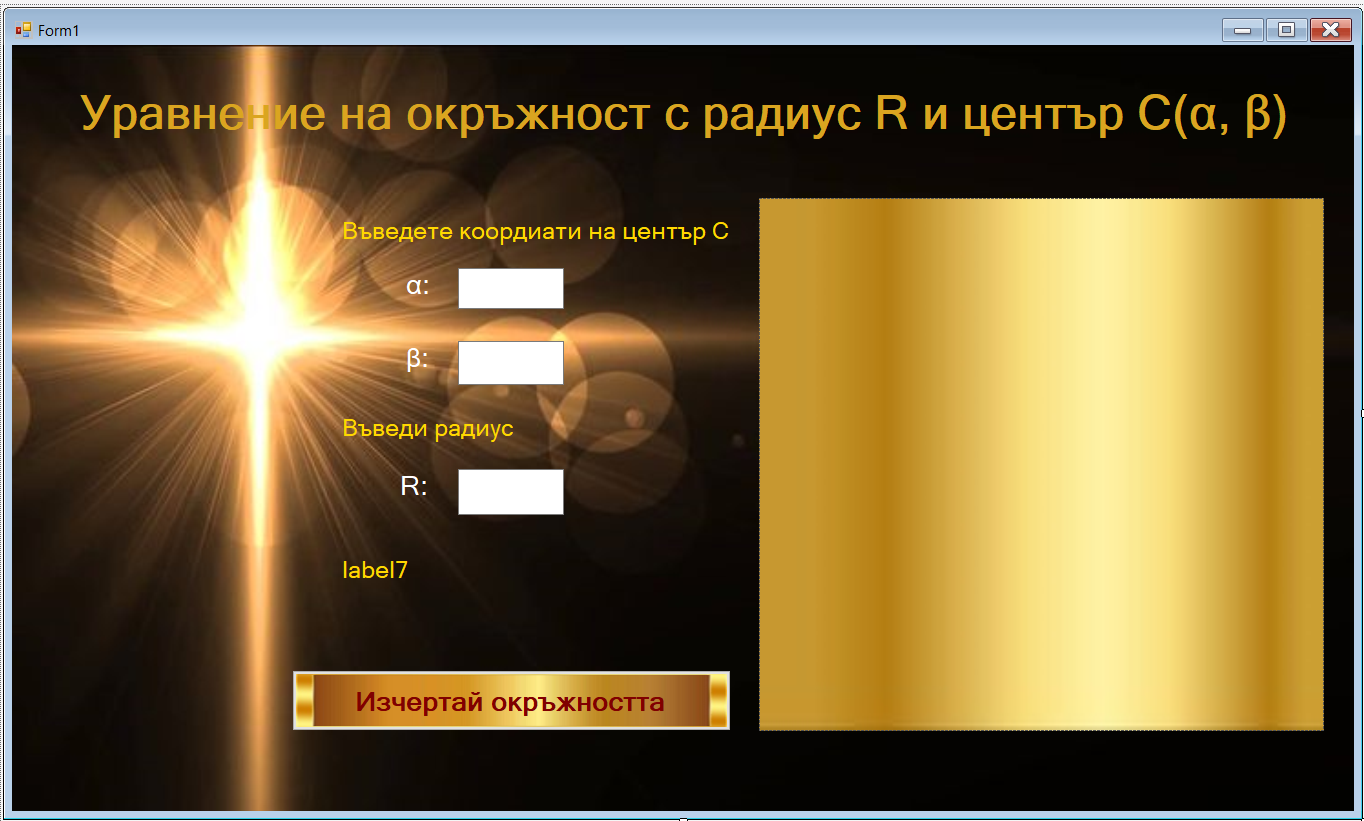
25.11.2024g. Кодове:

**Задача1:** Съставете уравнение на окръжност с център C и радиус R:

* C(2,−5) и R=4 – **1 екип**
* C(-3,4) и минаваща през т. O(0,0) – **2 екип**
* C(0, 4) и минаваща през през т.А(5,-8) **– 3 екип**



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 Задача1\_\_\_Windows\_Form

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

// Създаване на Bitmap за рисуване

// Прочитане на координатите на центъра и радиуса

int centerX = int.Parse(textBox1.Text);

int centerY = int.Parse(textBox2.Text);

int radius = int.Parse(textBox3.Text);

// Показване на уравнението на окръжността в Label

label7.Text = $"Уравнение на окръжността: \n (x - {centerX})² + (y - {centerY})² = {radius}²";

// Създаване на Bitmap за рисуване

Bitmap bmp = new Bitmap(pictureBox1.Width, pictureBox1.Height);

using (Graphics g = Graphics.FromImage(bmp))

{

g.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.AntiAlias;

// Преместване на началната точка (0,0) в центъра на PictureBox

g.TranslateTransform(pictureBox1.Width / 2, pictureBox1.Height / 2);

// Обръщане на оста Y така, че положителната Y ос да сочи нагоре

g.ScaleTransform(1, -1);

// Задаване на мащаб - всяко деление е 10 пиксела

int scale = 10;

// Рисуване на оси X и Y

g.DrawLine(Pens.Black, -pictureBox1.Width / 2, 0, pictureBox1.Width / 2, 0); // X ос

g.DrawLine(Pens.Black, 0, -pictureBox1.Height / 2, 0, pictureBox1.Height / 2); // Y ос

// Рисуване на деления по X оста

for (int i = -pictureBox1.Width / 2; i <= pictureBox1.Width / 2; i += scale)

{

g.DrawLine(Pens.Gray, i, -5, i, 5); // Малки вертикални линии за деленията по X оста

