Windows lietojumprogrammu izstrāde

1. piemērs

```
1.    using System.Windows.Forms;
2.    class HelloForm
3.    {
4.          public static void Main()
5.          {
6.                Form form = new Form();
7.                form.Text = "Hello!";
8.                Application.Run(form);
9.          }
10.   }
```

2.piemērs

```
1.    using System;
2.    using System.Drawing;
3.    using System.Windows.Forms;
4.    class InheritForm : Form
5.    {
6.          public static void Main()
7.          {
8.                Application.Run(new InheritForm());
9.          }
10.         public InheritForm()
11.         {
12.               Text = "Inherit with constructor";
13.               BackColor = Color.White;
14.         }
15.   }
```

3. piemērs

```
1.    Form2 ModDialog = new Form2();
2.    ModDialog.ShowDialog();
3.    if (ModDialog.DialogResult == DialogResult.OK)
4.          MessageBox.Show("Processing request");
5.    else if (ModDialog.DialogResult == DialogResult.Cancel)
6.          MessageBox.Show("Cancelling request");
7.    ModDialog.Dispose();
```

4. piemērs

```
1.    using System;
2.    using System.Drawing;
3.    using System.Windows.Forms;
4.    class PaintEvent
5.    {
6.          public static void Main()
7.          {
8.                Form form = new Form();
9.                form.Text = "Paint Event";
10.               form.Paint += new PaintEventHandler(MyPaintHandler);
11.               Application.Run(form);
12.         }
13.         static void MyPaintHandler(object objSender, PaintEventArgs pea)
14.         {
```

```
15.                 Graphics grfx = pea.Graphics;
16.                 grfx.Clear(Color.Chocolate);
17.           }
18.     }
```

5. piemērs

```
1.      using System;
2.      using System.Drawing;
3.      using System.Windows.Forms;
4.      class HelloWorld : Form
5.      {
6.            public static void Main()
7.            {
8.                  Application.Run(new HelloWorld());
9.            }
10.           public HelloWorld()
11.           {
12.                 Text = "Hello World";
13.                 BackColor = Color.White;
14.           }
15.           protected override void OnPaint(PaintEventArgs pea)
16.           {
17.                 Graphics grfx = pea.Graphics;
18.                 grfx.DrawString("Hello!", Font, Brushes.Black, 0, 0);
19.           }
20.     }
```

6. piemērs

```
1.      using System;
2.      using System.Drawing;
3.      using System.Windows.Forms;
4.      class MyForm1 : Form
5.      {
6.            public static void Main()
7.            {
8.                  MyForm1 frm = new MyForm1();
9.                  frm.Text = "1";
10.                 frm.Show();
11.                 Application.Run();
12.           }
13.           protected override void OnClosed(EventArgs e)
14.           {
15.                 MyForm2 frm = new MyForm2();
16.                 frm.Text = "2";
17.                 frm.Show();
18.           }
19.     }
20.     class MyForm2 : Form
21.     {
22.           protected override void OnClosed(EventArgs e)
23.           {
24.                 Application.Exit();
25.           }
26.     }
```

7. piemērs

```
1.      using System;
2.      using System.Drawing;
```

```
3.    using System.Windows.Forms;
4.    class SimpleDialog : Form
5.    {
6.          string strDisplay = "";
7.          public static void Main()
8.          {
9.                Application.Run(new SimpleDialog());
10.         }
11.         public SimpleDialog()
12.         {
13.               Text = "SimpleDialog";
14.               Menu = new MainMenu();
15.               Menu.MenuItems.Add("Dialog!", new EventHandler(MenuOnClick));
16.         }
17.         void MenuOnClick(object obj, EventArgs ea)
18.         {
19.               SimpleDialogBox dlg = new SimpleDialogBox();
20.               dlg.ShowDialog();
21.               strDisplay = "Dialog box terminated with " +
22.                     dlg.DialogResult + "!";
23.               Invalidate();
24.         }
25.         protected override void OnPaint(PaintEventArgs pea)
26.         {
27.               Graphics grfx = pea.Graphics;
28.               grfx.DrawString(strDisplay, Font, new SolidBrush(ForeColor), 0, 0);
29.         }
30.   }
31.   class SimpleDialogBox : Form
32.   {
33.         public SimpleDialogBox()
34.         {
35.               Text = "Simple Dialog Box";
36.               FormBorderStyle = FormBorderStyle.FixedDialog;
37.               ControlBox = false;
38.               MaximizeBox = false;
39.               MinimizeBox = false;
40.               ShowInTaskbar = false;
41.               Button btn = new Button();
42.               btn.Parent = this;
43.               btn.Text = "OK";
44.               btn.Location = new Point(50,50);
45.               btn.Size = new Size(10*Font.Height, 2*Font.Height);
46.               btn.Click += new EventHandler(ButtonOkOnClick);
47.               btn = new Button();
48.               btn.Parent = this;
49.               btn.Text = "Cancel";
50.               btn.Location = new Point(50,100);
51.               btn.Size = new Size(10*Font.Height, 2*Font.Height);
52.               btn.Click += new EventHandler(ButtonCancelOnClick);
53.         }
54.         void ButtonOkOnClick(object obj, EventArgs ea)
55.         {
56.               DialogResult = DialogResult.OK;
57.         }
58.         void ButtonCancelOnClick(object obj, EventArgs ea)
59.         {
60.               DialogResult = DialogResult.Cancel;
61.         }
62.   }
```

