Am ales ca temă pentru Soft educațional capitolul Clasa Graphics în C#, dar elemente comune sunt și alte softuri:Java Script.

Aplicația este folosită pentru:

-elevii și profesorii de la clasă când se predă la Programarea Orientată pe Obiecte capitolul Graphics

-elementele din acest capitol sunt folosite la proiectare jocuri;

-elementele din capitol sunt folosite la proiecte educaționale, utilitare ; .

-elementele din capitol sunt folosite la prezentări complexe care au nevoie de baze de date;

-de anumite animații care nu se gasesc printre cele predefinite .

Ce aduce nou acest proiect?

-Abordează problema resurselor. În cazul acesta de memorie, poate înlocui anumite obiecte care consumă memorie.

-Se pot realiza proiecte care lasă utilizatorului libertatea de a inteveni în alegerea unei imagini, a unor informații (imagini,test) pentru teste, slide Show-uri

În felul acesta se pot refuza anumite violențe(ne referim la imaginile și animațiile unor jocuri), dar și stereotipii.

- pentru alegerea aplicațiilor am combinat (produs cartezian) elementele mulțimii instrumentelor informatice care fi folosite la această lecție ( elementele grafice de bază, animații, evenimente de taste, rotiri, traslații ) și elementele mulțimii evenimente din lumea reală care se pot modela cu Graphics ( lecții, teste, prezentări, jocuri).

Am structurat proiectul pe mai multe capitol:

Prezentare de desene ale unor figuri elementare, desenare contur, umplere cu hasura, culoare, gradient, imagine.

1. Desenarea unei linii

g = this.CreateGraphics();

g.Clear(Color.White);

var pen = new Pen(Color.Red, 3);

pen.DashStyle = DashStyle.DashDotDot;

int x1 = 100;

int y1 = 100;

int x2 = 400;

int y2 = 100;

g.DrawLine(pen, x1, y1, x2, y2);// - Cu ajutorul acestui cod am desenat o linie punctată. Mai întâi am ales culoarea, apoi stilul si coordonatele liniei.

g = this.CreateGraphics();

g.Clear(Color.White);

var pen = new Pen(Color.Red, 3);

pen.DashStyle = DashStyle.Dash;

int x1 = 100;

int y1 = 100;

int x2 = 400;

int y2 = 100;

g.DrawLine(pen, x1, y1, x2, y2); //- Acest cod a desenat tot o linie punctata, dar într-un stil diferit (Dash diferit de DashDotDot, care era o linie și două puncte), am ales culoarea și coordonatele.

g = this.CreateGraphics();

g.Clear(Color.White);

var pen = new Pen(Color.Red, 3);

pen.DashStyle = DashStyle.Dot;

int x1 = 100;

int y1 = 100;

int x2 = 400;

int y2 = 100;

g.DrawLine(pen, x1, y1, x2, y2);// - Acest cod a desenat o linie punctată.

g = this.CreateGraphics();

g.Clear(Color.White);

Pen redPen = new Pen(Brushes.Red, 6.0F);

redPen.StartCap = System.Drawing.Drawing2D.LineCap.RoundAnchor;

redPen.EndCap = System.Drawing.Drawing2D.LineCap.ArrowAnchor;

g.DrawLine(redPen, 80.0F, 40.0F, 300.0F, 185.0F);

//- Acest cod a desenat o săgeată, folosind System.Drawing.Drawing2D.LineCap.RoundAnchor la început și System.Drawing.Drawing2D.LineCap.ArrowAnchor la final.

g = this.CreateGraphics();

g.Clear(Color.White);

AdjustableArrowCap bigArrow = new AdjustableArrowCap(11, 11);

Pen p = new Pen(Color.Blue, 1);

p.CustomEndCap = bigArrow;

g.DrawLine(p, 60, 30, 320, 200); //- Acest cod a desenat tot o săgeată.

g = this.CreateGraphics();

g.Clear(Color.White);

Pen penita = new Pen(Color.Black, 3);

PointF punct1 = new PointF(200.0F, 100.0F);

PointF punct2 = new PointF(450.0F, 100.0F);

g.DrawLine(penita, punct1, punct2); - Acest cod a desenat o linie normală.

Am folosit și eventul MouseHover și MouseLeave la butoane ca să schimb culoarea butonului când mouse-ul este pe el.

2. Desenarea unui Dreptunghi (desen, umplere, gradient, imagine)

3. Desenarea unei Elipse (desen, umplere, gradient)

4. Desenarea unui Poligon

Graphics g;

Pen p = new Pen(Color.Black);

g = this.CreateGraphics();

g.Clear(Color.White);

PointF point1 = new PointF(150.0F, 70.0F);

PointF point2 = new PointF(190.0F, 75.0F);

PointF point3 = new PointF(250.0F, 50.0F);

PointF point4 = new PointF(330.0F, 100.0F);

PointF point5 = new PointF(370.0F, 150.0F);

PointF point6 = new PointF(420.0F, 250.0F);

PointF point7 = new PointF(250.0F, 300.0F);

PointF[] curvePoints =

{

point1,

point2,

point3,

point4,

point5,

point6,

point7

};

g.DrawPolygon(p, curvePoints); - Am ales punctele poligonului si brush-ul, apoi am folosit comanda g.DrawPolygon pentru a desena poligonul.

g = this.CreateGraphics();

g.Clear(Color.White);

PointF point1 = new PointF(150.0F, 70.0F);

PointF point2 = new PointF(190.0F, 75.0F);

PointF point3 = new PointF(250.0F, 50.0F);

PointF point4 = new PointF(330.0F, 100.0F);

PointF point5 = new PointF(370.0F, 150.0F);

PointF point6 = new PointF(420.0F, 250.0F);

PointF point7 = new PointF(250.0F, 300.0F);

PointF[] curvePoints =

{

point1,

point2,

point3,

point4,

point5,

point6,

point7

};

Image newImage = Image.FromFile("blue.jpg");

Bitmap bimage = new Bitmap(newImage);

TextureBrush tb = new TextureBrush(bimage);

g.FillPolygon(tb, curvePoints); - Am pus o imagine folosind Image newImage, Bitmap bimage si TextureBrush tb.

5. Desenarea cu Mouse

Utilizatorul realizează desene folosind mouse-ul.

Ausing System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace ProiectGraphics

{

public partial class desenmouse : Form

{

bool ok1, ok2, ok3,ok4,ok5,ok6;

int nr = 0,a,b,i=-1,start=0,start2=0;

Pen p = new Pen(Color.White, 2);

Graphics g;

ColorDialog MyDialog;

SolidBrush s,s1;

public bool ok,okey;

Point[] v = new Point[400];

public desenmouse()

{

InitializeComponent();

}

// Desenarea unei linii. La primul click se reține coordonata punctului care reprezintă un capăt al dreptei, iar la al doilea click se deseneaza dreapta.

private void desenmouse\_MouseClick(object sender, MouseEventArgs e)

{

start=1;

int x = int.Parse(numericUpDown1.Value.ToString());

int y = int.Parse(numericUpDown2.Value.ToString());

g=this.CreateGraphics();

nr++;

if (s != null)

s = new SolidBrush(MyDialog.Color);

else

s = new SolidBrush(Color.Black);

if (ok1)

{

if (nr % 2 == 1)

{ a = e.X; b = e.Y; }

else

{

g.DrawLine(p, a, b, e.X, e.Y);

}

}

//desenerarea unui dreptunghi.

//Din controalele NumericUpDown1,NumericUpDown1 se selectează lungimile laturilor dreptunghiului. La un click se desenează dreptunghiul în punctul în care s-a dat click pe formă.

//poate fi folosit pentru a desena mozaicuri.

if (ok2)

{

{ a = e.X; b = e.Y; }

Rectangle r = new Rectangle(a, b, x,y);

// g.DrawRectangle(p, r);

g.FillRectangle(s, r);

}

//desenare elipse. Acelasi principiu ca la dreptunghi

if(ok3)

{

{ a = e.X; b = e.Y; }

Rectangle r = new Rectangle(a, b, 100, 100);

g.DrawEllipse(p, r);

g.FillEllipse(s, r);

}

}

private void radioButton1\_CheckedChanged(object sender, EventArgs e)

{

if (radioButton1.Checked == true)

ok1 = true;

else

ok1 = false;

}

private void radioButton2\_CheckedChanged(object sender, EventArgs e)

{

if (radioButton2.Checked == true)

ok2 = true;

else

ok2 = false;

}

private void radioButton3\_CheckedChanged(object sender, EventArgs e)

{

if (radioButton3.Checked == true)

ok3 = true;

else

ok3 = false;

}

//alegere culoare folosind obiectul ColorDialog

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

{

MyDialog = new ColorDialog();

MyDialog.AllowFullOpen = false;

MyDialog.ShowHelp = true;

if (MyDialog.ShowDialog() == DialogResult.OK) { s = new SolidBrush(MyDialog.Color); }

}

private void radioButton4\_CheckedChanged(object sender, EventArgs e)

{

if (radioButton4.Checked == true)

ok4 = true;

else

ok4 = false;

}

// Desenare poligon. Cu ajutorul evenimentului click pe butonul stâng al mouse-ului se desenează punctele poligonului, în timp ce ajutorul evenimentului click pe butonul drept se umple poligonul cu o culoare.

private void desenmouse\_MouseDown(object sender, MouseEventArgs e)

{

int x = int.Parse(numericUpDown1.Value.ToString());

int y = int.Parse(numericUpDown2.Value.ToString());

if (ok4)

{

switch (MouseButtons)

{

case MouseButtons.Left:

{

i++;

v[i] = new Point(int.Parse(e.X.ToString()), int.Parse(e.Y.ToString()));

g = this.CreateGraphics();

s = new SolidBrush(MyDialog.Color);

Rectangle r = new Rectangle(e.X, e.Y, x, y);

g.FillEllipse(s, r); i++;

if (i == 0)

{

a = int.Parse(e.X.ToString());

b = int.Parse(e.Y.ToString());

}

break;

}

case MouseButtons.Right:

{

g = this.CreateGraphics();

s = new SolidBrush(MyDialog.Color);

for (int j = i + 1; j < v.Count(); j++)

v[j] = new Point(a, b);

g.FillPolygon(s, v);

i = -1;

break;

}

}

}

if(ok5)

switch (MouseButtons)

{

case MouseButtons.Left: { ok = false; ; break; }

case MouseButtons.Right: { ok = true; break; }

}

}

private void desenmouse\_MouseMove(object sender, MouseEventArgs e)

{

// Am realizat o gumă. Desenează un dreptunghi având culoarea albă. Pornitul și închiderea ștergerii se realizează cu ajutorul unui buton.

if (ok6)

{

if (start2 == 1)

{

g = this.CreateGraphics();

Rectangle r = new Rectangle(e.X, e.Y, 7, 7);

SolidBrush s2 = new SolidBrush(Color.White);

g.FillRectangle(s2, r);

}

// Se deseneaza un poligon. Se da click pe forma cu ajutorul evenimentului MouseMove se deseneaza punctele unui poligon. Apasam butonul dreapta al mouse-ului si se umple poligonul cu o culoare

}

if (ok5)

{

if (start == 1)

{

a = e.X;

b = e.Y;

if (ok == false)

{

i++; ;

g = this.CreateGraphics();

Rectangle r = new Rectangle(e.X, e.Y, 3, 3);

s = new SolidBrush(MyDialog.Color);

g.FillEllipse(s, r);

v[i] = new Point(int.Parse(e.X.ToString()), int.Parse(e.Y.ToString()));

}

else

{

i++;

v[i] = new Point(a, b);

for (int j = i + 1; j < v.Count(); j++)

v[j] = new Point(a, b);

g = this.CreateGraphics();

s = new SolidBrush(MyDialog.Color);

g.FillPolygon(s, v);

i = -1;

}

}

}

}

private void radioButton5\_CheckedChanged(object sender, EventArgs e)

{

if (radioButton5.Checked == true)

ok5 = true;

else

ok5 = false;

}

private void radioButton6\_CheckedChanged(object sender, EventArgs e)

{

if (radioButton6.Checked == true)

ok6 = true;

else

ok6 = false;

}

private void button2\_Click(object sender, EventArgs e)

{

if (start2 == 0)

start2 = 1;

else

start2 = 0;

}

}

}

Graficele unor funcții matematice

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace graphics

{

public partial class grafice\_matematice : Form

{

public grafice\_matematice()

{

InitializeComponent();

}

//functie polinomiala frad 3

/\* private float pol(float x)

{//X^3-2X^2-X+2

return x \* x \* x - 2 \* x \* x - x + 2;