}

// Рисуване на деления по Y оста

for (int i = -pictureBox1.Height / 2; i <= pictureBox1.Height / 2; i += scale)

{

g.DrawLine(Pens.Gray, -5, i, 5, i); // Малки хоризонтални линии за деленията по Y оста

}

// Преобразуване на координатите на центъра и радиуса спрямо мащаба

int drawX = centerX \* scale - radius \* scale;

int drawY = centerY \* scale - radius \* scale;

int diameter = radius \* 2 \* scale;

// Рисуване на окръжността

g.DrawEllipse(Pens.Maroon, drawX, drawY, diameter, diameter);

}

// Задаване на изображението в PictureBox

pictureBox1.Image = bmp;

}

private void Form1\_Load(object sender, EventArgs e)

{

}

}

}

**Обяснение на кода:**

**1. Пространства от имена (Using Directives):**

* using System;  
  Основна библиотека за C#, включваща типове като числа, дати и методи за работа с конзолата.
* using System.Drawing;  
  Библиотека за работа с графики – рисуване, изображения и цветове.
* using System.Windows.Forms;  
  Предоставя класове за създаване на Windows Forms приложения, като прозорци, бутони, текстови полета и др.

**2. Пространство на имената:**

namespace Задача1\_\_\_Windows\_Form

* Това е контейнер за всички класове в програмата. То позволява организиране на кода и предотвратява конфликти между имената на класове.

**3. Клас Form1:**

public partial class Form1 : Form

* Form1 е основният клас на приложението и наследява функционалността на базовия клас Form.
* Този клас съдържа всички елементи на формата, като текстови полета, бутони и логиката за тях.

**4. Конструктор:**

public Form1()

{

InitializeComponent();

}

* Конструкторът инициализира компонентите на формата. Методът InitializeComponent автоматично добавя всички визуални елементи (бутони, текстови полета и др.).

**5. Събитие при кликване на бутона (button1\_Click):**

private void button1\_Click(object sender, EventArgs e)

* Методът се изпълнява при натискане на бутона. Ето какво прави:

**a. Прочитане на входни данни:**

int centerX = int.Parse(textBox1.Text);

int centerY = int.Parse(textBox2.Text);

int radius = int.Parse(textBox3.Text);

* Извлича стойностите за центъра (x,y)(x, y)(x,y) и радиуса R, въведени от потребителя в текстовите полета.

**b. Показване на уравнението:**

label7.Text = $"Уравнение на окръжността: \n (x - {centerX})² + (y - {centerY})² = {radius}²";

* Съставя и показва уравнението на окръжността на етикета label7.

**c. Създаване на графичен обект (Bitmap):**

Bitmap bmp = new Bitmap(pictureBox1.Width, pictureBox1.Height);

* Създава празно изображение с размерите на PictureBox.

**d. Настройка на графиката:**

using (Graphics g = Graphics.FromImage(bmp))

{

g.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.AntiAlias;

}

* Graphics се използва за рисуване на графични обекти върху Bitmap.
* SmoothingMode.AntiAlias подобрява качеството на рисунката.

**e. Рисуване на оси:**

g.DrawLine(Pens.Black, -pictureBox1.Width / 2, 0, pictureBox1.Width / 2, 0); // X ос

g.DrawLine(Pens.Black, 0, -pictureBox1.Height / 2, 0, pictureBox1.Height / 2); // Y ос

* Рисува X и Y оси в центъра на PictureBox.

**f. Рисуване на окръжността:**

g.DrawEllipse(Pens.Maroon, drawX, drawY, diameter, diameter);

* Изчислява координатите и размера на окръжността, като мащабира стойностите спрямо PictureBox.
* Използва DrawEllipse за да начертае окръжността.

**g. Задаване на изображението в PictureBox:**

pictureBox1.Image = bmp;

* Задава нарисуваното изображение като съдържание на PictureBox.

**6. Празен метод (Form1\_Load):**

private void Form1\_Load(object sender, EventArgs e)

* Този метод се изпълнява при зареждане на формата, но в момента е празен. Може да се използва за предварителни настройки.

