

Alegem ce vem sa desanam

Patrat, dreptunghi, cerc sau linie

Butoane pentru salvare si deschidere a imaginii

Alegerea culorii

Fig. 1 Programa de baza

Zona de desenare

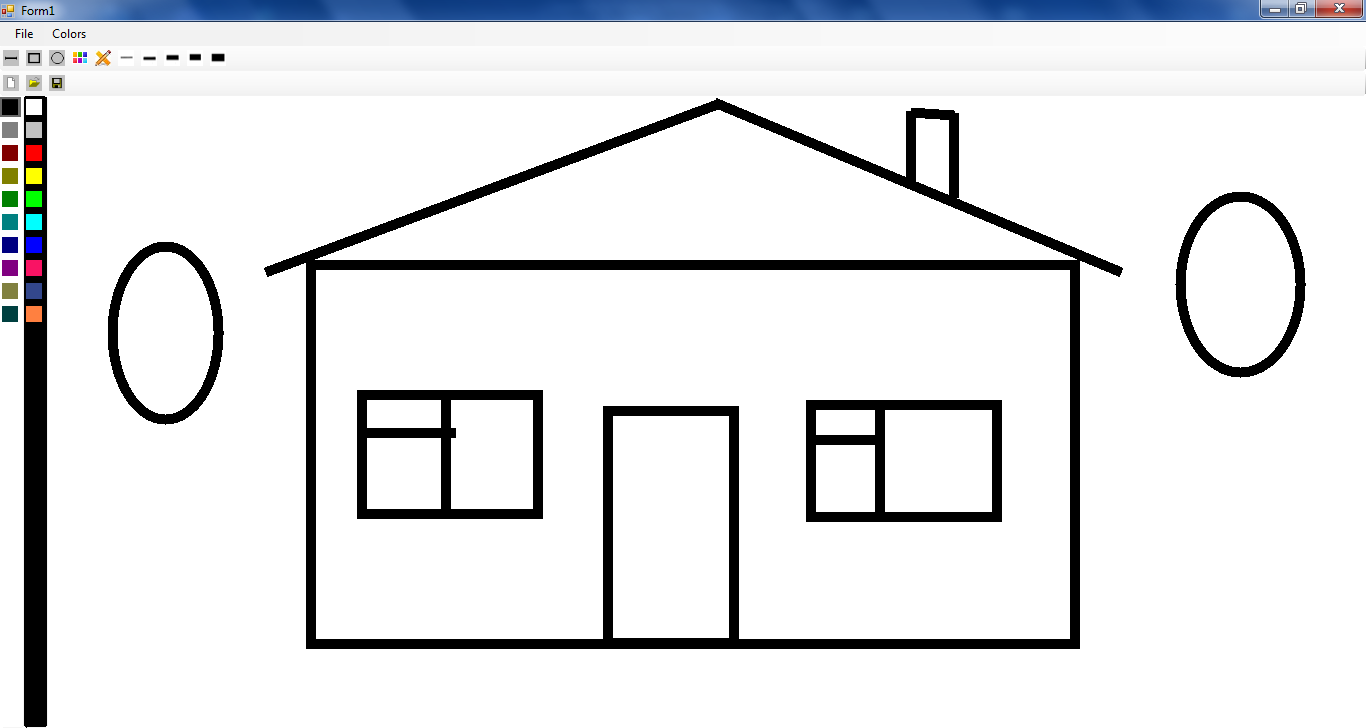


Fig. 2 Generarare imaginii