}

private float pol(float x)

{

// return float.Parse(Math.Log(

}\*/

private float d1(float x)

{//derivata 1

return float.Parse(textBox3.Text)\* x \* x +float.Parse(textBox4.Text)\* 4 \* x + float.Parse(textBox5.Text);

}

private float d2(float x)

{//derivata 2

// return int.Parse(textBox1.Text.Trim()) \* x +int.Parse(textBox2.Text.Trim());

return float.Parse(textBox1.Text) \* x + float.Parse(textBox2.Text);

// return 3 \* x - 6;

}

private float radical(float x)

{//derivata 2

//return (float) Math.Pow(x,(float)0.5);

return (float)Math.Sqrt((float)x);

}

private float exponentiala(float x)

{//derivata 2

//return (float) Math.Pow(x,(float)0.5);

return (float)Math.Exp((float)x);

}

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

{

Graphics G = pictureBox1.CreateGraphics();

G.Clear(Color.White);

int x = pictureBox1.Width / 2;

int y = pictureBox1.Height / 2;

//grid

Pen pg = new Pen(Color.LightGray);

for (int i = -50; i <= 50; i++)

G.DrawLine(pg, x - 30 \* i, 0, x - 30 \* i, pictureBox1.Height);

for (int j = -50; j <= 50; j++)

G.DrawLine(pg, 0, y - 30 \* j, pictureBox1.Width, y - 30 \* j);

//axele

Pen pn = new Pen(Color.Black);

G.DrawLine(pn, 0, y, pictureBox1.Width, y);

G.DrawLine(pn, x, 0, x, pictureBox1.Height);

//graficul polinomului

Pen pr = new Pen(Color.Red);

/\* for (float i = -3; i <= 3; i = i + (float)0.001)

{

G.DrawEllipse(pr, x + i \* 30, y - 30 \* pol(i), 1, 1);

}

\*/

//graficul functiei de grad 2

Pen pv = new Pen(Color.Green);

for (float i = -3; i <= 3; i = i + (float)0.001)

{

G.DrawEllipse(pv, x + i \* 30, y - 30 \* d2(i), 1, 1);

}

// grafic functie liniara

Pen pa = new Pen(Color.Blue);

for (float i = -4; i <= 3; i = i + (float)0.001)

{

G.DrawEllipse(pa, x + i \* 30, y - 30 \* d2(i), 1, 1);

}

}

private void button2\_Click(object sender, EventArgs e)

{

Graphics G = pictureBox2.CreateGraphics();

G.Clear(Color.White);

int x = pictureBox2.Width / 2;

int y = pictureBox2.Height / 2;

//grid

Pen pg = new Pen(Color.LightGray);

for (int i = -50; i <= 50; i++)

G.DrawLine(pg, x - 30 \* i, 0, x - 30 \* i, pictureBox2.Height);

for (int j = -50; j <= 50; j++)

G.DrawLine(pg, 0, y - 30 \* j, pictureBox2.Width, y - 30 \* j);

//axele

Pen pn = new Pen(Color.Black);

G.DrawLine(pn, 0, y, pictureBox2.Width, y);

G.DrawLine(pn, x, 0, x, pictureBox2.Height);

//graficul polinomului

Pen pr = new Pen(Color.Red);

for (float i = 0; i <= 5; i = i + (float)0.001)

{

G.DrawEllipse(pr, x + i \* 30, y - 30 \* radical(i), 1, 1);

}

}

private void button3\_Click(object sender, EventArgs e)

{

Graphics G = pictureBox3.CreateGraphics();

G.Clear(Color.White);

int x = pictureBox3.Width / 2;

int y = pictureBox3.Height / 2;

//grid

Pen pg = new Pen(Color.LightGray);

for (int i = -50; i <= 50; i++)

G.DrawLine(pg, x - 30 \* i, 0, x - 30 \* i, pictureBox3.Height);

for (int j = -50; j <= 50; j++)

G.DrawLine(pg, 0, y - 30 \* j, pictureBox3.Width, y - 30 \* j);

//axele

Pen pn = new Pen(Color.Black);

G.DrawLine(pn, 0, y, pictureBox3.Width, y);

G.DrawLine(pn, x, 0, x, pictureBox3.Height);

//graficul polinomului

Pen pr = new Pen(Color.Red);

for (float i = -3; i <= 5; i = i + (float)0.001)

{

G.DrawEllipse(pr, x + i \* 30, y - 30 \* exponentiala(i), 1, 1);

}

}

private void button4\_Click(object sender, EventArgs e)

{

Graphics G = pictureBox4.CreateGraphics();

G.Clear(Color.White);

int x = pictureBox4.Width / 2;

int y = pictureBox4.Height / 2;

//grid

Pen pg = new Pen(Color.LightGray);

for (int i = -50; i <= 50; i++)

G.DrawLine(pg, x - 30 \* i, 0, x - 30 \* i, pictureBox2.Height);

for (int j = -50; j <= 50; j++)

G.DrawLine(pg, 0, y - 30 \* j, pictureBox4.Width, y - 30 \* j);

//axele

Pen pn = new Pen(Color.Black);

G.DrawLine(pn, 0, y, pictureBox2.Width, y);

G.DrawLine(pn, x, 0, x, pictureBox4.Height);

//graficul polinomului

Pen pr = new Pen(Color.Red);

for (float i = -5; i <= 5; i = i + (float)0.001)

{

G.DrawEllipse(pr, x + i \* 30, y - 30 \* d1(i), 1, 1);

}

}

}

}

II. Prezentare de rotații si translații

1. Rotație cu Translatie

Am adus pe formă un buton și un timer. Butonul pornește timer-ul ,care execută setul de instrucțiuni care schimbă poziția și unghiul unui obiect rectangle.

public partial class rotatie\_translatie : Form

{

int unghi = 0, xa = 220, ya = 50, xb, yb;

public rotatie\_translatie()

{

InitializeComponent();

}

private void timer1\_Tick(object sender, EventArgs e)

{

xa++;

ya++;

unghi = unghi + 1;

Invalidate();

}

private void rotatie\_translatie\_Paint(object sender, PaintEventArgs e)

{

Graphics g = this.CreateGraphics();

Pen blackPen = new Pen(Color.Black, 3);

g.TranslateTransform(xa, ya);

Rectangle rect = new Rectangle(-xa / 2, -ya / 2, 100, 100);

g.RotateTransform(unghi);

g.DrawRectangle(blackPen, rect);

}

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

{

timer1.Start();

}

2. Desenare Soare

Am adus pe formă un buton. La evenimentul click se desenează un cerc. Se mai desenează o linie, și cu ajutorul metodei RotateTransform desenam celelalte raze, fiecare rotiță față de cealaltă cu un unghi.

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

{

SolidBrush s = new SolidBrush(Color.Yellow);

Pen p = new Pen(Color.Yellow, 2);

Graphics g = this.CreateGraphics();

Rectangle r = new Rectangle(250, 200, 100, 100);

g.DrawEllipse(p, r);

g.FillEllipse(s, r);

g.TranslateTransform(300.0F, 250.0F);

for (int i = 1; i <= 36; i++)

{

g.DrawLine(p, new Point(0, 50), new Point(0, 200));

g.RotateTransform(10.0F);

}

3. Rotație Dreptunghi

Am adus pe formă trei butoane și un timer. Primul buton desenează mai multe dreptunghiuri în jurul unui punct, al doilea pornește timer-ul care schimbă unghiul unui obiect rectangle.

Al treilea buton genereaza patru dreptunghiuri in jurul unui cerc.

int unghi = 0;

SolidBrush s;

public rotatie\_imagine()

{

InitializeComponent();

}

private void rotatie\_imagine\_Paint(object sender, PaintEventArgs e)

{

Graphics g = this.CreateGraphics();

Pen blackPen = new Pen(Color.Black, 3);

g.TranslateTransform(300, 300);

Rectangle rect = new Rectangle(-100, -25, 200, 50);

g.RotateTransform(unghi);

g.DrawRectangle(blackPen, rect);

}

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

{

Graphics g = this.CreateGraphics();

g.Clear(Color.White);

Pen blackPen = new Pen(Color.Black, 3);

g.TranslateTransform(300, 300);

Rectangle rect = new Rectangle(0, 0, 200, 50);

for (int i = 1; i <= 12; i++)

{

g.DrawRectangle(blackPen, rect);

g.RotateTransform(30);

}

}

private void button2\_Click(object sender, EventArgs e)

{

timer1.Start();

}

private void timer1\_Tick(object sender, EventArgs e)

{

unghi = unghi + 1;

Invalidate();

}

private void button3\_Click(object sender, EventArgs e)

{

Graphics g = this.CreateGraphics();

Pen blackPen = new Pen(Color.Black, 3);

s = new SolidBrush(Color.LightPink);

SolidBrush s1 = new SolidBrush(Color.Orchid);

g.TranslateTransform(300, 300);

Rectangle rect = new Rectangle(-100, -25, 200, 50);

Rectangle rect1 = new Rectangle(-25, -25, 50, 50);

Rectangle rect2 = new Rectangle(-25, -100, 50, 200);

g.RotateTransform(unghi);

g.FillRectangle(s, rect);

g.FillRectangle(s, rect2); g.FillEllipse(s1, rect1);

}

4. Mișcarea unei Imagini

Am adus pe formă un buton și un timer. Butonul pornește timer-ul, care mișcă o imagine mai mică spre una mai mare până ce acestea se intersectează, dupa care se oprește.

Am folosit event-ul paint ca să desenez cele două imagini.

public partial class miscare\_imagine : Form

{

Image newImage = Image.FromFile("blue.jpg");

Rectangle destRect1 = new Rectangle(60, 70, 40, 40);

Image newImage1 = Image.FromFile("blue.jpg");

Rectangle destRect2 = new Rectangle(240, 190, 220, 220);

public miscare\_imagine()

{

InitializeComponent();

}

private void timer1\_Tick(object sender, EventArgs e)

{

destRect1.X += 3;

destRect1.Y += 3;

if (destRect1.IntersectsWith(destRect2)) timer1.Stop();

Invalidate();

}

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

{

timer1.Start();

}

private void miscare\_imagine\_Paint(object sender, PaintEventArgs e)

{

Graphics g = e.Graphics;

g.DrawImage(newImage, destRect1);

g.DrawImage(newImage1, destRect2);

}

5. Desenare Floare

Am adus pe formă un buton care desenează o floare folosindu-se de g.DrawBezier. Desenăm o petală folosind curbe bezier, și după aceea desenăm celelalte petale folosind metoda TrasformRatate.

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

{

Graphics g = this.CreateGraphics();

Pen blackPen = new Pen(Color.Black, 1);

Point start = new Point(0, 0);

Point control1 = new Point(100, 50);

Point control2 = new Point(200, 0);

Point end = new Point(0, 0);

g.TranslateTransform(280.0F, 220.0F);

for (int i = 1; i <= 12; i++)

{

g.DrawBezier(blackPen, start, control1, control2, end);

g.RotateTransform(30.0F);

}

6. Rotiri String.

// Desenăm un string si îl rotim cu Transform Rotate. La toate aplicațiile unde am folosit rotiri, am folosit și TransformTranslate pentru a detemina punctul în jurul căruia se face rotire, altfel ar rămâne punctul o(0,0) originea, colțul din stanga sus al ecranului.

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

{

SolidBrush s = new SolidBrush(Color.Yellow);

Pen p = new Pen(Color.Yellow, 2);

Graphics g = this.CreateGraphics();

Rectangle r = new Rectangle(150, 150, 100, 100);

// g.DrawEllipse(p, r);

// g.FillEllipse(s, r);

g.TranslateTransform(200.0F, 200.0F);

for (int i = 1; i <= 18; i++)

{

// g=this.CreateGraphics();

String drawString = "\*\*\*\*\*\*\*\*\*\*";

Font drawFont = new Font("Arial", 12);

SolidBrush drawBrush = new SolidBrush(Color.Black);

float x = 0.0F;

float y = 0.0F;

g.DrawString(drawString, drawFont, drawBrush, x, y);

// g.DrawLine(p, new Point(0, 50), new Point(0, 200));

g.RotateTransform(20.0F);

// g.Clear(Color.White);

Thread.Sleep(100);

}

}

// Este vorba de lista circulară alocată dinamic. Se desenează un nod și arcul care face legatura cu nodul următor și cu transformRotate se desenează celelalte noduri.

