка за принадлежност на точка point към правоъгълника.

/// В случая на правоъгълник този метод може да не бъде пренаписван, защото

/// Реализацията съвпада с тази на абстрактния клас Shape, който проверява

/// дали точката е в обхващащия правоъгълник на елемента (а той съвпада с

/// елемента в този случай).

/// </summary>

public override bool Contains(PointF point)

{

if (base.Contains(point))

// Проверка дали е в обекта само, ако точката е в обхващащия правоъгълник.

// В случая на правоъгълник - директно връщаме true

return true;

else

// Ако не е в обхващащия правоъгълник, то неможе да е в обекта и => false

return false;

}

/// <summary>

/// Частта, визуализираща конкретния примитив.

/// </summary>

public override void DrawSelf(Graphics grfx)

{

base.DrawSelf(grfx);

base.RotateShape(grfx);

grfx.FillRectangle(new SolidBrush(Color.FromArgb(Opacity, FillColor)), Rectangle.X, Rectangle.Y, Rectangle.Width, Rectangle.Height);

grfx.DrawRectangle(new Pen(BorderColor, BorderWidth), Rectangle.X, Rectangle.Y, Rectangle.Width, Rectangle.Height);

//Center point

Point center = new Point((int)(Rectangle.X + Rectangle.Width / 2), (int)(Rectangle.Y + Rectangle.Height / 2));

//From top left to center

Point p1 = new Point((int)Rectangle.X, (int)Rectangle.Y);

grfx.DrawLine(new Pen(BorderColor, BorderWidth), p1, center);

//From bottom left to center

Point p2 = new Point((int)Rectangle.X, (int)(Rectangle.Y + Rectangle.Height));

grfx.DrawLine(new Pen(BorderColor, BorderWidth), p2, center);

//From middle right to center

Point p3 = new Point((int)((Rectangle.X + Rectangle.Width + Rectangle.X + Rectangle.Width) / 2), (int)((Rectangle.Y + (Rectangle.Y + Rectangle.Height)) / 2));

grfx.DrawLine(new Pen(BorderColor, BorderWidth), p3, center);

grfx.ResetTransform();

}

}

}

—-----------------------------------------DIALOG—---------------------------------------------------------

public void AddRandomExamFigure2()

{

Random rnd = new Random();

int x = rnd.Next(100, 1000);

int y = rnd.Next(100, 600);

ExamFigure2 examFigure2 = new ExamFigure2(new Rectangle(x, y, 200, 100));

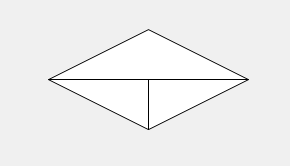
examFigure2.FillColor = Color.White;

examFigure2.BorderColor = Color.Black;

examFigure2.Opacity = 255;

ShapeList.Add(examFigure2);

}



using System;

using System.Collections.Generic;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Draw.src.Model

{

class ExamFigure8 : Shape

{

#region Constructor

public ExamFigure8(RectangleF tri) : base(tri)

{

}

public ExamFigure8(TriangleShape triangle) : base(triangle)

{

}

#endregion

/// <summary>

/// Проверка за принадлежност на точка point към правоъгълника.

/// В случая на правоъгълник този метод може да не бъде пренаписван, защото

/// Реализацията съвпада с тази на абстрактния клас Shape, който проверява

/// дали точката е в обхващащия правоъгълник на елемента (а той съвпада с

/// елемента в този случай).

/// </summary>

public override bool Contains(PointF point)

{

if (base.Contains(point))

// Проверка дали е в обекта само, ако точката е в обхващащия правоъгълник.

// В случая на правоъгълник - директно връщаме true

return true;

else

// Ако не е в обхващащия правоъгълник, то неможе да е в обекта и => false

return false;

}

/// <summary>

/// Частта, визуализираща конкретния примитив.