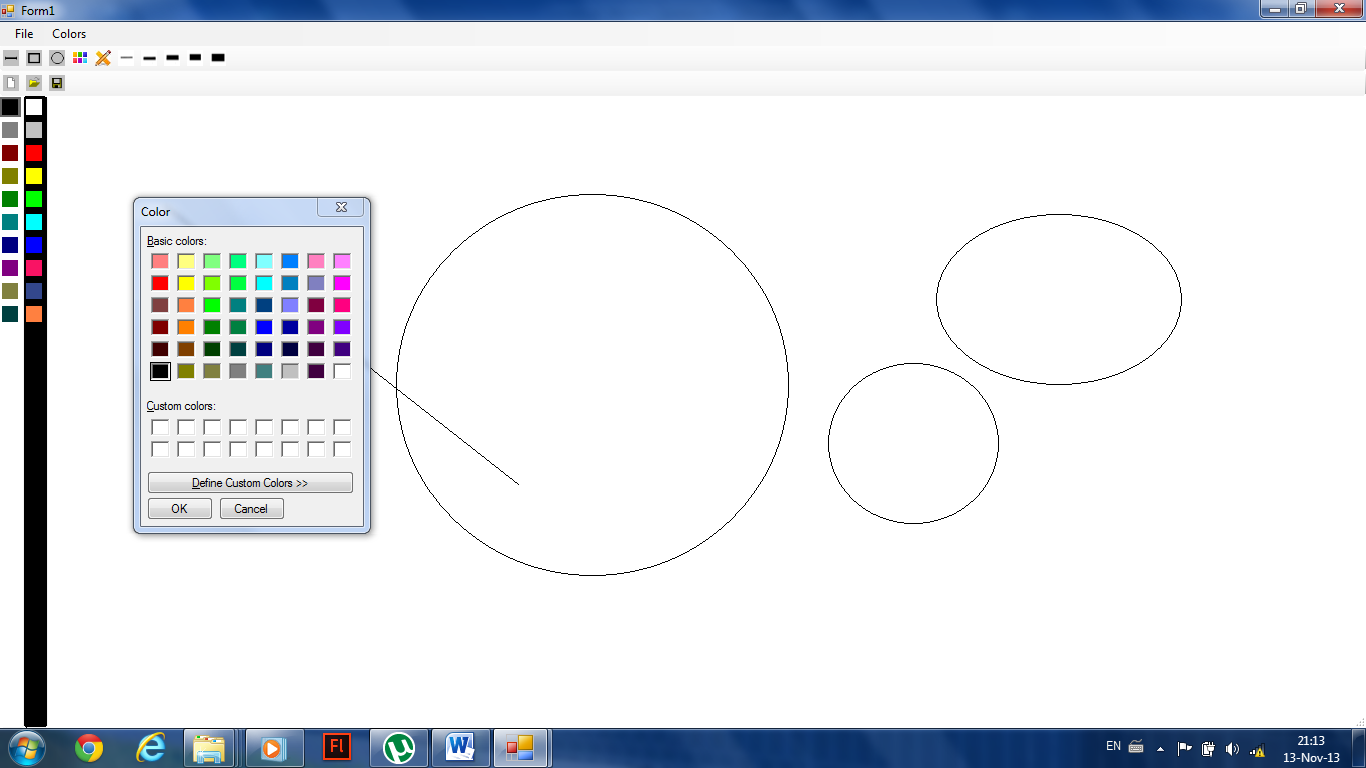


Fig. 3 Alegerea culorii pentru desenarea imaginii

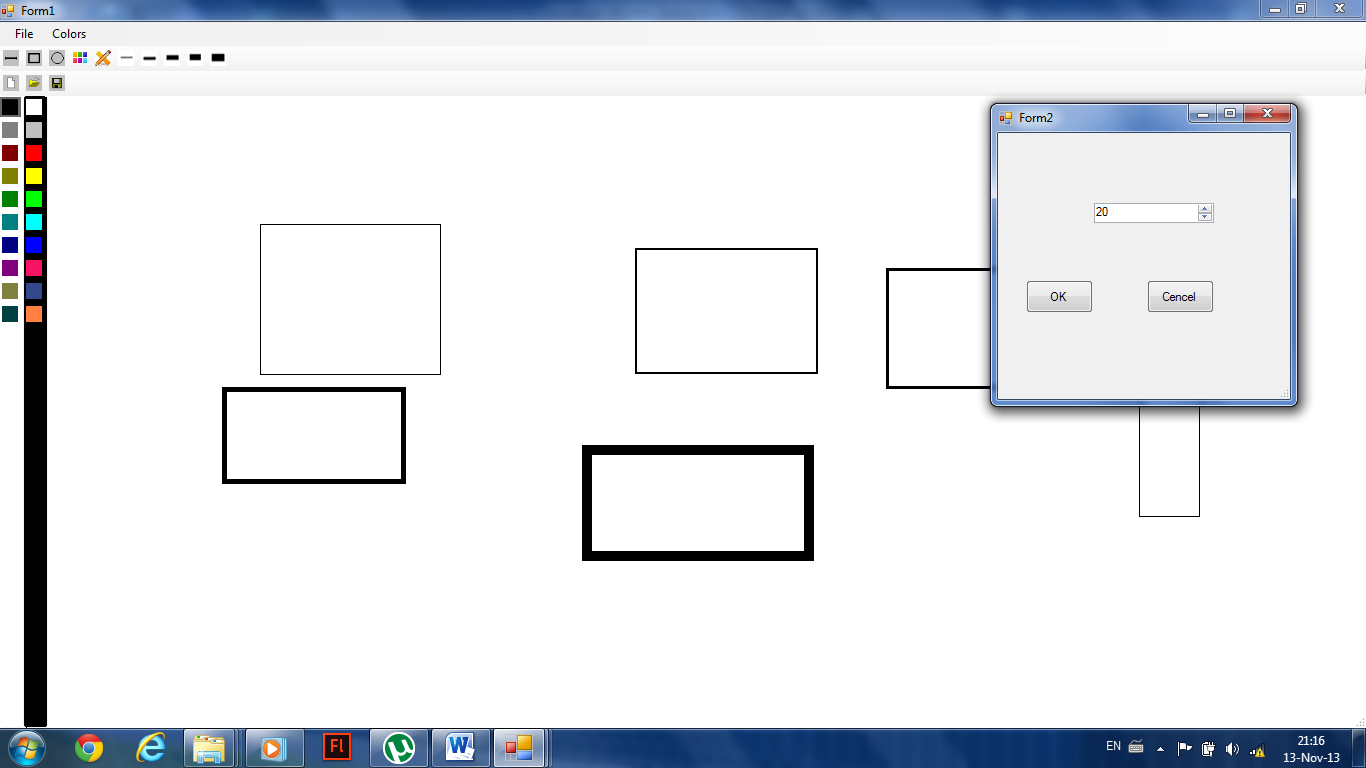


Fig. 4 Alegerea grosimii

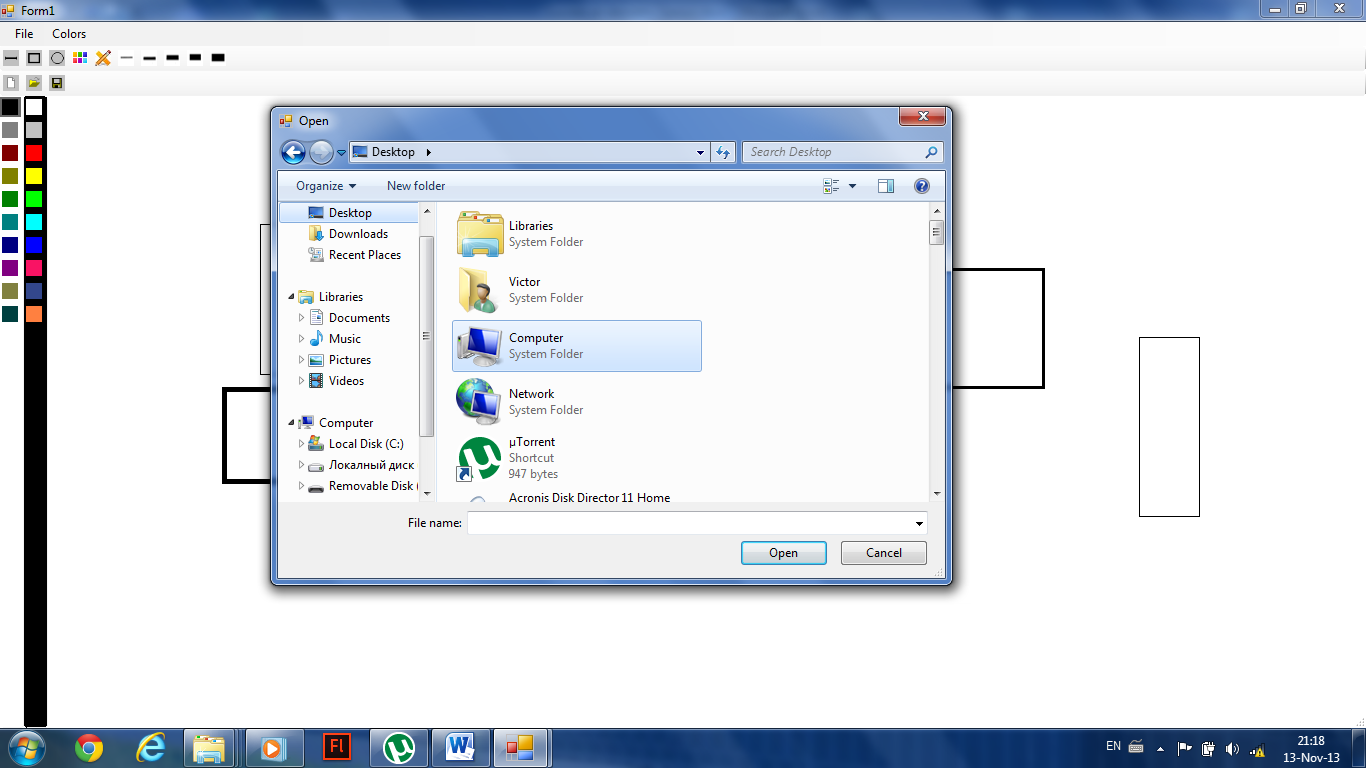


Fig. 5 Salvarea Imaginii

**Listingul Programului:**

1. Clasa principala de unde se incepe programul:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows.Forms;

namespace WindowsFormsApplication19

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

1. Clasa Form1

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.Collections;