7. Lista Circulară

Graphics g;

int n;

int[] v = new int[100];

string s;

public lista\_circulara()

{

InitializeComponent();

}

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

{

if (openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

s = openFileDialog1.InitialDirectory + openFileDialog1.FileName;

}

using (StreamReader fin = new StreamReader(s))

{

string s1 = fin.ReadLine();

n = int.Parse(s1);

string s2 = fin.ReadLine();

string[] vs = s2.Split(' ');

for (int i = 0; i < vs.Count(); i++)

v[i] = int.Parse(vs[i].ToString());

fin.Close();

}

for (int i = 1; i <= n; i++) MessageBox.Show(v[i].ToString());

g = this.CreateGraphics();

g.TranslateTransform(300, 300);

int raza = 150;

for (int i = 1; i <= 12; i++)

{

//desen(100);

AdjustableArrowCap bigArrow = new AdjustableArrowCap(4, 4);

Pen p = new Pen(Color.Red, 2);

Rectangle r1 = new Rectangle(-30, raza + 30, 30, 30);

Rectangle r2 = new Rectangle(0, raza + 30, 30, 30);

g.DrawRectangle(p, r1);

g.DrawRectangle(p, r2);

g.RotateTransform(30.0F);

Pen penarc = new Pen(Color.Red, 2);

penarc.CustomEndCap = bigArrow;

float x = -195.0F;

float y = -195.0F;

float width = 390.0F;

float height = 390.0F;

float startAngle = 5.0F;

float sweepAngle = 16.0F;

g.DrawArc(penarc, x, y, width, height, startAngle, sweepAngle);

String drawString = v[i].ToString();

Font drawFont = new Font("Arial", 12);

SolidBrush drawBrush = new SolidBrush(Color.Black);

PointF drawPoint = new PointF(3, raza + 35);

g.DrawString(drawString, drawFont, drawBrush, drawPoint);

}

}

III. Jocuri

1. Deplasarea cu Taste

Un obiect pleacă dintr-un punct, se deplasează dirijat de taste. La fiecare ciclare a obiectului Timer se produce o deplasare in direcția indicată de tastele de deplasare.

Dacă întâlnește obstacol (rectangle3.Intersect(rectangle2)) revine la poziția de start, dacă intersectează un obiect ( elipsa) care reprezintă punctual, jocul se oprește.

public partial class rotatie\_taste : Form

{

bool r, l, sus, jos, r1, l1, sus1, jos1;

Random rand = new Random();

int i = 100, j = 100, i1 = 700, j1 = 300, i2 = 430, j2 = 200;

int score1 = 0, score2 = 0;

Pen p;

Graphics g;

SolidBrush br = new SolidBrush(Color.Red);

public rotatie\_taste()

{

InitializeComponent();

}

private void timer1\_Tick(object sender, EventArgs e)

{

g = this.CreateGraphics();

g.Clear(Color.White);

if (r == true) { i += 5; }

if (l == true) { i -= 5; }

if (sus == true) { j += 5; }

if (jos == true) { j -= 5; }

if (r1 == true) { i1 += 5; }

if (l1 == true) { i1 -= 5; }

if (sus1 == true) { j1 += 5; }

if (jos1 == true) { j1 -= 5; }

p = new Pen(Color.Black, 2);

Rectangle rectangle1 = new Rectangle(i, j, 100, 100);

Rectangle rectangle2 = new Rectangle(i1, j1, 100, 100);

Rectangle rectangle3 = new Rectangle(i2, j2, 50, 50);

g.DrawRectangle(Pens.Black, rectangle1);

g.DrawRectangle(Pens.Red, rectangle2);

g.DrawRectangle(Pens.Red, rectangle3);

g.FillRectangle(br, rectangle3);

if (rectangle2.IntersectsWith(rectangle3))

{

rectangle3.Intersect(rectangle2);

if (!rectangle2.IsEmpty)

{

i1 = 700; j1 = 300;

score2++;

label6.Text = score2.ToString();

}

}

if (rectangle1.IntersectsWith(rectangle3))

{

rectangle3.Intersect(rectangle1);

if (!rectangle1.IsEmpty)

{

i = 100; j = 100;

score1++;

label5.Text = score1.ToString();

}

}

}

private void rotatie\_taste\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Right) { r = true; }

if (e.KeyCode == Keys.Left) { l = true; }

if (e.KeyCode == Keys.Down) { sus = true; }

if (e.KeyCode == Keys.Up) { jos = true; }

if (e.KeyCode == Keys.D) { r1 = true; }

if (e.KeyCode == Keys.A) { l1 = true; }

if (e.KeyCode == Keys.S) { sus1 = true; }

if (e.KeyCode == Keys.W) { jos1 = true; }

}

private void rotatie\_taste\_KeyUp(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Right) { r = false; }

if (e.KeyCode == Keys.Left) { l = false; }

if (e.KeyCode == Keys.Down) { sus = false; }

if (e.KeyCode == Keys.Up) { jos = false; }

if (e.KeyCode == Keys.D) { r1 = false; }

if (e.KeyCode == Keys.A) { l1 = false; }

if (e.KeyCode == Keys.S) { sus1 = false; }

if (e.KeyCode == Keys.W) { jos1 = false; }

}

private void rotatie\_taste\_Load(object sender, EventArgs e)

{

timer1.Start();

}

2. X și 0 generalizat. ( Este cunoscutul X și 0), dar cu dimensiunile mărite. )

Folosim o matrice de rectangle (butoanele sunt niște figuri geometrice) și o matrice cu elemente întregi. Înlocuim obiectele (button) ca să economisim memorie, dar putem obține și interfețe mai ușoare.

La click pe butoane matrice de numere intregi se modifica 1 pentru jucatorul 1 si 2 pentru jucatorul 2. Matricea este de dimensiuni 6X6 și un jucător trebuie să selecteze o secvență de 4 butoane consecutive pe linie sau coloană sau pe diagonal (paralele cu diagonala principală sau cu diagonala secundară). Cu ajutorul lecției de secvențe de la clasa a IX-a se verifică dacă există acea secvență de 4 elemente pe linie/coloană/diagonale. Jocul poate fi extins.

int[,] a1 = new int[10, 10];

Rectangle[,] a = new Rectangle[10, 10];

int[] vf = new int[22];

int[] v = new int[17];

int nr = 0;

Color[] c = new Color[10];

Random r = new Random();

int[,] ok = new int[100, 100];

Graphics g;

int l;

public Form1()

{

InitializeComponent();

}

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

{

int i, j;

for (i = 0; i < 6; i++)

for (j = 0; j < 6; j++)

{

a[i, j] = new Rectangle();

a[i, j].Height = 40;

a[i, j].Width = 40;

a[i, j].Location = new Point(100 + i \* 41, 100 + j \* 41);

SolidBrush s = new SolidBrush(Color.Blue);

g = this.CreateGraphics();

g.FillRectangle(s, a[i, j]);

}

}

private void Form1\_MouseClick(object sender, MouseEventArgs e)

{

int i1, j1;

g = this.CreateGraphics();

SolidBrush s2 = new SolidBrush(Color.Green);

SolidBrush s3 = new SolidBrush(Color.Orange);

i1 = (e.X - 100) / 41;

j1 = (e.Y - 100) / 41;

if (i1 >= 0 && i1 < 6 && j1 >= 0 && j1 < 6)

{

nr++;

if (nr % 2 == 0)

{

if (a1[i1, j1] == 0)

{

a1[i1, j1] = 1;

g.FillRectangle(s3, a[i1, j1]);

String drawString = "x";

Font drawFont = new Font("Arial", 11);

SolidBrush drawBrush = new SolidBrush(Color.Black);

PointF drawPoint = new PointF(100 + i1 \* 41 + 10, 100 + j1 \* 41 + 10);

g.DrawString(drawString, drawFont, drawBrush, drawPoint);

//((Button)sender).Text = "x";

}

}

else

{

if (a1[i1, j1] == 0)

{

g.FillRectangle(s2, a[i1, j1]);

String drawString = "o";

Font drawFont = new Font("Arial", 11);

SolidBrush drawBrush = new SolidBrush(Color.Black);

PointF drawPoint = new PointF(100 + i1 \* 41 + 10, 100 + j1 \* 41 + 10);

g.DrawString(drawString, drawFont, drawBrush, drawPoint);

//((Button)sender).Text = "x";

// ((Button)sender).Text = "o";

a1[i1, j1] = 2;

}

}

}

int i, j;

for (i = 0; i < 6; i++)

{

l = 0;

// int nrx = 0, nro = 0;

for (j = 0; j < 6; j++)

if (a1[i, j] == 1) { l++; }

else

{

if (l >= 4) MessageBox.Show("A castigat X!");

l = 0;

}

for (i = 0; i < 6; i++)

{ l = 0;

for (j = 0; j < 6; j++)

if (a1[i, j] == 2) l++;

else

{

if (l >= 4) MessageBox.Show("A castigat 0!");

l = 0;

}

}

for (j = 0; j < 6; j++)

{

l = 0;

// int nrx = 0, nro = 0;

for (i = 0; i < 6; i++)

if (a1[i, j] == 1) { l++; }

else

{

if (l >= 4) MessageBox.Show("A castigat X!");

l = 0;

}

for (j = 0; j < 6; j++)

{ l = 0;

for (i = 0; i < 6; i++)

if (a1[i, j] == 2) l++;

else

{

// if (l != 0)// MessageBox.Show(l.ToString());

if (l >= 4) MessageBox.Show("A castigat 0!");

l = 0;

}

}

// deasupra princ

for (i = 0; i < 6; i++)

{

l = 0;

for (j = 0; j <= i; j++)

if (a1[j, 6 - i + j] == 1) l++;

else

{

if (l >= 4) MessageBox.Show("A castigat X!");

l = 0;

}

for ( i = 0; i < 6; i++)

{ l = 0;

for (j = 0; j <= i; j++)

if (a1[j, 6 - i + j] == 2) l++;

else

{

if (l >= 4) MessageBox.Show("A castigat 0!");

l = 0;

}

}

//sub prin

for (i = 0; i < 6; i++)

{

l = 0;

for (j = 0; j <= i; j++)

if (a1[6 - i + j, j] == 1) l++;

else

{

if (l >= 4) MessageBox.Show("A castigat X!");

l = 0;

}

for (i = 0; i < 6; i++)

{ l = 0;

for (j = 0; j <= i; j++)

if (a1[6 - i + j, j] == 2) l++;

else

{

if (l >= 4) MessageBox.Show("A castigat 0!");

l = 0;

}

}

// secundare

for (i = 0; i < 6; i++)

{

l = 0;

for (j = 0; j <= i; j++)

if (a1[i - j, j] == 1) l++;

else

{

if (l >= 4) MessageBox.Show("A castigat X!");

l = 0;

}

for (i = 0; i < 6; i++)

{ l = 0;

for (j = 0; j <= i; j++)

if (a1[i - j, j] == 2) l++;

else

{

if (l >= 4) MessageBox.Show("A castigat 0!");

l = 0;

}

}

for (i = 0; i < 6; i++)

{

l = 0;

for (j = 0; j <= i; j++)

if (a1[6 - j, 6 - i + j] == 1) l++;

else

{

if (l >= 4) MessageBox.Show("A castigat X!");

l = 0;

}

for (i = 0; i < 6; i++)

{ I = 0;

for (j = 0; j <= i; j++)

if (a1[6 - j, 6 - i + j] == 2) l++;

else

{

if (l >= 4) MessageBox.Show("A castigat 0!");

l = 0;

}

}

}

3. Harta

Am desenat pe formă o hartă. Cu evenimentul MouseClick a salvat într-un fișier coordonatele capitalelor și numele lor. Se aleg randomizat cinci capitale și se desenează în dreptul lor un număr.