8. piemērs

```
1.    using System;
2.    using System.Drawing;
```

```
3.    using System.Windows.Forms;
4.    class MyMainForm : Form
5.    {
6.          public static void Main()
7.          {
8.                Application.Run(new MyMainForm());
9.          }
10.         public MyMainForm()
11.         {
12.               Text = "My Main Form";
13.               Menu = new MainMenu();
14.               Menu.MenuItems.Add("Options");
15.               Menu.MenuItems[0].MenuItems.Add("Title..",
16.                     new EventHandler(MenuTitleOnClick));
17.         }
18.         void MenuTitleOnClick(object obj, EventArgs ea)
19.         {
20.               MyDlg dlg = new MyDlg();
21.               dlg.Owner = this;
22.               dlg.Changed += new EventHandler(TitleOnChanged);
23.               dlg.Show();
24.         }
25.         void TitleOnChanged(object obj, EventArgs ea)
26.         {
27.               MyDlg dlg = (MyDlg)obj;
28.               Text = dlg.MyTitle;
29.         }
30.   }
31.   class MyDlg : Form
32.   {
33.         public string MyTitle;
34.         public event EventHandler Changed;
35.         private TextBox tb;
36.         public MyDlg()
37.         {
38.               Text = "My Modeless Dialog Box";
39.               MyTitle = "";
40.               ControlBox = false;
41.               MaximizeBox = false;
42.               MinimizeBox = false;
43.               ShowInTaskbar = false;
44.               tb = new TextBox();
45.               tb.Parent = this;
46.               tb.Location = new Point(10,10);
47.               Button btn = new Button();
48.               btn.Parent = this;
49.               btn.Text = "Apply";
50.               btn.Location = new Point(10,50);
51.               btn.Size = new Size(10*Font.Height, 2*Font.Height);
52.               btn.Click += new EventHandler(ButtonApplyOnClick);
53.               btn = new Button();
54.               btn.Parent = this;
55.               btn.Text = "Close";
56.               btn.Location = new Point(155,50);
57.               btn.Size = new Size(10*Font.Height, 2*Font.Height);
58.               btn.Click += new EventHandler(ButtonCloseOnClick);
59.         }
60.         void ButtonApplyOnClick(object obj, EventArgs ea)
61.         {
62.               MyTitle = tb.Text;
63.               if (Changed != null)
64.                     Changed(this, new EventArgs());
65.         }
66.         void ButtonCloseOnClick(object obj, EventArgs ea)
67.         {
68.               this.Close();
```

```
69.          }
70.    }
```

9. piemērs

```
1.     if (MessageBox.Show("Continue?","Question",
2.     MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
3.           DialogResult.Yes)
4.     { ...  }
```

10. piemērs

```
1.     OpenFileDialog fdlg = new OpenFileDialog();
2.     fdlg.Title = "Open Image";
3.     fdlg.InitialDirectory = "C:\\";
4.     fdlg.CheckFileExists = true;
5.     fdlg.Filter = "Bitmap files (*.bmp)|*.bmp|" +
6.           "Jpeg files (*.jpg)|*.jpg|" +
7.           "All valid files (*.bmp;*.jpg)|*.bmp;*.jpg";
8.     fdlg.FilterIndex = 2;
9.     if (fdlg.ShowDialog() == DialogResult.OK)
10.          try
11.          {
12.                pictureBox1.Image = Image.FromFile(fdlg.FileName);
13.          }
14.          catch (Exception ex)
15.          {
16.                MessageBox.Show("Error");
17.          }
```