using System.IO;

using System.Runtime.Serialization.Formatters.Binary;

using System.Drawing.Drawing2D;

namespace WindowsFormsApplication19

{

public partial class Form1 : Form

{

private ArrayList ds;

private Point mold;

private Point mcur;

private int mshape;

private float mwidth;

private Color mcolor;

public Form1()

{

InitializeComponent();

ds = new ArrayList();

mshape = 0;

mwidth = 1;

mcolor = Color.Black;

}

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

{

this.Cursor = Cursors.Cross;

if (e.Button == MouseButtons.Left)

{

mold = e.Location;

}

}

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

{

if (e.Button == MouseButtons.Left)

{

mcur = e.Location;

Invalidate();

}

}

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

{

this.Cursor = Cursors.Default;

Class1 a = new Class1(mold, mcur, mshape, mwidth, mcolor);

ds.Add(a);

}

private Rectangle rec(Point p1, Point p2)

{

Rectangle a = new Rectangle();

a.X = (p1.X > p2.X ? p2.X : p1.X);

a.Y = (p1.Y > p2.Y ? p2.Y : p1.Y);

a.Width = Math.Abs(p1.X - p2.X);

a.Height = Math.Abs(p1.Y - p2.Y);

return a;

}

private void Form1\_Paint(object sender, PaintEventArgs e)

{

foreach( Class1 a in ds)

{

draw(e.Graphics, a.old, a.cur, a.shape, a.width, a.color);

}

draw(e.Graphics, mold, mcur, mshape, mwidth, mcolor);

}

private void draw(Graphics e,Point mold,Point mcur,int mshape,float mwidth,Color mcolor)

{

Pen p=new Pen(mcolor, mwidth);

switch(mshape)

{

case 0:

e.DrawLine(p, mold, mcur);

break;

case 1:

e.DrawRectangle(p, rec(mold, mcur));

break;

case 2:

e.DrawEllipse(p, rec(mold, mcur));

break;

}

}

private void toolStripButton1\_Click(object sender, EventArgs e)

{

mshape = 0;

}

private void toolStripButton2\_Click(object sender, EventArgs e)

{

mshape = 1;

}

private void toolStripButton3\_Click(object sender, EventArgs e)

{

mshape = 2;

}

private void toolStripButton4\_Click(object sender, EventArgs e)

{

ColorDialog dlg = new ColorDialog();

dlg.Color = mcolor;

if (dlg.ShowDialog() == DialogResult.OK)

{

mcolor = dlg.Color;

}

}

private void toolStripButton5\_Click(object sender, EventArgs e)

{

Form2 dlg = new Form2();

dlg.width = mwidth;

dlg.ShowDialog();

mwidth = dlg.width;

}

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

{

SaveFileDialog dlg = new SaveFileDialog();

if (dlg.ShowDialog() == DialogResult.OK)

{

FileStream f = new FileStream(dlg.FileName, FileMode.Create);

BinaryFormatter bf = new BinaryFormatter();

bf.Serialize(f, ds);

}

}

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

{

OpenFileDialog dlg = new OpenFileDialog();

if (dlg.ShowDialog() == DialogResult.OK)

{

FileStream f = new FileStream(dlg.FileName, FileMode.Open);

BinaryFormatter bf = new BinaryFormatter();

ds = (ArrayList)bf.Deserialize(f);

Invalidate();

}

}

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

{

Form1 frm = new Form1();

frm.ShowDialog();

}

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

{

this.Close();

}

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

{

}

private void toolStripButton6\_Click(object sender, EventArgs e)

{

mwidth = 1;

}

private void toolStripButton7\_Click(object sender, EventArgs e)

{

mwidth = 2;

}

private void toolStripButton8\_Click(object sender, EventArgs e)

{

mwidth = 5;

}

private void toolStripButton9\_Click(object sender, EventArgs e)

{

mwidth = 7;