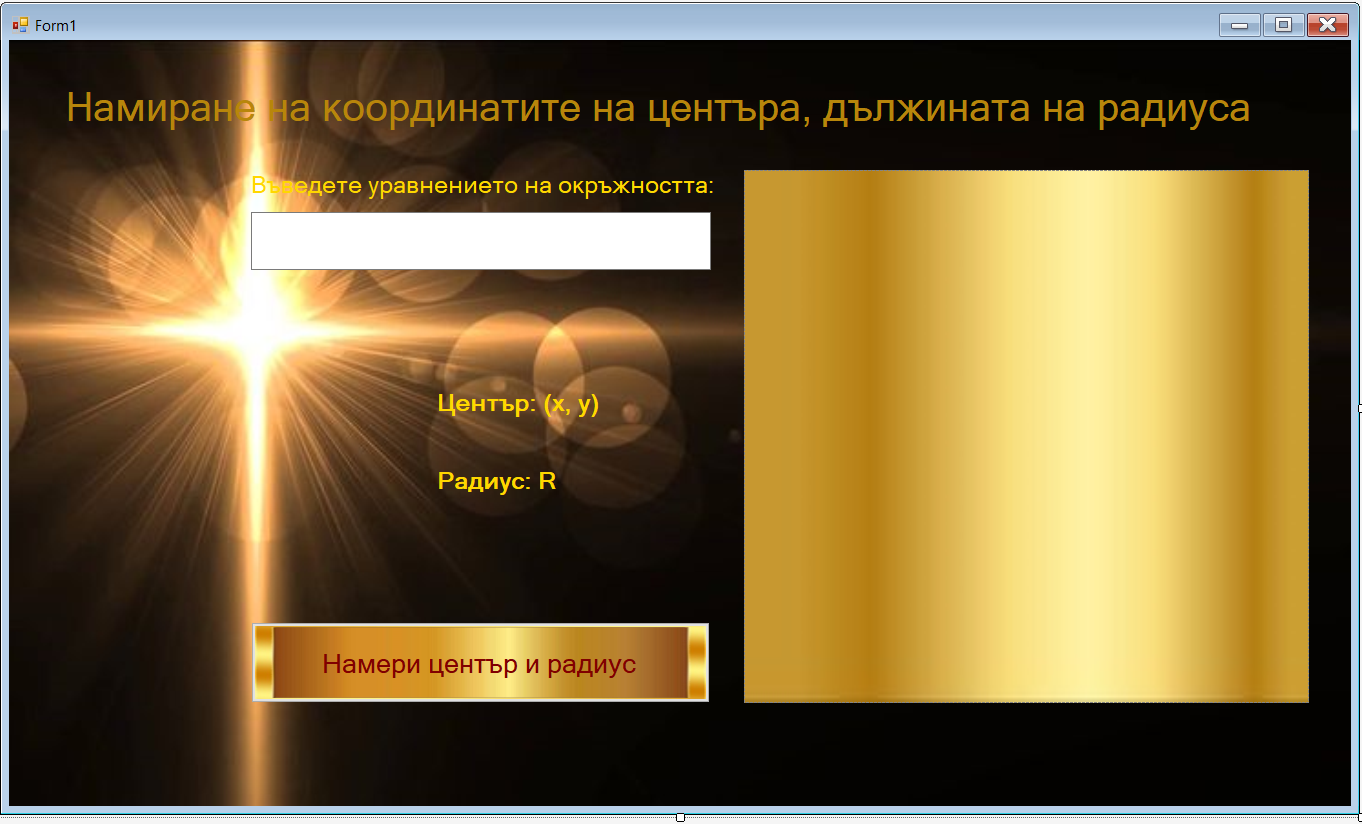
**Заключение:**

Кодът създава Windows Forms приложение, което:

1. Изчислява уравнението на окръжността.
2. Рисува окръжността върху координатна система.
3. Визуализира резултатите в PictureBox.

**Задача 2:** Дадена е окръжност K: x2 + y2 − 2x + 6y − 6 = 0 .

* Да се намерят координатите на центъра и дължината на радиуса на K.
* Да се напише нормалното ѝ уравнение.
* Да се начертае окръжността в координатната равнина.



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 Задача2\_25.\_11.\_24г

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

// Дефинирано уравнение: x^2 + y^2 - 2x + 6y - 6 = 0

// Статично зададени коефициенти

int D = -2; // Коэффициент пред x

int E = 6; // Коэффициент пред y

int F = -6; // Свободен член

// Изчисляване на координатите на центъра

double centerX = -D / 2.0;

double centerY = -E / 2.0;

// Изчисляване на радиуса

double radius = Math.Sqrt((D / 2.0) \* (D / 2.0) + (E / 2.0) \* (E / 2.0) - F);

// Показване на резултатите в етикетите

label3.Text = $"Център: ({centerX}, {centerY})";

label4.Text = $"Радиус: {radius:F2}";

// Рисуване на окръжността

DrawCircle(centerX, centerY, radius);

}

private void DrawCircle(double centerX, double centerY, double radius)

{

// Създаване на Bitmap за рисуване

Bitmap bmp = new Bitmap(pictureBox1.Width, pictureBox1.Height);

using (Graphics g = Graphics.FromImage(bmp))

{

g.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.AntiAlias;

// Центриране на координатната система в PictureBox

g.TranslateTransform(pictureBox1.Width / 2, pictureBox1.Height / 2);

g.ScaleTransform(1, -1); // Обръщане на оста Y

// Рисуване на оси

g.DrawLine(Pens.Gray, -pictureBox1.Width / 2, 0, pictureBox1.Width / 2, 0); // X ос

g.DrawLine(Pens.Gray, 0, -pictureBox1.Height / 2, 0, pictureBox1.Height / 2); // Y ос

// Рисуване на окръжността

int scale = 20; // Мащаб: 1 единица = 20 пиксела

int drawX = (int)(centerX \* scale - radius \* scale);

int drawY = (int)(centerY \* scale - radius \* scale);

int diameter = (int)(radius \* 2 \* scale);

g.DrawEllipse(Pens.Maroon, drawX, drawY, diameter, diameter);

}

// Задаване на изображението в PictureBox

pictureBox1.Image = bmp;

}

}

}

**Обяснение на кода:**

**1. Пространства от имена (using directives):**

* using System;  
  Библиотека, която предоставя основни типове и методи, като числа, дати и текст.
* using System.Drawing;  
  За работа с графики – рисуване на форми, изображения и използване на цветове.
* using System.Windows.Forms;  
  Основната библиотека за създаване на Windows Forms приложения (визуални приложения в Windows).

**2. Основен клас Form1:**

public partial class Form1 : Form

* Класът Form1 е наследник на класа Form, който съдържа всички необходими функции за създаване на прозоречни приложения (Windows Forms).
* Ключовият метод е InitializeComponent(), който настройва елементите на формата.

**3. Бутон за изчисления button1\_Click:**

Този метод се изпълнява, когато потребителят кликне върху бутона.

**a. Определяне на коефициентите:**

int D = -2; // Коэффициент пред x

int E = 6; // Коэффициент пред y

int F = -6; // Свободен член

* Уравнението на окръжността е дадено в общ вид: x2+y2+Dx+Ey+F=0x^2 + y^2 + Dx + Ey + F = 0x2+y2+Dx+Ey+F=0
* Стойностите D,ED, ED,E и FFF са взети директно от уравнението.