/// </summary>

public override void DrawSelf(Graphics grfx)

{

base.DrawSelf(grfx);

base.RotateShape(grfx);

Point[] p = { new Point((int)(Rectangle.X + Rectangle.Width / 2), (int)Rectangle.Y),

new Point((int)((Rectangle.X + Rectangle.X) / 2), (int)((Rectangle.Y + (Rectangle.Y + Rectangle.Height)) / 2)),

new Point((int)(Rectangle.X + Rectangle.Width / 2), (int)(Rectangle.Y + Rectangle.Height)),

new Point((int)((Rectangle.X + Rectangle.Width + Rectangle.X + Rectangle.Width) / 2), (int)((Rectangle.Y + (Rectangle.Y + Rectangle.Height)) / 2)) };

grfx.FillPolygon(new SolidBrush(Color.FromArgb(Opacity, FillColor)), p);

grfx.DrawPolygon(new Pen(BorderColor, BorderWidth), p);

//Middle line

Point p1 = new Point((int)((Rectangle.X + Rectangle.X) / 2), (int)((Rectangle.Y + (Rectangle.Y + Rectangle.Height)) / 2));

Point p2 = new Point((int)((Rectangle.X + Rectangle.Width + Rectangle.X + Rectangle.Width) / 2), (int)((Rectangle.Y + (Rectangle.Y + Rectangle.Height)) / 2));

grfx.DrawLine(new Pen(BorderColor, BorderWidth), p1, p2);

//From bottom to center

Point center = new Point((int)(Rectangle.X + Rectangle.Width / 2), (int)(Rectangle.Y + Rectangle.Height / 2));

Point p3 = new Point((int)(Rectangle.X + Rectangle.Width / 2), (int)(Rectangle.Y + Rectangle.Height));

grfx.DrawLine(new Pen(BorderColor, BorderWidth), p3, center);

grfx.ResetTransform();

}

}

}

—-----------------------------------------DIALOG—---------------------------------------------------------

public void AddRandomExamFigure8()

{

Random rnd = new Random();

int x = rnd.Next(100, 1000);

int y = rnd.Next(100, 600);

ExamFigure8 examFigure8 = new ExamFigure8(new Rectangle(x, y, 200, 100));

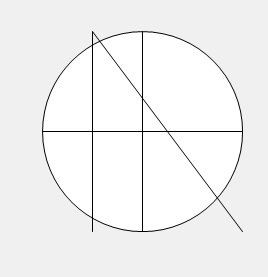
examFigure8.FillColor = Color.White;

examFigure8.BorderColor = Color.Black;

examFigure8.Opacity = 255;

ShapeList.Add(examFigure8);

}



using System;

using System.Collections.Generic;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Draw.src.Model

{

class ExamFigure5 : Shape

{

#region Constructor

public ExamFigure5(RectangleF elp) : base(elp)

{

}

public ExamFigure5(EllipseShape ellipse) : base(ellipse)

{

}

#endregion

/// <summary>

/// Проверка за принадлежност на точка point към правоъгълника.

/// В случая на правоъгълник този метод може да не бъде пренаписван, защото

/// Реализацията съвпада с тази на абстрактния клас Shape, който проверява

/// дали точката е в обхващащия правоъгълник на елемента (а той съвпада с

/// елемента в този случай).

/// </summary>

public override bool Contains(PointF point)

{

if (base.Contains(point))

{

// Проверка дали е в обекта само, ако точката е в обхващащия правоъгълник.

// В случая на правоъгълник - директно връщаме true

float a = Width / 2;

float b = Height / 2;

float x0 = Location.X + a;

float y0 = Location.Y + b;

return Math.Pow((point.X - x0) / a, 2) + Math.Pow((point.Y - y0) / b, 2) - 1 <= 0;

}

else

// Ако не е в обхващащия правоъгълник, то неможе да е в обекта и => false

return false;

}

/// <summary>

/// Частта, визуализираща конкретния примитив.

/// </summary>

public override void DrawSelf(Graphics grfx)

{

base.DrawSelf(grfx);

base.RotateShape(grfx);

grfx.FillEllipse(new SolidBrush(Color.FromArgb(Opacity, FillColor)), Rectangle.X, Rectangle.Y, Rectangle.Width, Rectangle.Height);

grfx.DrawEllipse(new Pen(BorderColor, BorderWidth), Rectangle.X, Rectangle.Y, Rectangle.Width, Rectangle.Height);

//Horizontal line

Point p1 = new Point((int)((Rectangle.X + Rectangle.X) / 2), (int)((Rectangle.Y + (Rectangle.Y + Rectangle.Height)) / 2));

Point p2 = new Point((int)((Rectangle.X + Rectangle.Width + Rectangle.X + Rectangle.Width) / 2), (int)((Rectangle.Y + (Rectangle.Y + Rectangle.Height)) / 2));

grfx.DrawLine(new Pen(BorderColor, BorderWidth), p1, p2);