}

private void toolStripButton10\_Click(object sender, EventArgs e)

{

mwidth = 10;

}

private void toolStripButton15\_Click(object sender, EventArgs e)

{

mcolor = Color.Black;

}

private void toolStripButton19\_Click(object sender, EventArgs e)

{

mcolor = Color.White;

}

private void toolStripButton16\_Click(object sender, EventArgs e)

{

mcolor = Color.AliceBlue;

}

private void toolStripButton20\_Click(object sender, EventArgs e)

{

mcolor = Color.DarkGray;

}

private void toolStripButton17\_Click(object sender, EventArgs e)

{

mcolor = Color.DarkRed;

}

private void toolStripButton21\_Click(object sender, EventArgs e)

{

mcolor = Color.Red;

}

private void toolStripButton18\_Click(object sender, EventArgs e)

{

mcolor = Color.DarkOliveGreen;

}

private void toolStripButton14\_Click(object sender, EventArgs e)

{

mcolor = Color.DarkGreen;

}

private void toolStripButton24\_Click(object sender, EventArgs e)

{

mcolor = Color.DarkCyan;

}

private void toolStripButton25\_Click(object sender, EventArgs e)

{

mcolor = Color.DarkBlue;

}

private void toolStripButton26\_Click(object sender, EventArgs e)

{

mcolor = Color.DarkMagenta;

}

private void toolStripButton27\_Click(object sender, EventArgs e)

{

mcolor = Color.DarkKhaki;

}

private void toolStripButton28\_Click(object sender, EventArgs e)

{

mcolor = Color.DarkSlateGray;

}

private void toolStripButton22\_Click(object sender, EventArgs e)

{

mcolor = Color.Yellow;

}

private void toolStripButton23\_Click(object sender, EventArgs e)

{

mcolor = Color.Green;

}

private void toolStripButton29\_Click(object sender, EventArgs e)

{

mcolor = Color.DarkTurquoise;

}

private void toolStripButton30\_Click(object sender, EventArgs e)

{

mcolor = Color.Blue;

}

private void toolStripButton31\_Click(object sender, EventArgs e)

{

mcolor = Color.DeepPink;

}

private void toolStripButton33\_Click(object sender, EventArgs e)

{

mcolor = Color.DarkSalmon;

}

private void toolStripButton32\_Click(object sender, EventArgs e)

{

mcolor = Color.SlateBlue;

}

private void toolStripButton11\_Click(object sender, EventArgs e)

{

Form1 frm = new Form1();

frm.ShowDialog();

}

private void toolStripButton13\_Click(object sender, EventArgs e)

{

SaveFileDialog dlg = new SaveFileDialog();

if (dlg.ShowDialog() == DialogResult.OK)

{

FileStream f = new FileStream(dlg.FileName, FileMode.Create);

BinaryFormatter bf = new BinaryFormatter();

bf.Serialize(f, ds);

}

}

private void toolStripButton12\_Click(object sender, EventArgs e)

{

OpenFileDialog dlg = new OpenFileDialog();

if (dlg.ShowDialog() == DialogResult.OK)

{

FileStream f = new FileStream(dlg.FileName, FileMode.Open);

BinaryFormatter bf = new BinaryFormatter();

ds = (ArrayList)bf.Deserialize(f);

Invalidate();

}

}

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

{

}

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

{

ColorDialog dlg = new ColorDialog();

dlg.Color = mcolor;

if (dlg.ShowDialog() == DialogResult.OK)

{

mcolor = dlg.Color;

}

}

}

}

1. Clasa Class1

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Drawing;

namespace WindowsFormsApplication19

{

[Serializable]

class Class1

{

private Point mold;

private Point mcur;

private int mshape;

private float mwidth;

private Color mcolor;

public Class1()

{

mold = new Point(0, 0);

mcur = new Point(0, 0);

mshape = 0;

mwidth = 1;

mcolor = Color.Black;

}

public Class1(Point old, Point cur, int shape, float width, Color color)

{

mold = old;

mcur = cur;

mshape = shape;

mwidth = width;

mcolor = color;

}

public Point old

{

get

{

return mold;

}

set

{

mold = value;

}

}

public Point cur

{

get

{

return mcur;

}

set

{

mcur = value;

}

}

public int shape

{

get

{

return mshape;

}

set

{

mshape = value;

}

}

public float width

{

get

{

return mwidth;

}

set

{

mwidth = value;

}

}

public Color color

{

get

{

return mcolor;

}

set

{

mcolor = value;

}

}

}

}

**Concluzii:**