11. piemērs

```
1.     using System;
2.     using System.Drawing;
3.     using System.Windows.Forms;
4.     class MyForm: Form
5.     {
6.           static public void Main()
7.           {
8.                 Application.Run(new MyForm());
9.           }
10.          public MyForm()
11.          {
12.                Paint += new PaintEventHandler(MyPaintHandler);
13.                Click += new EventHandler(MyClickHandler);
14.                FormBorderStyle = FormBorderStyle.None;
15.          }
16.          static void MyPaintHandler(object objSender, PaintEventArgs pea)
17.          {
18.                MyForm frm = (MyForm)objSender;
19.                frm.BackColor = Color.Blue;
20.                System.Drawing.Drawing2D.GraphicsPath gp =
21.                      new System.Drawing.Drawing2D.GraphicsPath();
22.                Rectangle rec = new Rectangle(0, 0,
23.                      frm.ClientSize.Width, frm.ClientSize.Height);
24.                gp.StartFigure();
25.                gp.AddPie(rec, 0, 90);
26.                gp.AddPie(rec, 180, 90);
27.                gp.CloseFigure();
28.                frm.Region = new Region(gp);
29.          }
30.          static void MyClickHandler(object objSender, EventArgs ea)
31.          {
```

```
32.                 MyForm frm = (MyForm)objSender;
33.                 frm.Close();
34.           }
35.     }
```

12. piemērs

```
1.      using System;
2.      using System.Windows.Forms;
3.      class MDIMain : Form
4.      {
5.            public static void Main()
6.            {
7.                  Application.Run(new MDIMain());
8.            }
9.            private int WindowCount = 0;
10.           public MDIMain()
11.           {
12.                 this.IsMdiContainer = true;
13.                 this.WindowState = FormWindowState.Maximized;
14.                 MenuItem iNewDoc = new MenuItem("New",
15.                       new EventHandler(MenuNewDoc));
16.                 MenuItem iClose = new MenuItem("Close",
17.                       new EventHandler(MenuClose));
18.                 MenuItem iExit = new MenuItem("Exit",
19.                       new EventHandler(MenuExit));
20.                 MenuItem iFile = new MenuItem("File",
21.                       new MenuItem[] {iNewDoc, iClose, iExit});
22.                 MenuItem iCascade = new MenuItem("Cascade",
23.                       new EventHandler(MenuCascade));
24.                 MenuItem iTile = new MenuItem("Tile",
25.                       new EventHandler(MenuTitle));
26.                 MenuItem iWindow = new MenuItem("Window",
27.                       new MenuItem [] {iCascade, iTile});
28.                 iWindow.MdiList = true;
29.                 this.Menu = new MainMenu(new MenuItem [] {iFile, iWindow});
30.                 MenuNewDoc(this, new EventArgs());
31.           }
32.           void MenuNewDoc(object obj, EventArgs ea)
33.           {
34.                 Form frm = new Form();
35.                 frm.MdiParent = this;
36.                 frm.Text = "Form " + WindowCount.ToString();
37.                 frm.Show();
38.                 WindowCount++;
39.           }
40.           void MenuClose(object obj, EventArgs ea)
41.           {
42.                 this.ActiveMdiChild.Close();
43.           }
44.           private void MenuExit(object obj, EventArgs ea)
45.           {
46.                 Close();
47.           }
48.           private void MenuCascade(object obj, EventArgs ea)
49.           {
50.                 this.LayoutMdi(MdiLayout.Cascade);
51.           }
52.           private void MenuTitle(object obj, EventArgs ea)
53.           {
54.                 this.LayoutMdi(MdiLayout.TileHorizontal);
55.           }
56.     }
```

13. piemērs

Darbība	Notikums	Īpašības		
		KeyCode	Modifiers	KeyData
nospiests <Shift>	KeyDown	Keys.ShiftKey	Keys.Shift	Keys.Shift | Keys.ShiftKey
nospiests <D>	KeyDown	Keys.D	Keys.Shift	Keys.Shift | Keys.D
atlaists <D>	KeyUp	Keys.D	Keys.Shift	Keys.Shift | Keys.D
atlaists <Shift>	KeyUp	Keys.ShiftKey	Keys.None	Keys.ShiftKey