**b. Изчисляване на центъра на окръжността:**

double centerX = -D / 2.0;

double centerY = -E / 2.0;

* Центърът на окръжността се изчислява по формулата: C(a,b)=(−D2,−E2)C(a, b) = \left(-\frac{D}{2}, -\frac{E}{2}\right)C(a,b)=(−2D​,−2E​)

**c. Изчисляване на радиуса:**

double radius = Math.Sqrt((D / 2.0) \* (D / 2.0) + (E / 2.0) \* (E / 2.0) - F);

* Радиусът на окръжността се изчислява чрез: R=(D2)2+(E2)2−FR = \sqrt{\left(\frac{D}{2}\right)^2 + \left(\frac{E}{2}\right)^2 - F}R=(2D​)2+(2E​)2−F​

**d. Показване на резултати:**

label3.Text = $"Център: ({centerX}, {centerY})";

label4.Text = $"Радиус: {radius:F2}";

* Резултатите (центърът и радиусът) се показват в label3 и label4.

**e. Рисуване на окръжността:**

DrawCircle(centerX, centerY, radius);

* Методът DrawCircle визуализира окръжността в PictureBox.

**4. Метод DrawCircle:**

Този метод се грижи за визуализацията на окръжността в координатна система.

**a. Създаване на Bitmap:**

Bitmap bmp = new Bitmap(pictureBox1.Width, pictureBox1.Height);

using (Graphics g = Graphics.FromImage(bmp))

* Създава празно изображение с размерите на PictureBox.
* Обектът Graphics се използва за рисуване върху това изображение.

**b. Настройки за рисуване:**

g.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.AntiAlias;

g.TranslateTransform(pictureBox1.Width / 2, pictureBox1.Height / 2);

g.ScaleTransform(1, -1);

* SmoothingMode.AntiAlias прави линиите гладки.
* TranslateTransform премества центъра на координатната система в центъра на PictureBox.
* ScaleTransform обръща оста Y, за да съответства на стандартната математическа координатна система (където положителната Y ос сочи нагоре).

**c. Рисуване на координатни оси:**

g.DrawLine(Pens.Gray, -pictureBox1.Width / 2, 0, pictureBox1.Width / 2, 0); // X ос

g.DrawLine(Pens.Gray, 0, -pictureBox1.Height / 2, 0, pictureBox1.Height / 2); // Y ос

* Осите X и Y се рисуват със сив цвят.

**d. Рисуване на окръжността:**

int scale = 20;

int drawX = (int)(centerX \* scale - radius \* scale);

int drawY = (int)(centerY \* scale - radius \* scale);

int diameter = (int)(radius \* 2 \* scale);

g.DrawEllipse(Pens.Maroon, drawX, drawY, diameter, diameter);

* **Мащабиране:** 111 математическа единица е равна на 202020 пиксела.
* Изчислява се диаметърът и позицията на окръжността в мащабирания координатен план.
* Окръжността се рисува с DrawEllipse.

**e. Показване в PictureBox:**

pictureBox1.Image = bmp;

* Задава нарисуваното изображение като съдържание на PictureBox.