Din acestea se alege un număr și elevul trebuie să spună ce capitala este la acel numar selectând numele dintr-o lista existentă într-un obiect comboBox.

int nr = 0, n, no = 0, y, test = 0, nr\_itemi;

double nota = 1;

Class5[] v1 = new Class5[100];

Random r = new Random();

int[] vf = new int[100];

int[] v2 = new int[100];

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

if (string.Compare(comboBox1.Text, v1[v2[y]].capitala) == 0)

{ MessageBox.Show("Raspuns corect!"); nota += (double)9 / nr\_itemi; }

else

MessageBox.Show("Raspuns gresit!");

this.Invalidate();

}

Graphics g;

public harta()

{

InitializeComponent();

}

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

{

using (StreamReader fin = new StreamReader("TextFile1.txt"))

{

while (!fin.EndOfStream)

{

nr++;

string linie = fin.ReadLine();

string[] v = linie.Split(' ');

v1[nr] = new Class5();

v1[nr].x = int.Parse(v[0].ToString());

v1[nr].y = int.Parse(v[1].ToString());

v1[nr].capitala = v[2].ToString();

v1[nr].nr1 = nr;

}

fin.Close();

}

n = nr;

nr\_itemi = int.Parse(listBox1.Text);

MessageBox.Show(nr\_itemi.ToString());

}

private void button2\_Click(object sender, EventArgs e)

{

test++;

if (test <= nr\_itemi)

{

no = 1;

for (int i = 1; i <= 5; i++)

{

int ok = 0;

while (ok == 0)

{

int x = r.Next(1, n);

if (vf[x] == 0)

{

vf[x] = 1;

v2[i] = x;

ok = 1;

}

}

}

y = r.Next(1, 6);

label1.Text = " Ce capitala se gaseste la numarul " + y.ToString();

for (int i = 1; i <= 5; i++)

{

no = 1;

if (i <= 5)

{

g = this.CreateGraphics();

String drawString = i.ToString();

Font drawFont = new Font("Arial", 12);

SolidBrush drawBrush = new SolidBrush(Color.Black);

PointF drawPoint = new PointF(int.Parse(v1[v2[i]].x.ToString()) - 5, int.Parse(v1[v2[i]].y.ToString()) + 10);

g.DrawString(drawString, drawFont, drawBrush, drawPoint);

}

}

}

else MessageBox.Show(nota.ToString());

}

private void button3\_Click(object sender, EventArgs e)

{

this.Invalidate();

}

4. Împerecherea Culorilor

Noutatea este că matricea jocului este matrice de desene( drepunghiuri). Culorile sunt alese din mulțimea culorilor existente, nu o mulțime aleasă de programator.

O altă noutate este că jucatorul poate vedea un timp solutia finala.

int[,] a1 = new int[10, 10];

Rectangle[,] a = new Rectangle[10, 10];

int[] vf = new int[22];

int[] v = new int[17];

int nr1 = 0, np = 0, x1, x2, y1, y2, timp = 0;

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

{

int i, j, x, k = 0;

for (i = 0; i <= 16; i++) vf[i] = 0;

int nr = 8;

while (nr != 0)

{

x = r.Next(1, 9);

if (vf[x] < 2)

{

k++;

v[k] = x;

vf[x]++;

if (vf[x] == 2)

nr--;

}

}

k = 0;

for (i = 0; i < 4; i++)

for (j = 0; j < 4; j++)

{

k++;

a1[i, j] = v[k];

}

for (i = 1; i <= 8; i++)

c[i] = Color.FromArgb(r.Next(2, 255), r.Next(12, 255), r.Next(2, 255));

for (i = 0; i < 4; i++)

for (j = 0; j < 4; j++)

{

a[i, j] = new Rectangle();

a[i, j].Height = 40;

a[i, j].Width = 40;

a[i, j].Location = new Point(100 + i \* 41, 100 + j \* 41);

SolidBrush s = new SolidBrush(Color.Blue);

g = this.CreateGraphics();

g.FillRectangle(s, a[i, j]);

}

}

private void timer1\_Tick(object sender, EventArgs e)

{

if (timp == 1000)

{

timer1.Stop(); for (int i = 0; i < 4; i++)

for (int j = 0; j < 4; j++)

{

SolidBrush s5 = new SolidBrush(Color.Blue);

g = this.CreateGraphics();

g.FillRectangle(s5, a[i, j]);

}

// a[i, j].BackColor = Color.Blue;

timp = 0;

}

timp++;

}

private void button2\_Click(object sender, EventArgs e)

{

for (int i = 0; i < 4; i++)

for (int j = 0; j < 4; j++)

{

SolidBrush s4 = new SolidBrush(c[a1[i, j]]);

g = this.CreateGraphics();

g.FillRectangle(s4, a[i, j]);

}

// a[i, j].BackColor = c[a1[i, j]];

timer1.Start();

}

private void imperechere\_culori\_MouseClick(object sender, MouseEventArgs e)

{

int i1, j1;

i1 = (e.X - 100) / 41;

j1 = (e.Y - 100) / 41;

// if (i1 >= 0 && i1 < 4 && j1 >= 0 && j1 < 4)

// {

np++;

if (np % 2 == 0)

{

x1 = i1;

y1 = j1;

if (a1[x1, y1] == a1[x2, y2])

{

SolidBrush s3 = new SolidBrush(Color.Aqua);

g.FillRectangle(s3, a[x1, y1]);

g.FillRectangle(s3, a[x2, y2]);

ok[x1, y1] = 1; ok[x2, y2] = 1;

if (bun() == 1) MessageBox.Show("Ai reusit!");

}

}

else

{

x2 = i1;

y2 = j1;

if (a1[x1, y1] == a1[x2, y2] && x1 != 0)

{

// a[x1, y1].Text = b1[x1, y1].ToString();

// a[x2, y2].Text = b1[x2, y2].ToString();

ok[x1, y1] = 1; ok[x2, y2] = 1;

if (bun() == 1) MessageBox.Show("Ai reusit!");

}

}

}

Color[] c = new Color[10];

Random r = new Random();

int[,] ok = new int[100, 100];

Graphics g;

public imperechere\_culori()

{

InitializeComponent();

}

int bun()

{

int i1, j1, ok1 = 1;

for (i1 = 0; i1 < 4; i1++)

for (j1 = 0; j1 < 4; j1++)

if (ok[i1, j1] == 0)

ok1 = 0;

return ok1;

}

IV. Animații

1. Animație String

Am adus pe formă trei butoane, fiecare aduce pe formă câte o aplicație unde este desenat un tip diferit de string și o anumită animație.

Primul buton duce la o formă unde este desenat un string normal.

int nr = 0;

string s = "informatica si matematica";

Graphics g;

// animație cu apariția succesivă a câte unui caracter în plus

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

{

g = this.CreateGraphics();

g.Clear(Color.White);

timer1.Start();

}

private void timer1\_Tick(object sender, EventArgs e)

{

nr++;

int l = s.Length;

label3.Text = s.Substring(0, nr % l);

}

// apar succesiv trei stringuri și al patrulea se deplasează

int nr = 0;

Graphics g;

string s1 = "OLIMPIADA";

string s2 = "DE INOVARE SI CREATIVITATE ";

string s3 = "DIGITALA (INFOEDUCATIA)";

string s4 = "TEMA proiect CLASA GRAPHICS";

int nr2;

float x;

Graphics g1;

private void button2\_Click(object sender, EventArgs e)

{

timer1.Start();

}

private void timer1\_Tick(object sender, EventArgs e)

{

g = this.CreateGraphics();

g.Clear(Color.White);

nr++;

if (nr % 4 == 1)

{

Font drawFont = new Font("Arial", 16);

SolidBrush drawBrush = new SolidBrush(Color.Black);

float x = 150.0F;

float y = 50.0F;

g.DrawString(s1, drawFont, drawBrush, x, y);

}

else if (nr % 4 == 2)

{

Font drawFont = new Font("Arial", 16);

SolidBrush drawBrush = new SolidBrush(Color.Black);

float x = 160.0F;

float y = 80.0F;

g.DrawString(s2, drawFont, drawBrush, x, y);

}

else

if (nr % 4 == 3)

{

Font drawFont = new Font("Arial", 16);

SolidBrush drawBrush = new SolidBrush(Color.Black);

float x = 170.0F;

float y = 110.0F;

g.DrawString(s3, drawFont, drawBrush, x, y);

}

else

{

timer1.Stop();

nr2 = 0;

x = 500;

timer2.Start();

}

}

private void timer2\_Tick(object sender, EventArgs e)

{

g1 = this.CreateGraphics();

if (nr2 == 60)

{

Thread.Sleep(2000);

timer1.Stop();

g1.Clear(Color.White);

}

else

{

g1.Clear(Color.White);

Font drawFont = new Font("Arial", 16);

SolidBrush drawBrush = new SolidBrush(Color.Black);

g.DrawString(s1, drawFont, drawBrush, 150, 50);

g.DrawString(s2, drawFont, drawBrush, 160, 80);

g.DrawString(s3, drawFont, drawBrush, 170, 110);

nr2++;

x = 100.0F - nr2 \* 5;

float y = 150.0F;

g1.DrawString(s4, drawFont, drawBrush, x, y);

}

}

Al treilea buton duce la un string care schimbă culorile.

private void button3\_Click(object sender, EventArgs e)

{

timer1.Start();

}

private void timer1\_Tick(object sender, EventArgs e)

{

g = this.CreateGraphics();

g.Clear(Color.White);

String drawString = "VALENII DE MUNTE, JUD PRAHOVA";

Font drawFont = new Font("Arial", 16);

SolidBrush drawBrush = new SolidBrush(System.Drawing.Color.FromArgb(r.Next(256), r.Next(256), r.Next(256)));

float x = 20.0F;

float y = 150.0F;

g.DrawString(drawString, drawFont, drawBrush, x, y);

}

2. Slideshow

Apar succesiv imagini și date existente din fișier text la un interval de timp. Imaginea apare într-o formă poligonală, mai placută decat cele existente în prezentările obișnuite.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.IO;

namespace ProiectGraphics

{

public partial class slideshow : Form

{

PointF[] v2 = new PointF[100];

Class3[] v = new Class3[100];

int k = 0, i = 0, n, m;

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

{

using (StreamReader fin = new StreamReader("C:\\Users\\Wind 10\\source\\repos\\ProiectGraphics\\bin\\Debug\\netcoreapp3.1\\textslide.txt"))

{

while (!fin.EndOfStream)

{

i++;

string s = fin.ReadLine();

string[] v = s.Split(' ');

v2[i] = new PointF(int.Parse(v[0].ToString()), int.Parse(v[1].ToString()));

}

fin.Close(); m = i;

}

using (StreamReader fin = new StreamReader("C:\\Users\\Wind 10\\source\\repos\\ProiectGraphics\\bin\\Debug\\netcoreapp3.1\\slide.txt"))

{

while (!fin.EndOfStream)

{

string linie = fin.ReadLine();

string[] v1 = linie.Split('/');

v[k] = new Class3();

v[k].denumire = v1[0].ToString();

v[k].regiune = v1[1].ToString();

v[k].poza = v1[2].ToString(); k++;

}

n = k;

fin.Close();

}

k = 1; timer1.Start(); i = 0;

}

private void label2\_Click(object sender, EventArgs e)

{

this.Close();

meniu f = new meniu();

f.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

this.Hide();

cod\_slideshow f = new cod\_slideshow();

f.Show();

}

private void button2\_MouseHover(object sender, EventArgs e)

{

button2.BackColor = Color.DarkViolet;

}

private void button2\_MouseLeave(object sender, EventArgs e)

{

button2.BackColor = Color.DarkOrchid;

}

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

{

button1.BackColor = Color.DarkViolet;

}

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

{

button1.BackColor = Color.DarkOrchid;

}

Graphics g;

public slideshow()

{

InitializeComponent();

}

private void timer1\_Tick(object sender, EventArgs e)

{

i++;

g = this.CreateGraphics(); g.Clear(Color.White);

String drawString = v[i % n].denumire.ToString();

Font drawFont = new Font("Arial", 16);

SolidBrush drawBrush = new SolidBrush(Color.Black);

float x = 400.0F;

float y = 70.0F;

g.DrawString(drawString, drawFont, drawBrush, x, y);

String drawString2 = v[i % n].regiune.ToString();

Font drawFont2 = new Font("Arial", 16);

SolidBrush drawBrush2 = new SolidBrush(Color.Blue);

float x2 = 400.0F;

float y2 = 150.0F;

g.DrawString(drawString2, drawFont2, drawBrush2, x2, y2);

Image newImage = Image.FromFile(v[i % n].poza.ToString());

Bitmap bimage = new Bitmap(newImage);

TextureBrush tb = new TextureBrush(bimage);

//SolidBrush s = new SolidBrush(Color.Red);

//g.FillPolygon(s, v2);

g.FillPolygon(tb, v2);

}

}

}

3. Mărire Imagine

Am adus un buton care generează doua dreptunghiuri, unul cu culoare randomizată, altul cu o imagine. Acestea vor crește până se intersectează, dupa care programul se va restarta, și culoarea dreptunghiului se va schimba.

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

{

timer2.Start();

}

private void timer2\_Tick(object sender, EventArgs e)

{

Invalidate();

x = (x + 1) % 100;

}

private void marire\_Paint(object sender, PaintEventArgs e)

{

g = this.CreateGraphics();

if (x % 50 == 0)

s = new SolidBrush(System.Drawing.Color.FromArgb(r.Next(256), r.Next(256), r.Next(256)));

Bitmap bimage = new Bitmap(newImage);

TextureBrush tb = new TextureBrush(bimage);

g.TranslateTransform(300.0F, 300.0F);

Pen penita = new Pen(Color.Red, 1);

Rectangle dreptunghi = new Rectangle(-x, -x, x+75, x);

Rectangle dreptunghi1 = new Rectangle(-x-100, -x-100, x+75, x);

g.FillRectangle(tb, dreptunghi1);

g.FillRectangle(s, dreptunghi);

}

4. Secvență Vector

O simulare grafică utilă pentru lecția secvențe de la capitolul vectori din clasa a IX-a.