//First verticle line

Point p3 = new Point((int)(Rectangle.X + Rectangle.Width / 4), (int)(Rectangle.Y));

Point p4 = new Point((int)(Rectangle.X + Rectangle.Width / 4), (int)(Rectangle.Y + Rectangle.Height));

grfx.DrawLine(new Pen(BorderColor, BorderWidth), p3, p4);

//Second verticle line

Point p5 = new Point((int)(Rectangle.X + Rectangle.Width / 2), (int)Rectangle.Y);

Point p6 = new Point((int)(Rectangle.X + Rectangle.Width / 2), (int)(Rectangle.Y + Rectangle.Height));

grfx.DrawLine(new Pen(BorderColor, BorderWidth), p5, p6);

//Third verticle line

Point p7 = new Point((int)(Rectangle.X + Rectangle.Width / 4), (int)(Rectangle.Y));

Point p8 = new Point((int)(Rectangle.X + Rectangle.Width), (int)(Rectangle.Y + Rectangle.Height));

grfx.DrawLine(new Pen(BorderColor, BorderWidth), p7, p8);

grfx.ResetTransform();

}

}

}

—-----------------------------------------DIALOG—---------------------------------------------------------

public void AddRandomExamFigure5()

{

Random rnd = new Random();

int x = rnd.Next(100, 1000);

int y = rnd.Next(100, 600);

ExamFigure5 examFigure5 = new ExamFigure5(new Rectangle(x, y, 200, 200));

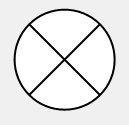
examFigure5.FillColor = Color.White;

examFigure5.BorderColor = Color.Black;

examFigure5.Opacity = 255;

ShapeList.Add(examFigure5);

}



using System;

using System.Collections.Generic;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Draw.src.Model

{

[Serializable]

public class FigNine:ShapeM

{

#region Constructor

public FigNine(RectangleF rect) : base(rect)

{

}

public FigNine(RectangleShapeM rectangle) : base(rectangle)

{

}

#endregion

/// <summary>

/// Проверка за принадлежност на точка point към правоъгълника.

/// В случая на правоъгълник този метод може да не бъде пренаписван, защото

/// Реализацията съвпада с тази на абстрактния клас Shape, който проверява

/// дали точката е в обхващащия правоъгълник на елемента (а той съвпада с

/// елемента в този случай).

/// </summary>

public override bool Contains(PointF point)

{

if (base.Contains(point))

// Проверка дали е в обекта само, ако точката е в обхващащия правоъгълник.

// В случая на правоъгълник - директно връщаме true

return true;

else

// Ако не е в обхващащия правоъгълник, то неможе да е в обекта и => false

return false;

}

/// <summary>

/// Частта, визуализираща конкретния примитив.

/// </summary>

public override void DrawSelf(Graphics grfx)

{

base.DrawSelf(grfx);

base.RotateShapeM(grfx);

grfx.FillEllipse(new SolidBrush(FillColor), Rectangle.X, Rectangle.Y, Rectangle.Width, Rectangle.Height);

grfx.DrawEllipse(new Pen(StrokeColor, BorderWidth), Rectangle.X, Rectangle.Y, Rectangle.Width, Rectangle.Height);

float c = (Rectangle.X + Rectangle.Width / 4) - (Location.X + (Width / 2));

float y = (float)((Location.Y - (-Height / 2)) + Math.Sqrt((Math.Pow(Height / 2, 2)) \* (1 - Math.Pow(c, 2) / Math.Pow(Width / 2, 2))));

PointF p1 = new PointF(Rectangle.X, Rectangle.Y + Rectangle.Height / 2);

PointF p2 = new PointF(Rectangle.X + Rectangle.Width, Rectangle.Y + Rectangle.Height / 2);

PointF p3 = new PointF(Rectangle.X + Rectangle.Width / 2, Rectangle.Y);

PointF p4 = new PointF(Rectangle.X + Rectangle.Width / 2, Rectangle.Y + Rectangle.Height);

grfx.DrawLine(new Pen(StrokeColor, BorderWidth), p1, p2);

grfx.DrawLine(new Pen(StrokeColor, BorderWidth), p3, p4);

grfx.ResetTransform();

}

}

}

—-----------------------------------------DIALOG—---------------------------------------------------------

public void AddRandomFigNine()