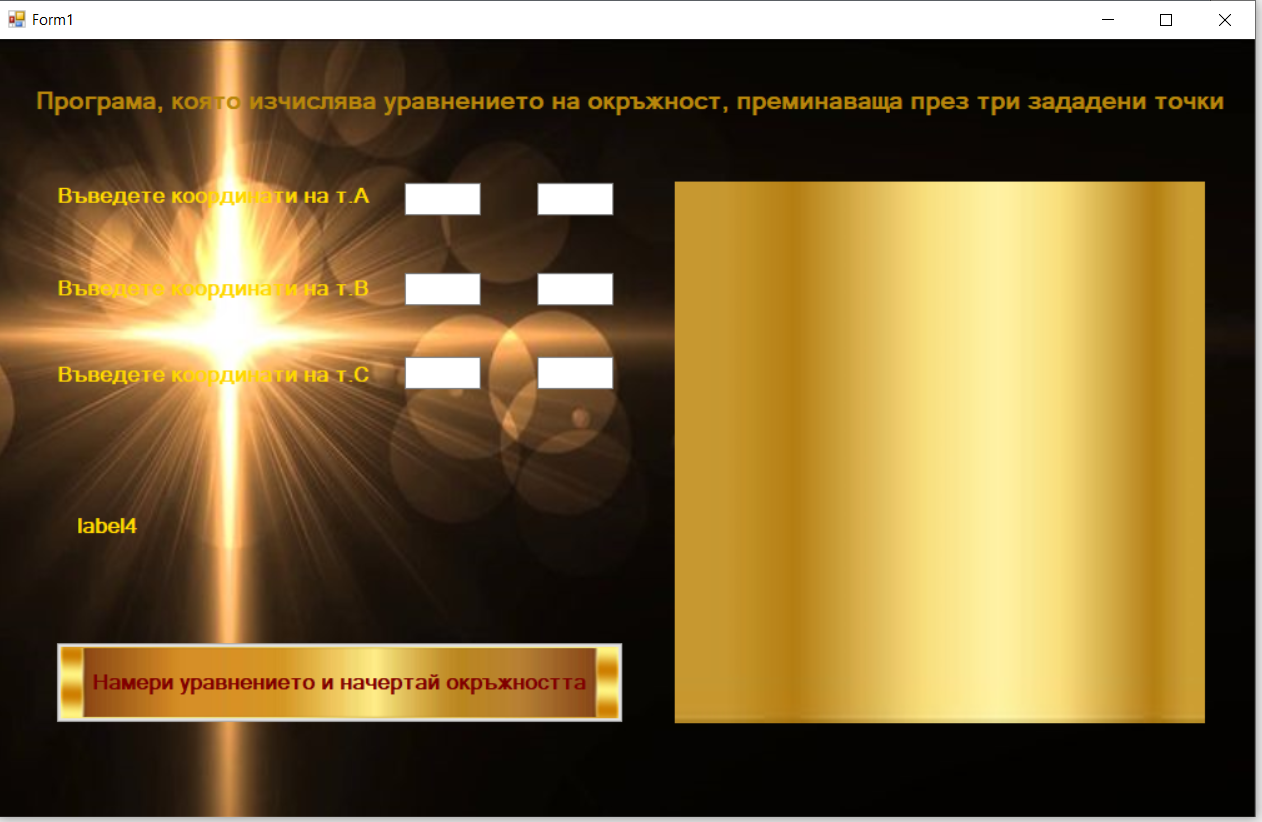
**5. Основни резултати:**

* Център: (1,−3)(1, -3)(1,−3)
* Радиус: 444
* Визуализация на окръжността върху координатна система.

**Задача 3:** Дадени са точките A(1,3), B(3,−1), C(5,0).

* Да се намери уравнението на окръжността, определена от точките A, B и C.

Начертайте окръжността в координатната система



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 Задача3\_\_\_11.\_11.\_24

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

// Задаване на координати на точките

double ax = 1, ay = 3;

double bx = 3, by = -1;

double cx = 5, cy = 0;

// Изчисляване на детерминанти за центъра на окръжността

double d = 2 \* (ax \* (by - cy) + bx \* (cy - ay) + cx \* (ay - by));

double ux = ((Math.Pow(ax, 2) + Math.Pow(ay, 2)) \* (by - cy) +

(Math.Pow(bx, 2) + Math.Pow(by, 2)) \* (cy - ay) +

(Math.Pow(cx, 2) + Math.Pow(cy, 2)) \* (ay - by)) / d;

double uy = ((Math.Pow(ax, 2) + Math.Pow(ay, 2)) \* (cx - bx) +

(Math.Pow(bx, 2) + Math.Pow(by, 2)) \* (ax - cx) +

(Math.Pow(cx, 2) + Math.Pow(cy, 2)) \* (bx - ax)) / d;

// Изчисляване на радиуса на окръжността

double radius = Math.Sqrt(Math.Pow(ux - ax, 2) + Math.Pow(uy - ay, 2));

// Показване на уравнението на окръжността

label4.Text = $"Уравнение на окръжността:\n(x - {ux})² + (y - {uy})² = {Math.Pow(radius, 2)}";

// Рисуване на окръжността в PictureBox

Bitmap bmp = new Bitmap(pictureBox1.Width, pictureBox1.Height);

using (Graphics g = Graphics.FromImage(bmp))

{

g.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.AntiAlias;

// Центриране на координатната система в средата на PictureBox

g.TranslateTransform(pictureBox1.Width / 2, pictureBox1.Height / 2);

g.ScaleTransform(1, -1); // Обръщане на Y оста

int scale = 10; // мащаб

// Рисуване на оси

g.DrawLine(Pens.Black, -pictureBox1.Width / 2, 0, pictureBox1.Width / 2, 0);

g.DrawLine(Pens.Black, 0, -pictureBox1.Height / 2, 0, pictureBox1.Height / 2);

// Рисуване на окръжността

int drawX = (int)((ux - radius) \* scale);

int drawY = (int)((uy - radius) \* scale);

int diameter = (int)(2 \* radius \* scale);

g.DrawEllipse(Pens.Blue, drawX, drawY, diameter, diameter);

}

// Задаване на изображението в PictureBox

pictureBox1.Image = bmp;

}

private void Form1\_Load(object sender, EventArgs e)

{

}

}

}

**Обяснение на кода:**

**1. Пространство от имена и клас**

namespace Задача3\_25.\_11.\_24г

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

}

}

* **namespace**: Това е пространството от имена за проекта. Кодът е в проект с име Задача3\_25.\_11.\_24г.
* **Form1**: Това е основният клас на Windows Forms приложението. Той наследява класа Form, което означава, че представлява формуляр с графичен интерфейс.
* **InitializeComponent()**: Методът се използва за инициализиране на формата и всички графични елементи (бутони, текстови полета и т.н.).

**2. Методът button1\_Click**

Това е събитие, което се задейства при натискане на бутона.

**а) Координати на точките**

double ax = 1, ay = 3;

double bx = 3, by = -1;

double cx = 5, cy = 0;

* Определят координатите на три точки A(x1,y1)A(x\_1, y\_1)A(x1​,y1​), B(x2,y2)B(x\_2, y\_2)B(x2​,y2​), C(x3,y3)C(x\_3, y\_3)C(x3​,y3​). Тези точки ще бъдат използвани за изчисляване на центъра и радиуса на окръжността.

**б) Изчисляване на центъра на окръжността**

double d = 2 \* (ax \* (by - cy) + bx \* (cy - ay) + cx \* (ay - by));

double ux = ((Math.Pow(ax, 2) + Math.Pow(ay, 2)) \* (by - cy) +

(Math.Pow(bx, 2) + Math.Pow(by, 2)) \* (cy - ay) +

(Math.Pow(cx, 2) + Math.Pow(cy, 2)) \* (ay - by)) / d;

double uy = ((Math.Pow(ax, 2) + Math.Pow(ay, 2)) \* (cx - bx) +

(Math.Pow(bx, 2) + Math.Pow(by, 2)) \* (ax - cx) +

(Math.Pow(cx, 2) + Math.Pow(cy, 2)) \* (bx - ax)) / d;

* **d**: Определител, който проверява дали точките са на една линия. Ако d=0d = 0d=0, точките не определят окръжност.
* **ux, uy**: Това са координатите на центъра на окръжността. Формулите използват детерминанти, за да изчислят точката, където е центърът на окръжността.