Am adus pe formă un obiect richtextBox și trei butoane.

În richtextBox vom introduce datele vectorului (pe prima linie un nr matural n, iar pe a doua linie n numere intregi).

La apăsarea pe un buton se generează vectorul, un vector de dreptunghiuri.

La apăsarea pe celălalt buton se simulează crearea de secvențe.

if (i == n)

timer1.Stop();

else

{

if (v1[i] > 0)

{

l++;

g.FillRectangle(br, v[i]);

String drawString = v1[i].ToString();

Font drawFont = new Font("Arial", 12);

SolidBrush drawBrush = new SolidBrush(Color.Blue);

float x = 118 + i \* 56;

float y = 238.0F;

g.DrawString(drawString, drawFont, drawBrush, x, y);

}

else

{

if (l > lmax)

{

lmax = l;

pmax = p;

}

l = 0; p = i;

}

i++;

} - Acest cod coloreaza numerele pozitive.

{

int i;

string s = richTextBox1.Lines[0];

n = int.Parse(s);

s = richTextBox1.Lines[1];

string[] numere = s.Split(' ');

for (i = 0; i <= n - 1; i++)

{

v1[i] = int.Parse(numere[i]);

}

g = this.CreateGraphics();

Pen p = new Pen(Color.Red, 1);

for (i = 0; i <= n - 1; i++)

{

v[i] = new Rectangle();

v[i].Height = 50;

v[i].Width = 50;

v[i].Location = new Point(100 + i \* 56, 230);

String drawString = v1[i].ToString();

Font drawFont = new Font("Arial", 12);

SolidBrush drawBrush = new SolidBrush(Color.Red);

float x = 118 + i \* 56;

float y = 238.0F;

g.DrawString(drawString, drawFont, drawBrush, x, y);

g.DrawRectangle(p, v[i]);}

} - Acest cod deseneaza secventa.

5. Parcurgere Matrice

O simulare grafica utilă pentru lecția parcurgeri în matrice de la capitolul matrice din clasa a IX-a.

Am adus pe formă două butoane, unul care desenează matricea de dreptunghiuri, altul care umple căsuțele de sub diagonala principală în timp ce le parcurge.

g = this.CreateGraphics();

Pen p = new Pen(Color.Red, 1);

for (int i = 1; i <= 6; i++)

for (int j = 1; j <= 6; j++)

{

a[i, j] = new Rectangle();

a[i, j].Height = 50;

a[i, j].Width = 50;

a[i, j].Location = new Point(100 + i \* 56, 100 + j \* 56);

g.DrawRectangle(p, a[i, j]);

}

for (int i = 1; i <= 6; i++)

for (int j = 1; j <= 6; j++)

{

if (i > j)

{

nr++;

v[nr] = new Class2();

v[nr].x = i;

v[nr].y = j;

}

} - Acest cod deseneaza matricea.

if (nr1 == nr)

{

timer1.Stop();

}

g.FillRectangle(s, a[v[nr1].y, v[nr1].x]);

nr1++; } - Acest cod umple casutele de sub diagonala principala.

6. Recursivitate.

Simulăm stiva internă pe care o creează mecanismul de recursivitate.

Se desenează la un interval de timp un nivel al stivei cu informațiile care se salvează pe stivă.

Temporizarea este facută cu metoda Thread.Sleap al bibliotecii Threading.

La revenirea din stivă se redesenează fiecare nivel cu culoarea formei. Apare efectul de ștergere.

void desen1(int i)

{

g = this.CreateGraphics();

Pen p = new Pen(Color.Blue, 1);

Rectangle r = new Rectangle(i \* 150, 100, 100, 100);

g.DrawRectangle(p, r);

if (i > 1 && i <= 5)

{

AdjustableArrowCap bigArrow = new AdjustableArrowCap(5, 5);

p.CustomEndCap = bigArrow;

g.DrawLine(p, i \* 150 - 50, 120, i \* 150, 120);

g.DrawLine(p, i \* 150, 180, i \* 150 - 50, 180);

}

String drawString = "i= " + i.ToString();

Font drawFont = new Font("Arial", 16);

SolidBrush drawBrush = new SolidBrush(Color.Red);

float x = i \* 150 + 30;

float y = 120.0F;

g.DrawString(drawString, drawFont, drawBrush, x, y);

Thread.Sleep(1000);

}

void desen2(int i)

{

g = this.CreateGraphics();

Pen p = new Pen(Color.White, 1);

Rectangle r = new Rectangle(i \* 150, 100, 100, 100);

g.DrawRectangle(p, r);

if (i >= 1)

{

AdjustableArrowCap bigArrow = new AdjustableArrowCap(5, 5);

// Pen p = new Pen(Color.Blue, 1);

p.CustomEndCap = bigArrow;

g.DrawLine(p, i \* 150 - 50, 120, i \* 150, 120);

g.DrawLine(p, i \* 150, 180, i \* 150 - 50, 180);

}

String drawString = "i= " + i.ToString();

Font drawFont = new Font("Arial", 16);

SolidBrush drawBrush = new SolidBrush(Color.White);

float x = i \* 150 + 30;

float y = 120.0F;

g.DrawString(drawString, drawFont, drawBrush, x, y);

Thread.Sleep(1000);

} - Acest cod a desenat inca odata recursivitatea, dar cu alb.

void afis(int i)

{

if (i <= 5)

{

desen1(i);

afis(i + 1);

}

desen2(i);

}

7. Liste alocate dinamic simplu înlănțuite și dublu înlănțuite. Se desenează un nod cu informațiile din noduri, se desenează sagețile și la un interval de timp se redesenează la alte coordonate. Temporizarea cu thread.Sleep.

Am adus pe forma cinci butoane, primul deseneaza o lista simplu înlănțuită și al doilea o parcurge folosind o altă culoare.

Al treilea buton desenează o listă dublu înlănțuită și al patrulea buton o parcurge din dreapta, iar al cincilea buton o parcurge din stanga.

Se impune modulizarea programului.

Graphics g;

int n;

int[] v = new int[100];

public liste()

{

InitializeComponent();

}

public class Node

{

public int val;

public Node leg;

public Node st, dr;

}

public Node c = new Node();

public Node p = new Node();

public Node u = new Node();

public void add()

{

c = new Node();

c.val = 1;

c.leg = null;

p = c; u = c;

desen(1);

for (int i = 2; i <= 7; i++)

{

c = new Node();

c.val = i;

desen(i);

c.leg = null;

u.leg = c; u = c;

}

}

public void add2()

{

c = new Node();

c.val = 1;

c.dr = null;

c.st = null;

p = c; u = c;

desen2(1);

for (int i = 2; i <= 7; i++)

{

c = new Node();

c.val = i;

desen2(i);

u.dr = c;

c.st = u;

u = c;

}

}

public void parcurgere\_dr()

{

c = p; int i = 1;

while(c!=null)

{

desen3(i);

i++;

c = c.dr;

}

}

public void parcurgere\_st()

{

c = u; int i = 7;

while (c != null)

{

desen4(i);

i--;

c = c.st;

}

}

public void parcurgere()

{

c = p; int i = 1;

while (c != null)

{

desen1(i);

i++;

c = c.leg;

}

}

void desen(int x)

{

g = this.CreateGraphics();

Pen p = new Pen(Color.Red, 1);

Pen p1 = new Pen(Color.Red, 1);

Rectangle r = new Rectangle(100 + x \* 130, 100, 50, 50);

Rectangle r1 = new Rectangle(100 + x \* 130 + 50, 100, 50, 50);

g.DrawRectangle(p, r1);

g.DrawRectangle(p, r);

p1.StartCap = System.Drawing.Drawing2D.LineCap.RoundAnchor;

p1.EndCap = System.Drawing.Drawing2D.LineCap.ArrowAnchor;

string drawString = "c";

string s1 = "NULL";

Font drawFont = new Font("Arial", 16);

Font drawFont2 = new Font("Arial", 10);

SolidBrush drawBrush = new SolidBrush(Color.Red);

if (x < 7)

{

g.DrawLine(p1, 100 + x \* 130 + 70, 125, 100 + (x + 1) \* 130, 125);

g.DrawString(drawString, drawFont, drawBrush, 60 + x \* 130 + 50, 110);

}

if (x == 7)

{

g.DrawString(s1, drawFont2, drawBrush, 98 + x \* 130 + 50, 115);

}

Thread.Sleep(500);

}

void desen1(int x)

{

g = this.CreateGraphics();

Pen p = new Pen(Color.Blue, 1);

Pen p1 = new Pen(Color.Blue, 1);

Rectangle r = new Rectangle(100 + x \* 130, 100, 50, 50);

Rectangle r1 = new Rectangle(100 + x \* 130 + 50, 100, 50, 50);

g.DrawRectangle(p, r1);

g.DrawRectangle(p, r);

p1.StartCap = System.Drawing.Drawing2D.LineCap.RoundAnchor;

p1.EndCap = System.Drawing.Drawing2D.LineCap.ArrowAnchor;

string drawString = "c";

string s1 = "NULL";

Font drawFont = new Font("Arial", 16);

Font drawFont2 = new Font("Arial", 10);

SolidBrush drawBrush = new SolidBrush(Color.Blue);

if (x < 7)

{

g.DrawLine(p1, 100 + x \* 130 + 70, 125, 100 + (x + 1) \* 130, 125);

g.DrawString(drawString, drawFont, drawBrush, 60 + x \* 130 + 50, 110);

}

if (x == 7)

{

g.DrawString(s1, drawFont2, drawBrush, 98 + x \* 130 + 50, 115);

}

Thread.Sleep(500);

}

void desen2(int x)

{

g = this.CreateGraphics();

Pen p = new Pen(Color.Red, 1);

Pen p1 = new Pen(Color.Red, 2);

Pen p2 = new Pen(Color.Red, 2);

Rectangle r = new Rectangle(100 + x \* 130, 300, 120, 50);

g.DrawRectangle(p, r);

g.DrawLine(p, 140 + x \* 130, 300, 140 + x \* 130, 350);

g.DrawLine(p, 180 + x \* 130, 300, 180 + x \* 130, 350);

AdjustableArrowCap bigArrow = new AdjustableArrowCap(3, 3);

p1.CustomEndCap = bigArrow;

p2.CustomStartCap = bigArrow;

string drawString = "c";

string s1 = "NULL";

Font drawFont = new Font("Arial", 16);

Font drawFont2 = new Font("Arial", 10);

SolidBrush drawBrush = new SolidBrush(Color.Red);

if (x < 7)

{

g.DrawLine(p1, 130 + x \* 130 + 70, 315, 130 + (x + 1) \* 130, 315);

g.DrawLine(p2, 130 + x \* 130 + 70, 335, 130 + (x + 1) \* 130, 335);

g.DrawString(drawString, drawFont, drawBrush, 103 + x \* 130 + 50, 310);

}

if(x == 1)

{

g.DrawString(s1, drawFont2, drawBrush, 47 + x \* 130 + 50, 315);

}

if (x == 7)

{

g.DrawString(s1, drawFont2, drawBrush, 128 + x \* 130 + 50, 315);

g.DrawString(drawString, drawFont, drawBrush, 105 + x \* 130 + 50, 310);

}

Thread.Sleep(500);

}

void desen3(int x)

{

g = this.CreateGraphics();

Pen p = new Pen(Color.Blue, 1);

Pen p1 = new Pen(Color.Blue, 2);

Pen p2 = new Pen(Color.Blue, 2);

Rectangle r = new Rectangle(100 + x \* 130, 300, 120, 50);

g.DrawRectangle(p, r);

g.DrawLine(p, 140 + x \* 130, 300, 140 + x \* 130, 350);

g.DrawLine(p, 180 + x \* 130, 300, 180 + x \* 130, 350);

AdjustableArrowCap bigArrow = new AdjustableArrowCap(3, 3);

p1.CustomEndCap = bigArrow;

p2.CustomStartCap = bigArrow;

string drawString = "c";

string s1 = "NULL";

Font drawFont = new Font("Arial", 16);

Font drawFont2 = new Font("Arial", 10);

SolidBrush drawBrush = new SolidBrush(Color.Blue);

if (x < 7)