{

Random rnd = new Random();

int x = rnd.Next(100, 1000);

int y = rnd.Next(100, 600);

FigNine nine = new FigNine(new Rectangle(x, y, 100, 100));

nine.FillColor = Color.White;

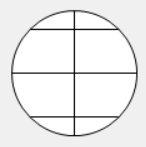
nine.StrokeColor = Color.Black;

nine.BorderWidth = 2;

nine.ShapeAngle = 45;

ShapeMList.Add(nine);

}



using System;

using System.Drawing;

using System.Drawing.Drawing2D;

namespace Draw

{

/// <summary>

/// Класът CircleShape представлява фигура кръг, която е наследник на базовия Shape.

/// </summary>

[Serializable]

public class Figure7 : Shape

{

#region Constructors

public Figure7(RectangleF rect) : base(TurnToCircle(rect))

{

}

public Figure7(Shape shape) : base(TurnToCircle(shape.Rectangle))

{

}

// Конструктор по подразбиране – нужен при зареждане от файл

public Figure7() : base(new RectangleF()) { }

#endregion

/// <summary>

/// Преобразува даден правоъгълник в квадрат със страната на по-малкия му размер.

/// Използва се при създаване на фигурата, за да гарантира, че тя винаги е кръг, а не елипса.

/// </summary>

private static RectangleF TurnToCircle(RectangleF rect)

{

float size = Math.Min(rect.Width, rect.Height);

return new RectangleF(rect.X, rect.Y, size, size);

}

/// <summary>

/// Проверява дали дадена точка се намира вътре в кръга.

/// Взима предвид трансформации като мащабиране, завъртане и преместване.

/// </summary>

public override bool Contains(PointF point)

{

// Клонираме и инвертираме матрица, за да върнем точката обратно в локалната система

Matrix m = TransformationMatrix.Clone();

m.Invert();

// Преобразуваме точката обратно към координатите на оригиналната фигура

PointF[] points = new PointF[] { point };

m.TransformPoints(points);

PointF transformedPoint = points[0];

// Центъра и радиуса на кръга

float centerX = Rectangle.X + Rectangle.Width / 2;

float centerY = Rectangle.Y + Rectangle.Height / 2;

float radius = Rectangle.Width / 2; // Width = Height

// Проверка на разтояние дали е между точката и центъра е по малко или равно на радиуса

float dx = transformedPoint.X - centerX;

float dy = transformedPoint.Y - centerY;

return dx \* dx + dy \* dy <= radius \* radius;

}

/// <summary>

/// Рисува кръга на зададената

/// </summary>

public override void DrawSelf(Graphics grfx)

{

base.DrawSelf(grfx);

var state = grfx.Save();

grfx.Transform = TransformationMatrix;

Color fill = Color.FromArgb(FillOpacity, FillColor.R, FillColor.G, FillColor.B);

Brush brush = new SolidBrush(fill);

Pen pen = new Pen(Color.Black, StrokeWidth);

// кръг

float size = Math.Min(Rectangle.Width, Rectangle.Height);

float x = Rectangle.X;

float y = Rectangle.Y;

RectangleF circleBounds = new RectangleF(x, y, size, size);

float cx = x + size / 2;

float cy = y + size / 2;

float radius = size / 2;

grfx.FillEllipse(brush, circleBounds);

grfx.DrawEllipse(pen, circleBounds);

// хоризонтални

float h1 = y + size \* 0.15f;

float h2 = y + size \* 0.50f;

float h3 = y + size \* 0.85f;

grfx.DrawLine(pen, x + size \* 0.15f, h1, x + size \* 0.85f , h1); // горна

grfx.DrawLine(pen, x , h2, x + size , h2); // централна

grfx.DrawLine(pen, x + size \* 0.15f, h3, x + size \* 0.85f, h3); // долна

// вертикална централна

grfx.DrawLine(pen, cx, y , cx, y + size );

grfx.Restore(state);

}

}

}

—-----------------------------------------DIALOG—---------------------------------------------------------

public void AddRandomFigure7()

{

Random rnd = new Random();

int x = rnd.Next(100, 1000);

int y = rnd.Next(100, 600);

Figure7 circle = new Figure7(new Rectangle(x, y, 100, 200));

circle.FillColor = Color.White;

circle.StrokeColor = Color.Black;

circle.StrokeWidth = 1;

ShapeList.Add(circle);

}