**в) Изчисляване на радиуса**

double radius = Math.Sqrt(Math.Pow(ux - ax, 2) + Math.Pow(uy - ay, 2));

* Радиусът се изчислява като разстоянието между центъра (ux,uyux, uyux,uy) и една от зададените точки, например A(ax,ay)A(ax, ay)A(ax,ay).

**г) Показване на уравнението на окръжността**

label4.Text = $"Уравнение на окръжността:\n(x - {ux})² + (y - {uy})² = {Math.Pow(radius, 2)}";

* Уравнението се форматира в стандартен вид: (x−ux)2+(y−uy)2=R2(x - u\_x)^2 + (y - u\_y)^2 = R^2(x−ux​)2+(y−uy​)2=R2
* Използва се свойството Text на label4, за да покаже резултата в графичния интерфейс.

**3. Рисуване на окръжността**

Bitmap bmp = new Bitmap(pictureBox1.Width, pictureBox1.Height);

using (Graphics g = Graphics.FromImage(bmp))

{

g.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.AntiAlias;

// Центриране на координатната система в PictureBox

g.TranslateTransform(pictureBox1.Width / 2, pictureBox1.Height / 2);

g.ScaleTransform(1, -1); // Обръщане на Y оста

int scale = 10; // мащаб

// Рисуване на оси

g.DrawLine(Pens.Black, -pictureBox1.Width / 2, 0, pictureBox1.Width / 2, 0);

g.DrawLine(Pens.Black, 0, -pictureBox1.Height / 2, 0, pictureBox1.Height / 2);

// Рисуване на окръжността

int drawX = (int)((ux - radius) \* scale);

int drawY = (int)((uy - radius) \* scale);

int diameter = (int)(2 \* radius \* scale);

g.DrawEllipse(Pens.Blue, drawX, drawY, diameter, diameter);

}

// Задаване на изображението в PictureBox

pictureBox1.Image = bmp;

**Обяснение:**

1. **Създаване на Bitmap**:
   * Това е обект, върху който се рисуват графики. Размерите са зададени спрямо PictureBox.
2. **Graphics обект**:
   * Отговаря за рисуването върху Bitmap.
   * **TranslateTransform**: Центрира координатната система в средата на PictureBox.
   * **ScaleTransform**: Обръща оста YYY, така че положителната посока да сочи нагоре (като в математическа координатна система).
3. **Мащабиране**:
   * Променливата scale определя колко пиксела съответстват на една единица от координатната система.
4. **Рисуване на оси**:
   * Чертае хоризонтална и вертикална линия през центъра на PictureBox.
5. **Рисуване на окръжността**:
   * Изчислява координатите на окръжността, мащабирани според scale.
   * Рисува окръжността с метод DrawEllipse.

**4. Как работи програмата**

1. Потребителят въвежда координатите на трите точки A(x,y)A(x, y)A(x,y), B(x,y)B(x, y)B(x,y), C(x,y)C(x, y)C(x,y).
2. При натискане на бутона:
   * Програмата изчислява центъра и радиуса на окръжността.
   * Показва уравнението на окръжността в текстов етикет.
   * Рисува окръжността и координатните оси в PictureBox.

**Забележки:**

* Ако d=0d = 0d=0, точките лежат на една линия, и програмата трябва да покаже съобщение за грешка.
* Уравнението и графиката се обновяват автоматично след всяко натискане на бутона.

**Задача 4:** Дадени са уравнения на окръжност и права. Да се намерят общите им точки или да се докаже, че няма такива.

(x − 2)2 + (2y + 2)2 = 1 и x − y + 1 = 0 - 1 екип

(x − 1)2+(y + 1)2 = 10 и 3x + y + 8 = 0 - 2 екип

(x + 3)2+(y − 2)2 = 13 и x + y − 4 = 0 - 3 екип

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 Задача4\_\_\_11.\_11.\_24

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void label9\_Click(object sender, EventArgs e)

{

}

private void textBox5\_TextChanged(object sender, EventArgs e)

{

}

private void textBox6\_TextChanged(object sender, EventArgs e)

{

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

}

private void label8\_Click(object sender, EventArgs e)

{

}

private void label7\_Click(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

// Прочитане на стойности от текстовите полета за окръжността

double h = double.Parse(textBox1.Text);

double k = double.Parse(textBox2.Text);

double R = double.Parse(textBox3.Text);

// Прочитане на стойности от текстовите полета за правата

double a = double.Parse(textBox4.Text);

double b = double.Parse(textBox5.Text);

double c = double.Parse(textBox6.Text);

// Създаване на ново изображение за рисуване в PictureBox

Bitmap bmp = new Bitmap(pictureBox1.Width, pictureBox1.Height);

using (Graphics g = Graphics.FromImage(bmp))

{

g.Clear(Color.White); // Запълване на фона с бял цвят

// Център на PictureBox (координатни оси)

int centerX = pictureBox1.Width / 2;

int centerY = pictureBox1.Height / 2;

int scale = 20; // мащаб - пиксели на единица

// Рисуване на координатни оси

g.DrawLine(Pens.Black, 0, centerY, pictureBox1.Width, centerY); // X ос

g.DrawLine(Pens.Black, centerX, 0, centerX, pictureBox1.Height); // Y ос

// Рисуване на окръжността

int radius = (int)(R \* scale);

int circleX = centerX + (int)(h \* scale) - radius;

int circleY = centerY - (int)(k \* scale) - radius;

g.DrawEllipse(Pens.Blue, circleX, circleY, 2 \* radius, 2 \* radius);