{

g.DrawLine(p1, 130 + x \* 130 + 70, 315, 130 + (x + 1) \* 130, 315);

g.DrawLine(p2, 130 + x \* 130 + 70, 335, 130 + (x + 1) \* 130, 335);

g.DrawString(drawString, drawFont, drawBrush, 103 + x \* 130 + 50, 310);

}

if (x == 1)

{

g.DrawString(s1, drawFont2, drawBrush, 47 + x \* 130 + 50, 315);

}

if (x == 7)

{

g.DrawString(s1, drawFont2, drawBrush, 128 + x \* 130 + 50, 315);

g.DrawString(drawString, drawFont, drawBrush, 105 + x \* 130 + 50, 310);

}

Thread.Sleep(500);

}

void desen4(int x)

{

g = this.CreateGraphics();

Pen p = new Pen(Color.Green, 1);

Pen p1 = new Pen(Color.Green, 2);

Pen p2 = new Pen(Color.Green, 2);

Rectangle r = new Rectangle(100 + x \* 130, 300, 120, 50);

g.DrawRectangle(p, r);

g.DrawLine(p, 140 + x \* 130, 300, 140 + x \* 130, 350);

g.DrawLine(p, 180 + x \* 130, 300, 180 + x \* 130, 350);

AdjustableArrowCap bigArrow = new AdjustableArrowCap(3, 3);

p1.CustomEndCap = bigArrow;

p2.CustomStartCap = bigArrow;

string drawString = "c";

string s1 = "NULL";

Font drawFont = new Font("Arial", 16);

Font drawFont2 = new Font("Arial", 10);

SolidBrush drawBrush = new SolidBrush(Color.Green);

if (x < 7)

{

g.DrawLine(p1, 130 + x \* 130 + 70, 315, 130 + (x + 1) \* 130, 315);

g.DrawLine(p2, 130 + x \* 130 + 70, 335, 130 + (x + 1) \* 130, 335);

g.DrawString(drawString, drawFont, drawBrush, 103 + x \* 130 + 50, 310);

}

if (x == 1)

{

g.DrawString(s1, drawFont2, drawBrush, 47 + x \* 130 + 50, 315);

}

if (x == 7)

{

g.DrawString(s1, drawFont2, drawBrush, 128 + x \* 130 + 50, 315);

g.DrawString(drawString, drawFont, drawBrush, 105 + x \* 130 + 50, 310);

}

Thread.Sleep(500);

}

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

{

add();

}

private void button2\_Click(object sender, EventArgs e)

{

parcurgere();

}

private void button3\_Click(object sender, EventArgs e)

{

g = this.CreateGraphics();

g.Clear(Color.White);

add2();

}

private void button4\_Click(object sender, EventArgs e)

{

parcurgere\_dr();

}

private void button5\_Click(object sender, EventArgs e)

{

parcurgere\_st();

}

8. Animație Matrice

Am adus pe formă doua butoane, un timer și un openfiledialog.

Cu ajutorul openfiledialog-ului, putem selecta o imagine din File Explorer-ul personal.

În textbox vom scrie un număr natural, care reprezintă numărul de linii și de coloane în care vom împărți imaginea aleasă. Primul buton înregistrează acest numar, iar al doilea pornește timer-ul care desenează imaginea. Se vor afișa elementele matricii linie cu linie.

public partial class anim\_matrice : Form

{

Graphics g;

Bitmap[,] bmps = new Bitmap[1000, 1000];

Bitmap[,] bmps1 = new Bitmap[1000, 1000];

Random r = new Random();

Rectangle[,] dr = new Rectangle[600, 600];

Button[,] b = new Button[10000, 10000];

int[,] a11 = new int[300, 300];

int[,] c = new int[100, 100];

public int x1, y1, x2, y2, a1 = 1, nr = 0, ok, x, k, nr1 = 1, n;

private void timer1\_Tick(object sender, EventArgs e)

{

if (nr1 == nr + 1) timer1.Stop();

else

{

Bitmap bimage = new Bitmap(bmps1[v[nr1].x, v[nr1].y]);

TextureBrush tb = new TextureBrush(bimage);

g.FillRectangle(tb, dr[v[nr1].x, v[nr1].y]);

nr1++;

}

}

private void button2\_Click(object sender, EventArgs e)

{

timer1.Start();

}

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

{

int i;

n = int.Parse(textBox1.Text);

if (openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

s = openFileDialog1.InitialDirectory + openFileDialog1.FileName;

}

int j;

Image img = Image.FromFile(s.ToString());

MessageBox.Show(img.Width.ToString() + " " + img.Height.ToString());

int widththird = (int)((double)img.Width / n);

int heightthird = (int)((double)img.Height / n);

MessageBox.Show(widththird.ToString() + " " + heightthird.ToString());

for (i = 0; i < n; i++)

for (j = 0; j < n; j++)

{

dr[i, j] = new Rectangle();

dr[i, j].Height = heightthird;

dr[i, j].Width = widththird;

bmps1[i, j] = new Bitmap(widththird, heightthird);

g1 = Graphics.FromImage(bmps1[i, j]);

g1.DrawImage(img, new Rectangle(0, 0, widththird, heightthird),

new Rectangle(i \* widththird, j \* heightthird, widththird, heightthird),

GraphicsUnit.Pixel);

g = this.CreateGraphics();

dr[i, j].Location = new Point(i \* (widththird), j \* (heightthird));

Pen p = new Pen(Color.Red, 1);

g.DrawRectangle(p, dr[i, j]);

}

for (i = 0; i < n; i++)

for (j = 0; j < n; j++)

{

nr++;

v[nr] = new Class2();

v[nr].x = i;

v[nr].y = j;

}

}

int[] vf = new int[100000];

int[] v1 = new int[100000];

int[] v2 = new int[1000];

int[,] d = new int[10000, 10000];

Class2[] v = new Class2[100000];

Graphics g1;

string s;

9. Algoritm Fill

Se colorează zonele de uscat care într-o matrice de tip întreg au valoarea 1, 0 pentru zonele de apă.

Programul creează randomizat o matrice de elemente întregi 0/apă și 1 pentru uscat.

Am adus două butoane pe formă și un timer. Primul buton desenează o matrice de patrate. Cu al doilea buton se pornește obiectul timer care colorează succesiv pătrățele care reprezintă uscatul.

public partial class algoritm\_fill : Form

{

int n = 8;

Random r = new Random();

int[,] a1 = new int[100, 100];

Rectangle[,] a = new Rectangle[100, 100];

Graphics g;

int i, j;

public algoritm\_fill()

{

InitializeComponent();

}

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

{

g = this.CreateGraphics();

Pen p = new Pen(Color.Red, 1);

int i, j;

for (i = 1; i <= n; i++)

for (j = 1; j <= n; j++)

a1[i, j] = r.Next(2);

for (i = 1; i <= n; i++)

for (j = 1; j <= n; j++)

{

a[i, j] = new Rectangle();

a[i, j].Height = 50;

a[i, j].Width = 50;

a[i, j].Location = new Point(100 + i \* 56, 100 + j \* 56);

g.DrawRectangle(p, a[i, j]);

}

}

private void timer1\_Tick(object sender, EventArgs e)

{

if (j <= n)

{

if (a1[i, j] == 1) fill(i, j);

else if (a1[i, j] != 2)

{

SolidBrush s = new SolidBrush(Color.Red);

g.FillRectangle(s, a[i, j]);

}

j++;

}

else { i++; j = 1; }

if (i == n && j == n + 1) { timer1.Stop(); button1.Enabled = true; }

}

void fill(int i, int j)

{

SolidBrush s = new SolidBrush(Color.Green);

g.FillRectangle(s, a[i, j]);

a1[i, j] = 2;

if (a1[i - 1, j] == 1) fill(i - 1, j);

if (a1[i, j + 1] == 1) fill(i, j + 1);

if (a1[i + 1, j] == 1) fill(i + 1, j);

if (a1[i, j - 1] == 1) fill(i, j - 1);

}

private void button2\_Click(object sender, EventArgs e)

{

i = 1;

j = 1;

timer1.Start();

}

V. Teste pentru verificare de cunoștințe

Pentru următoarele aplicații, pentru salvarea și prelucrarea datelor am folosit și fisiere test, dar și o bază de date în sqlServer care cuprinde tabele elev(id, nume\_prenume, clasa, e\_mail, telefon) și tabela note(id, nota, tip\_test,data). Utilizatorul are obligația de a crea niște fișiere text care cuprinde itemi (întrebări, posibile soluții și răspunsul corect).

Testele încarcă datele din fișierele utilizatorilor, creează automat testul, realizează evaluarea automat și trasmite utilizatorului rezultatele prelucrate. La încărcarea formei apar în ListBox numele și prenumele elevilor. Elevul își selectează numele din listă. Apare id și clasa automat în două obiecte TextBox. Se generează testul și nota obținută se trimite și către baza de date, și către fișier pentru prelucrări.

Valabil și pentru testul cu checkBox-uri.

1. Test Radio

Am adus 5 butoane, două textbox-uri, un listbox, patru radioButtons și un groupBox.

Mai întâi alegi user-ul din listbox, apoi apeși butonul care generează vectorul, apoi cu ajutorul butoanelor < și > alegi întrebarea. Dupa ce apeși răspunsul corect, îți verifici răspunsurile, rezultatul fiind salvat în database, folosind butonul de înregistrare rezultate.

public partial class radio : Form

{

Class1[] v = new Class1[100];

int[] s = new int[100];

int k, nr = 0, i;

int[] v1 = new int[100];

SqlConnection con = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Wind 10\source\repos\ProiectGraphics\Catalog.mdf;Integrated Security=True");

public radio()

{

InitializeComponent();

}

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

{

if (nr < k)

{

nr++; //MessageBox.Show(nr.ToString());

groupBox1.Text = v[nr].intrebare;

radioButton1.Text = v[nr].r1;

radioButton2.Text = v[nr].r2;

radioButton3.Text = v[nr].r3;

}

}

private void button2\_Click(object sender, EventArgs e)

{

if (nr > 1)

{

nr--;// MessageBox.Show(nr.ToString());

groupBox1.Text = v[nr].intrebare;

radioButton1.Text = v[nr].r1;

radioButton2.Text = v[nr].r2;

radioButton3.Text = v[nr].r3;

}

}

private void button4\_Click(object sender, EventArgs e)

{

int i, rezultat = 0;

for (i = 1; i <= k; i++)

rezultat = rezultat + s[i];

MessageBox.Show(rezultat.ToString());

}

private void radioButton1\_CheckedChanged(object sender, EventArgs e)

{

if (string.Compare(v[nr].r1.Trim(), v[nr].rc.Trim()) == 0)

s[nr] = 2;

else

s[nr] = 0;

}

private void radioButton2\_CheckedChanged(object sender, EventArgs e)

{

if (string.Compare(v[nr].r2.Trim().ToString(), v[nr].rc.Trim().ToString()) == 0)

s[nr] = 2;

else

s[nr] = 0;

}

private void radioButton3\_CheckedChanged(object sender, EventArgs e)

{

if (string.Compare(v[nr].r3.Trim(), v[nr].rc.Trim()) == 0)

s[nr] = 2;

else

s[nr] = 0;

}

private void button5\_Click(object sender, EventArgs e)

{

con.Open();

string insert = @"insert into catalog(tip\_test,nota,data,id\_clasa)values(@tip\_test,@nota,@data,@id\_clasa)";

SqlCommand cmd = new SqlCommand(insert, con);

cmd.Parameters.AddWithValue("tip\_test", "radio");

cmd.Parameters.AddWithValue("id\_clasa", textBox1.Text);

cmd.Parameters.AddWithValue("nota", nr.ToString());

cmd.Parameters.AddWithValue("data", DateTime.Now.ToString());

SqlDataReader r = cmd.ExecuteReader();

con.Close();

if (openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

s1 = openFileDialog1.InitialDirectory + openFileDialog1.FileName;

}

using (StreamWriter f = File.AppendText(s1))

{

f.WriteLine(textBox1.Text.Trim() + "|" + listBox1.Text.Trim() + "|" + textBox2.Text.Trim() + "|check|" + nr.ToString() + "|" + DateTime.Now.ToString());

f.Close();

}

}

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

con.Open();

string select = "select id, nume\_prenume,clasa from elevi";

SqlCommand cmd = new SqlCommand(select, con);

SqlDataReader r = cmd.ExecuteReader();

while (r.Read())

{

if (string.Compare(listBox1.Text.Trim(), r[1].ToString().Trim()) == 0)

{

textBox1.Text = r[0].ToString();

textBox2.Text = r[2].ToString();

}

}

con.Close();

}

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

{

for (i = 1; i <= 99; i++)

v1[i] = 0;

con.Open();

string select = "select id, nume\_prenume, clasa from elevi";

SqlCommand cmd = new SqlCommand(select, con);

SqlDataReader r = cmd.ExecuteReader();

while (r.Read())

{

listBox1.Items.Add(r[1].ToString());

}

con.Close();

}

private void button3\_Click(object sender, EventArgs e)

{

k = 0;

using (StreamReader fin = new StreamReader("TextFile3.txt"))

{

while (!fin.EndOfStream)

{

string linie = fin.ReadLine();

string[] v1 = linie.Split('|');

k++;

v[k] = new Class1();

v[k].intrebare = v1[0].ToString();

v[k].r1 = v1[1].ToString();

v[k].r2 = v1[2].ToString();

v[k].r3 = v1[3].ToString();

v[k].rc = v1[4].ToString();

}

fin.Close();

}

}

2. Test cu Check-uri

Am adus două butoane, două textbox-uri, un listbox și patru checkbox-uri.