// Рисуване на правата

if (b != 0)

{

// Преобразуване на уравнението на правата към y = (-a/b)x - c/b

float slope = (float)(-a / b);

float intercept = (float)(-c / b);

// Изчисляване на точките за рисуване на правата в рамките на PictureBox

PointF point1 = new PointF(0, centerY - intercept \* scale);

PointF point2 = new PointF(pictureBox1.Width, centerY - (slope \* (pictureBox1.Width - centerX) / scale + intercept) \* scale);

g.DrawLine(Pens.Red, point1, point2);

}

else

{

// Ако b = 0, правата е вертикална: x = -c / a

int x = centerX + (int)(-c / a \* scale);

g.DrawLine(Pens.Red, x, 0, x, pictureBox1.Height);

}

// Изчисляване на пресечните точки (ако съществуват)

FindIntersectionPoints(g, centerX, centerY, scale, h, k, R, a, b, c);

}

// Задаване на изображението в PictureBox

pictureBox1.Image = bmp;

}

private void FindIntersectionPoints(Graphics g, int centerX, int centerY, int scale, double h, double k, double R, double a, double b, double c)

{

// Изчисляване на параметрите за квадратно уравнение

double A = a \* a + b \* b;

double B = 2 \* (a \* c + a \* b \* k - b \* b \* h);

double C = c \* c + 2 \* b \* c \* k + b \* b \* (h \* h + k \* k - R \* R);

double discriminant = B \* B - 4 \* A \* C;

// Задаване на по-висока стойност за epsilon

const double epsilon = 0.6;

// Диагностика: показване на стойностите за проверка

label11.Text = $"A = {A:F2}, B = {B:F2}, C = {C:F2}, Discriminant = {discriminant:F2}";

if (discriminant < -epsilon)

{

// Няма реални пресечни точки

label10.Text = "Няма пресечни точки.";

}

else if (Math.Abs(discriminant) < epsilon)

{

// Една пресечна точка (допирателна)

double x = -B / (2 \* A);

double y = (-a \* x - c) / b;

// Преобразуване на координатите за рисуване

PointF intersection = new PointF((float)(centerX + x \* scale), (float)(centerY - y \* scale));

// Рисуване на пресечната точка

g.FillEllipse(Brushes.Green, intersection.X - 3, intersection.Y - 3, 6, 6);

// Показване на резултата в етикета

label10.Text = $"Една пресечна точка:\nТочка: ({x:F2}, {y:F2})";

}

else

{

// Две пресечни точки

double x1 = (-B + Math.Sqrt(discriminant)) / (2 \* A);

double x2 = (-B - Math.Sqrt(discriminant)) / (2 \* A);

double y1 = (-a \* x1 - c) / b;

double y2 = (-a \* x2 - c) / b;

// Преобразуване на координатите за рисуване

PointF intersection1 = new PointF((float)(centerX + x1 \* scale), (float)(centerY - y1 \* scale));

PointF intersection2 = new PointF((float)(centerX + x2 \* scale), (float)(centerY - y2 \* scale));

// Рисуване на пресечните точки

g.FillEllipse(Brushes.Green, intersection1.X - 3, intersection1.Y - 3, 6, 6);

g.FillEllipse(Brushes.Green, intersection2.X - 3, intersection2.Y - 3, 6, 6);

// Показване на резултата в етикета

label10.Text = $"Пресечни точки:\nТочка 1: ({x1:F2}, {y1:F2})\nТочка 2: ({x2:F2}, {y2:F2})";

}

}

}

}

**Какво да въведете?**

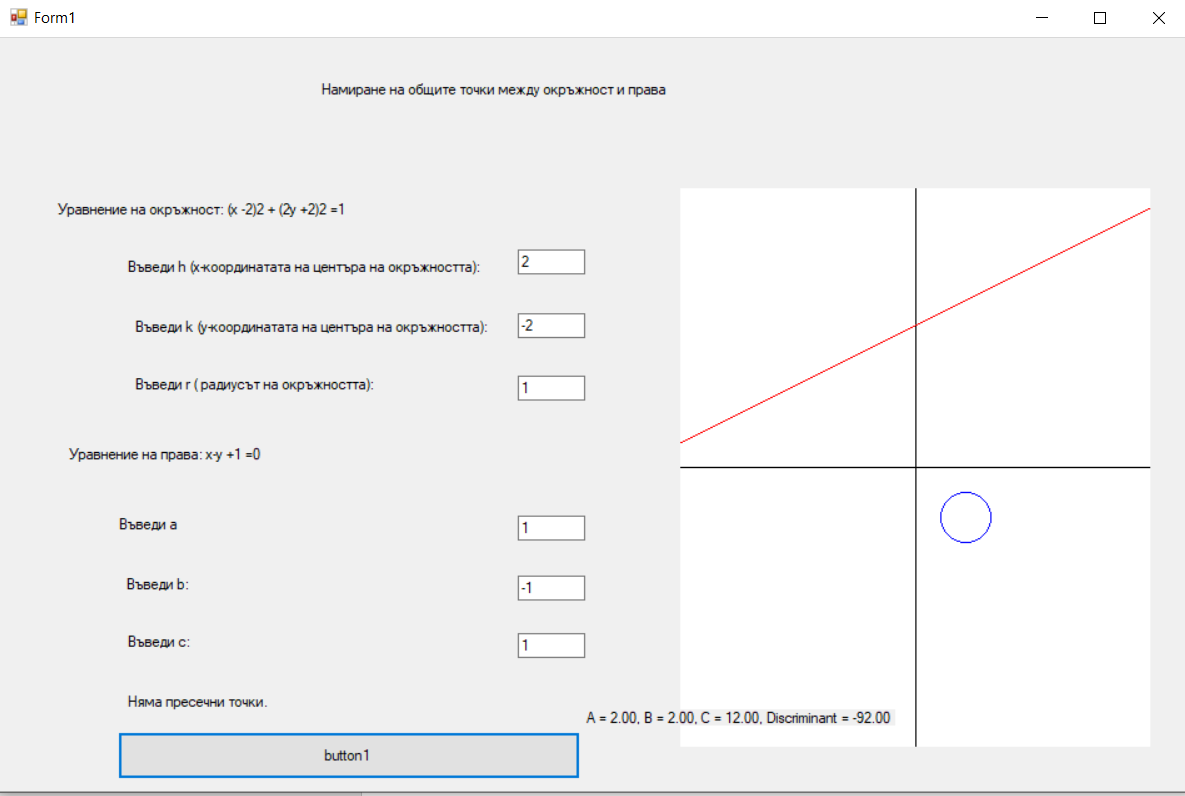
**За А) (x − 2)2 + (2y + 2)2 = 1 и x − y + 1 = 0 - 1 екип**