Elevul selectează numele din listbox.

Primul buton generează testul. Convenție: se bifează checkBox dacă răspunsul este "da".

Al treilea buton salveaza rezultatele.

{

SqlConnection con = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Wind 10\source\repos\ProiectGraphics\Catalog.mdf;Integrated Security=True");

int i, j, ok, x, n, nr = 0, n1 = 0;

int[] v1 = new int[100];

CheckBox[] c = new CheckBox[10];

int[] v = new int[100];

string s;

intreb[] vect = new intreb[100];

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

{

con.Open();

string select = "select id, nume\_prenume,clasa from elevi";

SqlCommand cmd = new SqlCommand(select, con);

SqlDataReader r = cmd.ExecuteReader();

while (r.Read())

{

listBox1.Items.Add(r[1].ToString());

}

con.Close();

}

struct intreb

{

public string test, raspuns;

};

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

con.Open();

string select = "select id, nume\_prenume, clasa from elevi";

SqlCommand cmd = new SqlCommand(select, con);

SqlDataReader r = cmd.ExecuteReader();

while (r.Read())

{

if (string.Compare(listBox1.Text.Trim(), r[1].ToString().Trim()) == 0)

{

textBox1.Text = r[0].ToString();

textBox2.Text = r[2].ToString();

}

}

con.Close();

}

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

{

if (openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

s = openFileDialog1.InitialDirectory + openFileDialog1.FileName;

}

StreamReader f = new StreamReader(s);

while (!f.EndOfStream)

{

n1++;

vect[n1].test = f.ReadLine();

vect[n1].raspuns = f.ReadLine();

//MessageBox.Show(vect[n1].test);

}

Random r = new Random();

for (i = 1; i <= 4; i++)

{

int ok = 0;

while (ok == 0)

{

x = r.Next(1, n1);

if (v1[x] == 0)

{

v1[x] = 1;

v[i] = x;

ok = 1;

}

}

}

checkBox1.Text = vect[v[1]].test;

checkBox2.Text = vect[v[2]].test;

checkBox3.Text = vect[v[3]].test;

checkBox4.Text = vect[v[4]].test;

}

private void button2\_Click(object sender, EventArgs e)

{

if (checkBox1.Checked && vect[v[1]].raspuns == "Da")

nr++;

if (checkBox2.Checked && vect[v[2]].raspuns == "Da")

nr++;

if (checkBox3.Checked && vect[v[3]].raspuns == "Da")

nr++;

if (checkBox4.Checked && vect[v[4]].raspuns == "Da")

nr++;

int nota = nr \* 2 + 2;

MessageBox.Show(nota.ToString());

}

private void button3\_Click(object sender, EventArgs e)

{

con.Open();

string insert = @"insert into catalog(tip\_test,nota,data,id\_clasa)values(@tip\_test,@nota,@data,@id\_clasa)";

SqlCommand cmd = new SqlCommand(insert, con);

cmd.Parameters.AddWithValue("id\_clasa", textBox1.Text);

cmd.Parameters.AddWithValue("tip\_test", "check");

cmd.Parameters.AddWithValue("nota", nota.ToString());

cmd.Parameters.AddWithValue("data", DateTime.Now.ToString());

SqlDataReader r = cmd.ExecuteReader();

textBox1.Clear();

con.Close();

if (openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

s = openFileDialog1.InitialDirectory + openFileDialog1.FileName;

}

using (StreamWriter f = File.AppendText(s))

{

f.WriteLine(textBox1.Text.Trim() + "|" + listBox1.Text.Trim() + "|" + textBox2.Text.Trim() + "|check|" + nota.ToString() + "|" + DateTime.Now.ToString());

f.Close();

}

}

3. Itemi de Completare

Am adus trei butoane, trei textbox-uri, un listbox și patru checkbox-uri.

Poți să alegi user-ul din listbox.

Scrii numărul de întrebări în textBox, după care vor apărea numărul de întrebări și textbox-uri lângă ele unde poți scrie răspunsul. După ce ai scris răspunsurile apasă butonul de verificare răspunsuri, care îți va arăta nota. Trebuie să apeși butonul care salvează rezultatul ca să salvezi nota în database.

public partial class itemi : Form

{

int i = 0, nr, n, x;

Class6[] v = new Class6[100];

Label[] l = new Label[100];

TextBox[] t = new TextBox[100];

int[] vf = new int[100];

SqlConnection con = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Wind 10\source\repos\ProiectGraphics\Catalog.mdf;Integrated Security=True");

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

{

int i1;

i = 0;

using (StreamReader f = new StreamReader("TextFile2.txt"))

{

while (!f.EndOfStream)

{

v[i] = new Class6();

v[i].intrebare = f.ReadLine();

v[i].raspuns = f.ReadLine();

i++;

}

f.Close();

}

n = int.Parse(textBox1.Text.ToString());

for (i1 = 0; i1 < n; i1++)

{

int ok = 1;

while (ok == 1)

{

x = r.Next(0, i);

if (vf[x] == 0)

{

ok = 0;

v1[i1] = x;

vf[x]++;

}

}

}

for (int j = 0; j < n; j++)

{

l[j] = new Label();

l[j].Text = v[v1[j]].intrebare;

this.l[j].AutoSize = true;

l[j].Location = new Point(400, 200 + j \* 40);

this.Controls.Add(l[j]);

t[j] = new TextBox();

t[j].Location = new Point(300, 200 + j \* 40);

this.Controls.Add(t[j]);

}

}

private void button2\_Click(object sender, EventArgs e)

{

nr = 0;

for (int j = 0; j < n; j++)

{

if (string.Compare(t[j].Text.Trim(), v[v1[j]].raspuns.Trim()) == 0)

nr++;

}

MessageBox.Show(nr.ToString());

}

private void label2\_Click(object sender, EventArgs e)

{

this.Close();

introducere f = new introducere();

f.Show();

}

private void button3\_Click(object sender, EventArgs e)

{

con.Open();

string insert = @"insert into catalog(tip\_test,nota,data,id\_clasa)values(@tip\_test,@nota,@data,@id\_clasa)";

SqlCommand cmd = new SqlCommand(insert, con);

cmd.Parameters.AddWithValue("id\_clasa", textBox1.Text);

cmd.Parameters.AddWithValue("tip\_test", "itemi");

cmd.Parameters.AddWithValue("nota", nr.ToString());

cmd.Parameters.AddWithValue("data", DateTime.Now.ToString());

SqlDataReader r = cmd.ExecuteReader();

con.Close();

if (openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

s1 = openFileDialog1.InitialDirectory + openFileDialog1.FileName;

}

using (StreamWriter f = File.AppendText(s1))

{

f.WriteLine(textBox1.Text.Trim() + "|" + listBox1.Text.Trim() + "|" + textBox2.Text.Trim() + "|check|" + nr.ToString() + "|" + DateTime.Now.ToString());

f.Close();

}

}

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

{

for (i = 1; i <= 99; i++)

v1[i] = 0;

con.Open();

string select = "select id, nume\_prenume, clasa from elevi";

SqlCommand cmd = new SqlCommand(select, con);

SqlDataReader r = cmd.ExecuteReader();

while (r.Read())

{

listBox1.Items.Add(r[1].ToString());

}

con.Close();

}

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

con.Open();

string select = "select id, nume\_prenume ,clasa from elevi";

SqlCommand cmd = new SqlCommand(select, con);

SqlDataReader r = cmd.ExecuteReader();

while (r.Read())

{

if (string.Compare(listBox1.Text.Trim(), r[1].ToString().Trim()) == 0)

{

textBox2.Text = r[0].ToString();

textBox3.Text = r[2].ToString();

}

}

con.Close();

}

Random r = new Random();

int[] v1 = new int[100];

public itemi()

{

InitializeComponent();

}

4. Test Grilă cu Flori

Noutatea este că am înlocuit acele obiecte (butoane radio) cu niște desene.

Am adus pe formă două butoane. Primul buton afișează testul, iar al doilea verifică răspunsul.

Evaluarea se realizează automat și se afișează rezultatul.

SolidBrush s;

Graphics g;

bool ok;

string rezultat;

public test\_grila()

{

InitializeComponent();

}

void floare(int a, int b, Color c, string nume)

{

g = this.CreateGraphics();

String drawString = nume;

Font drawFont = new Font("Arial", 14);

SolidBrush drawBrush = new SolidBrush(Color.Black);

PointF drawPoint = new PointF(a + 40, b - 15);

Pen blackPen = new Pen(Color.Orange, 3);

g.DrawString(drawString, drawFont, drawBrush, drawPoint);

SolidBrush s = new SolidBrush(c);

Point start = new Point(5, 0);

Point control1 = new Point(25, -4);

Point control2 = new Point(35, 0);

Point control3 = new Point(25, 4);

Point end = new Point(5, 0);

PointF[] curvePoints =

{

start,

control1,

control2,

control3,

end

};

g.TranslateTransform(a, b);

for (int i = 1; i <= 12; i++)

{

g.RotateTransform(30.0F);

g.FillClosedCurve(s, curvePoints);

}

SolidBrush s1 = new SolidBrush(Color.Red);

Rectangle r1 = new Rectangle(-7, -7, 15, 15);

g.FillEllipse(s1, r1);

}

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

{

g = this.CreateGraphics();

String drawString = "Care floare nu face parte din familia Asteraceae";

Font drawFont = new Font("Arial", 14);

SolidBrush drawBrush = new SolidBrush(Color.Black);

PointF drawPoint = new PointF(50.0F, 40.0F);

g.DrawString(drawString, drawFont, drawBrush, drawPoint);

floare(100, 100, Color.Yellow, "Trandafirul");

floare(100, 200, Color.Yellow, "Floarea Soarelui");

floare(100, 300, Color.Yellow, "Papadia");

}

private void test\_grila\_MouseClick(object sender, MouseEventArgs e)

{

if (Math.Sqrt((e.X - 100) \* (e.X - 100) + (e.Y - 100) \* (e.Y - 100)) < 35)

{

floare(100, 100, Color.Blue, "Trandafirul");

floare(100, 200, Color.Yellow, "Floarea Soarelui");

floare(100, 300, Color.Yellow, "Papadia");

ok = true;

}

else if (Math.Sqrt((e.X - 100) \* (e.X - 100) + (e.Y - 200) \* (e.Y - 200)) < 35)

{

floare(100, 100, Color.Yellow, "Trandafirul");

floare(100, 200, Color.Blue, "Floarea Soarelui");

floare(100, 300, Color.Yellow, "Papadia");

ok = false;

}

else if (Math.Sqrt((e.X - 100) \* (e.X - 100) + (e.Y - 300) \* (e.Y - 300)) < 35)

{

floare(100, 100, Color.Yellow, "Trandafirul");

floare(100, 200, Color.Yellow, "Floarea Soarelui");

floare(100, 300, Color.Blue, "Papadia");

ok = false;

}

}

private void button2\_Click(object sender, EventArgs e)

{

if (ok == true)

rezultat = "Raspuns corect!";

else

rezultat = "Raspuns gresit";

Font drawFont = new Font("Arial", 14);

SolidBrush drawBrush = new SolidBrush(Color.Black);

PointF drawPoint = new PointF(100, 400);

Pen blackPen = new Pen(Color.Blue, 3);

g = this.CreateGraphics();

g.DrawString(rezultat, drawFont, drawBrush, drawPoint);

}

5. Test de recunoaștere a imaginilor

Utilizatorul are mai multe imagini. Programul selectează imaginea din calculator cu ajutorul obiectului Open FileDialog și salvează într-un fișier numele imaginii și adresa.

Realizează automat testul, alege randomizat trei imagini. Selectează din cele trei alese una și construiește o întrebare "care este imaginea care are numele" + numele imaginii selectate.

Evaluare se face automat.

Am adus pe aceasta formă trei butoane, un openfiledialog și trei picturebox-uri.

public partial class test\_imagini : Form

{

Graphics g;

string s;

int n1 = 0, i, x, y, nota = 1;

int[] v = new int[100];

int[] v1 = new int[100];

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

{

using (StreamReader f = new StreamReader("TextFile4.txt"))

{

while (!f.EndOfStream)

{

n1++;

vect[n1] = new Class4();

string s1 = f.ReadLine();

string[] vs = s1.Split('/');

vect[n1].nume = vs[0].ToString();

vect[n1].imagine = vs[1].ToString();

}

f.Close();

}

// Random r = new Random();

for (i = 1; i <= 3; i++)

{

int ok = 0;

while (ok == 0)

{

x = r1.Next(1, n1);

if (v1[x] == 0)

{

v1[x] = 1;

v[i] = x;

ok = 1;

}

}

}

Image i1 = Image.FromFile(vect[v[1]].imagine.ToString());

pictureBox1.Image = i1;

Image i2 = Image.FromFile(vect[v[2]].imagine.ToString());

pictureBox2.Image = i2;

Image i3 = Image.FromFile(vect[v[3]].imagine.ToString());

pictureBox3.Image = i3;

y = r1.Next(1, 3);

g = this.CreateGraphics();

String drawString = "Care este imaginea care reprezinta " + vect[v[y]].nume.ToString();

Font drawFont = new Font("Arial", 14);

SolidBrush drawBrush = new SolidBrush(Color.Black);

float x1 = 250.0F;

float y1 = 250.0F;

float width = 400.0F;

float height = 50.0F;

RectangleF drawRect = new RectangleF(x1, y1, width, height);

Pen blackPen = new Pen(Color.Black);

g.DrawString(drawString, drawFont, drawBrush, drawRect);

}

private void button2\_Click(object sender, EventArgs e)

{

if (openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

s = openFileDialog1.InitialDirectory + openFileDialog1.FileName;

string result;

result = Path.GetFileName(s);

using (StreamWriter fout = File.AppendText("TextFile4.txt"))

{

fout.WriteLine(result + "/" + s);

fout.Close();

}

}

}

private void button3\_Click(object sender, EventArgs e)

{

MessageBox.Show(nota.ToString());

}

private void pictureBox1\_MouseClick(object sender, MouseEventArgs e)

{

x = 1;

if (v[x] == v[y])

nota = 10;

else

nota = 1;

}

private void pictureBox2\_MouseClick(object sender, MouseEventArgs e)

{

x = 2;

if (v[x] == v[y])

nota = 10;

else

nota = 1;

}

private void pictureBox3\_MouseClick(object sender, MouseEventArgs e)

{

x = 3;

if (v[x] == v[y])

nota = 10;

else

nota = 1;

}

Random r1 = new Random();

Class4[] vect = new Class4[100];

6. Înregistrare Elevi

Pe aceasta formă am adus un buton și 4 textbox-uri, în care se introduce numele, clasa, telefonul și email-ul. Acestea sunt apoi salvate într-un database .

public partial class inregistrare\_elevi : Form

{

SqlConnection c = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Wind 10\source\repos\ProiectGraphics\Catalog.mdf;Integrated Security=True");

public inregistrare\_elevi()

{

InitializeComponent();

}

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

{

c.Open();

string insert = @"insert into elevi(nume\_prenume,clasa, telefon,e\_mail)values(@nume\_prenume,@clasa, @telefon,@e\_mail)";

SqlCommand cmd = new SqlCommand(insert, c);

cmd.Parameters.AddWithValue("nume\_prenume", textBox1.Text);

cmd.Parameters.AddWithValue("clasa", textBox2.Text);

cmd.Parameters.AddWithValue("telefon", textBox3.Text);

cmd.Parameters.AddWithValue("e\_mail", textBox4.Text);

SqlDataReader r = cmd.ExecuteReader();

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

c.Close();

}

7. Interogări.

Afișăm într-un obiect DataGridView media elevilor și media fiecărei clase.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace ProiectGraphics

{

public partial class test1 : Form

{

SqlConnection c=new SqlConnection (@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Wind 10\source\repos\ProiectGraphics\Catalog.mdf;Integrated Security=True");

public introgare\_baze\_de\_date()

{

InitializeComponent();

}

//afisam mediile pe clase

private void button2\_Click(object sender, EventArgs e)

{

c.Open();

string select = "select b.clasa, avg(a.nota) from catalog a, elevi b group by b.clasa ";

SqlCommand cmd = new SqlCommand(select, c);

SqlDataReader r = cmd.ExecuteReader();

while (r.Read())

dataGridView2.Rows.Add(r[0], r[1]);

c.Close();

}

//afisam mediile elevilor

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

{

c.Open();

string select = "select b.nume\_prenume, avg(a.nota) from catalog a, elevi b group by b.nume\_prenume";

SqlCommand cmd = new SqlCommand(select, c);

SqlDataReader r = cmd.ExecuteReader();

while (r.Read())

dataGridView1.Rows.Add(r[0], r[1]);

c.Close();

}

}

}

8. Tabel Date

Afișăm într-un datagridview datele din baza de date elevi.

public partial class tabel : Form

{

SqlConnection c = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Wind 10\source\repos\ProiectGraphics\Catalog.mdf;Integrated Security=True");

public tabel()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

this.Close();

meniu f = new meniu();

f.Show();

}

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

{

c.Open();

string select = "select \* from elevi";

SqlCommand cmd = new SqlCommand(select, c);

SqlDataReader r = cmd.ExecuteReader();

while (r.Read())

dataGridView1.Rows.Add(r[0], r[1],r[2],r[3],r[4]);

c.Close();

}

}

În formă am folosit button1\_MouseHover și button1\_MouseLeave ca să schimb culoarea butonului și label1\_Click ca să închid forma și să o deschid pe cea dinainte.

Pagina de start

Pe forma am adus 4 textbox-uri, doua butoane și doua checkbox-uri.

În textbox-uri sunt introduse username și password, care sunt apoi scrise într-un database, folosind butonul de înregistrare.

Se poate folosi checkbox-ul ca să se afișeze parola.

Apoi, username-ul și parola sunt introduse în celelalte textbox-uri, iar user-ul se loghează.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace ProiectGraphics

{

public partial class pagina\_start : Form

{

SqlConnection c = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Wind 10\source\repos\ProiectGraphics\Catalog.mdf;Integrated Security=True");

public pagina\_start()

{

InitializeComponent();

}

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

{

c.Open();

string insert =@"insert into logare(username,password)values(@username,@password)";

SqlCommand cmd = new SqlCommand(insert, c);

cmd.Parameters.AddWithValue("username", textBox2.Text);

cmd.Parameters.AddWithValue("password", textBox1.Text);

SqlDataReader r = cmd.ExecuteReader();

textBox2.Clear();

textBox1.Clear();

c.Close();

}

private void button2\_Click(object sender, EventArgs e)

{

c.Open();

string select = "select \* from logare where username='" + textBox3.Text + "' and password= '" + textBox4.Text + "'";

SqlCommand cmd = new SqlCommand(select, c);

SqlDataReader r = cmd.ExecuteReader();

if(r.Read() == true)

{

textBox3.Clear();

textBox4.Clear();

meniu f = new meniu();

f.Show();

}

c.Close();

}

private void checkBox2\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox2.Checked)

{

textBox1.PasswordChar = (char)0;

}

else

{

textBox1.PasswordChar = '\*';

}

}

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox1.Checked)

{

textBox4.PasswordChar = (char)0;

}

else

{

textBox4.PasswordChar = '\*';

}

}

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

{

button1.BackColor = Color.Gold;

}

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

{

button1.BackColor = Color.FromArgb(236, 83, 59);

}

private void button2\_MouseHover(object sender, EventArgs e)

{

button2.BackColor = Color.Gold;

}

private void button2\_MouseLeave(object sender, EventArgs e)

{

button2.BackColor = Color.FromArgb(236, 83, 59);

}

private void pagina\_start\_Load(object sender, EventArgs e)

{

WinAPI.AnimateWindow(this.Handle, 2000, WinAPI.CENTER);

}

}

}

Am folosit o clasă ca să adaug o animație când se deschide forma.

using System;

using System.Collections.Generic;

using System.Text;

using System.Linq;

using System.Runtime.InteropServices;

namespace ProiectGraphics

{

class WinAPI

{

public const int HOR\_Positive = 0X1;

public const int HOR\_Negative = 0X2;

public const int VER\_Positive = 0X4;

public const int VER\_Negative = 0X8;

public const int CENTER = 0X10;

public const int BLEND = 0X80000;

[DllImport("user32.dll", CharSet = CharSet.Auto)]

public static extern int AnimateWindow(IntPtr hwand,int dwTime,int dwFlag);

}

}

Securitate

Utilizatorul folosește fișierul executabil care nu poate fi modificat.

Nu am permis nici măcar importuri de date persistente.

Când se importă date și imagini se folosesc pentru acea aplicație, importurile nu rămân.

În caz de ștergere a fișierului executabil se poate descărca din multe locuri unde este salvat.

Biografie

Note curs

MSDN c#

Charles Petzold- "Programare în Windows cu C#", Teora, 2003

Herbert Schildt -"C#", Teora, 2002

Constantin Gălăţan, Susana Gălăţan- Programare în Visual C#, L&S Infomat, 2008

Ana Întuneric, Cristina Sichim, Daniela Tarasă- Aplicaţii Windows Visual în C#, Polirom, 2010

Documentaţia online MSDN pentru mediul vizual de dezvoltare Microsoft Visual Studio (varianta C#)

Ileana Popescu, Letiţia Velcescu- Proiectarea Bazelor de Date, Editura Universităţii din Bucureşti, 2007

Sursa: https://www.youtube.com/watch?v=cJfRfagu7cw

Surse imagini:

https://www.proflowers.com/blog/blue-flowers/

https://doftoria.ro/cimbrisor/

https://ro.wikipedia.org/wiki/Floarea-reginei#/media/Fi%C8%99ier:LeontopodiumAlpinum-1.jpg

https://ro.wikipedia.org/wiki/Narcis%C4%83

https://theflowerboxatelier.ro/de-ce-sa-oferi-bujori-si-ce-semnificatie-au/

https://ro.wikipedia.org/wiki/G%C4%83lbenele#/media/Fi%C8%99ier:2006-10-22Calendula01.jpg

Figuri elementare

Linie - Enache Maria Eliza

Dreptunghi - Enache Maria Eliza

Elipsa - Enache Maria Eliza

Poligon - Enache Maria Eliza

Paint - Enache Maria Eliza

Rotatii si translatii

Rotatie cu translatie - Militaru Rucsandra Ioana

Desenare soare - Militaru Rucsandra Ioana

Rotatie dreptunghi - Enache Maria Eliza

Miscarea unei imagini - Militaru Rucsandra Ioana

Desenare floare - Militaru Rucsandra Ioana

Rotiri string - Militaru Rucsandra Ioana

Lista circulara - Enache Maria Eliza

Jocuri

Deplasare cu taste - Enache Maria Eliza

Joc memorie - Enache Maria Eliza

X si 0 - Militaru Rucsandra Ioana

Harta - Militaru Rucsandra Ioana

Animatii

Animatii string - Enache Maria Eliza

Slideshow - Militaru Rucsandra Ioana

Marire dreptunghi - Enache Maria Eliza

Secventa vector - Militaru Rucsandra Ioana

Parcurgere matrice - Militaru Rucsandra Ioana

Recursivitate - Militaru Rucsandra Ioana

Liste simplu inlantuite - Enache Maria Eliza

Liste dublu inlantuite - Enache Maria Eliza

Animatie matrice - Militaru Rucsandra Ioana

Algoritm fill - Enache Maria Eliza

Teste

Test radio - Enache Maria Eliza

Test cu check-uri - Enache Maria Eliza

Test cu itemi de completare - Enache Maria Eliza

Test cu alegere multipla - Militaru Rucsandra Ioana

Înregistrare elevi - Enache Maria Eliza

Tabel medie - Militaru Rucsandra Ioana

Test recunoastere imagini - Militaru Rucsandra Ioana

Tabel date – Militaru Rucsandra Ioana

Functii matematice

Grafice matematice - Militaru Rucsandra Ioana

Meniu - Enache Maria Eliza

Logare - Militaru Rucsandra Ioana