 Окръжност:

* В текстовото поле за h (x-координата на центъра): 2
* В текстовото поле за k (y-координата на центъра): -2
* В текстовото поле за радиуса R: 1

 Права:

* В текстовото поле за a: 1
* В текстовото поле за b: -1
* В текстовото поле за c: 1



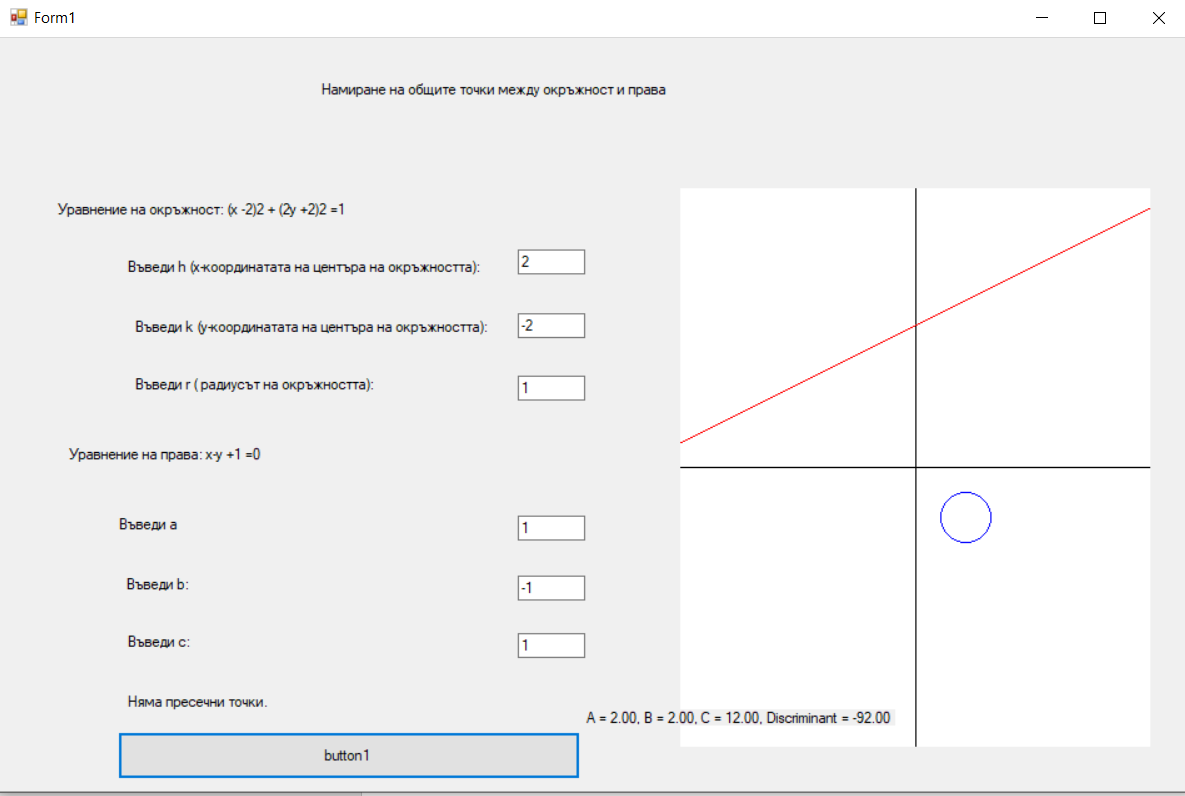
**За Б) (x − 1)2+(y + 1)2 = 10 и 3x + y + 8 = 0 - 2 екип:**

 **Окръжност:**

* В текстовото поле за h: **2**
* В текстовото поле за k: **-1**
* В текстовото поле за радиуса R: **2**

 **Права:**

* В текстовото поле за a: **3**
* В текстовото поле за b: **1**
* В текстовото поле за c: **8**



**За В) (x + 3)2+(y − 2)2 = 13 и x + y − 4 = 0 - 3 екип**

**Окръжност:**

* В текстовото поле за h: **-3**
* В текстовото поле за k: **2**
* В текстовото поле за радиуса R: **13≈3.61\sqrt{13} \approx 3.6113​≈3.61** (закръглете, ако програмата не поддържа дробни стойности)

**Права:**

* В текстовото поле за a: **1**
* В текстовото поле за b: **1**
* В текстовото поле за c: **-4**