14. piemērs

```
1.    using System;
2.    using System.Drawing;
3.    using System.Windows.Forms;
4.
5.    class CaptureLossNotifyWindow: NativeWindow
6.    {
7.           public DrawRect control;
8.           protected override void WndProc(ref Message message)
9.           {
10.                 if (message.Msg == 533) // WM_CAPTURECHANGED
11.                       control.OnLostCapture();
12.                 base.WndProc(ref message);
13.          }
14.   }
15.
16.   class DrawRect: Form
17.   {
18.          bool bBlocking, bValidBox;
19.          Point ptBeg, ptEnd;
20.          Rectangle rectBox;
21.
22.          public static void Main()
23.          {
24.                 Application.Run(new DrawRect());
25.          }
26.          public DrawRect()
27.          {
28.                 Text = "Draw Rectangle";
29.                 BackColor = SystemColors.Window;
30.                 ForeColor = SystemColors.WindowText;
31.                 CaptureLossNotifyWindow win =
32.                       new CaptureLossNotifyWindow();
33.                 win.control = this;
34.                 win.AssignHandle(Handle);
35.          }
36.          protected override void OnMouseDown(MouseEventArgs mea)
37.          {
38.                 if (mea.Button == MouseButtons.Left)
39.                 {
40.                       ptBeg = ptEnd = new Point(mea.X, mea.Y);
41.                       Graphics grfx = CreateGraphics();
42.                       grfx.DrawRectangle(new Pen(ForeColor),
43.                             Rect(ptBeg, ptEnd));
44.                       grfx.Dispose();
45.                       bBlocking = true;
46.                 }
47.          }
48.          protected override void OnMouseMove(MouseEventArgs mea)
49.          {
50.                 if (bBlocking)
51.                 {
```

```
52.                         Graphics grfx = CreateGraphics();
53.                         grfx.DrawRectangle(new Pen(BackColor),
54.                               Rect(ptBeg, ptEnd));
55.                         ptEnd = new Point(mea.X, mea.Y);
56.                         grfx.DrawRectangle(new Pen(ForeColor),
57.                               Rect(ptBeg, ptEnd));
58.                         grfx.Dispose();
59.                         Invalidate();
60.                   }
61.             }
62.             protected override void OnMouseUp(MouseEventArgs mea)
63.             {
64.                   if (bBlocking)
65.                   {
66.                         Graphics grfx = CreateGraphics();
67.                         rectBox = Rect(ptBeg, new Point(mea.X, mea.Y));
68.                         grfx.DrawRectangle(new Pen(ForeColor), rectBox);
69.                         grfx.Dispose();
70.                         bBlocking = false;
71.                         bValidBox = true;
72.                         Invalidate();
73.                   }
74.             }
75.             protected override void OnPaint(PaintEventArgs pea)
76.             {
77.                   Graphics grfx = pea.Graphics;
78.                   if (bValidBox)
79.                         grfx.FillRectangle(new SolidBrush(ForeColor), rectBox);
80.                   if (bBlocking)
81.                         grfx.DrawRectangle(new Pen(ForeColor), Rect(ptBeg, ptEnd));
82.             }
83.             Rectangle Rect(Point ptBeg, Point ptEnd)
84.             {
85.                   return new Rectangle(Math.Min(ptBeg.X, ptEnd.X),
86.                         Math.Min(ptBeg.Y, ptEnd.Y),
87.                         Math.Abs(ptEnd.X - ptBeg.X),
88.                         Math.Abs(ptEnd.Y - ptBeg.Y));
89.             }
90.             public void OnLostCapture()
91.             {
92.                   if (bBlocking)
93.                   {
94.                         Graphics grfx = CreateGraphics();
95.                         grfx.DrawRectangle(new Pen(BackColor), Rect(ptBeg, ptEnd));
96.                         grfx.Dispose();
97.                         bBlocking = false;
98.                         Invalidate();
99.                   }
100.            }
101.      }
```

15. piemērs

```
1.    using System;
2.    using System.Drawing;
3.    using System.Windows.Forms;
4.    class TwoStBarPanels : Form
5.    {
6.          public static void Main()
7.          {
8.                Application.Run(new TwoStBarPanels());
9.          }
10.         public TwoStBarPanels()
11.         {
```

```
12.                 Text = "Two Status Bar Panels";
13.                 BackColor = SystemColors.Window;
14.                 ForeColor = SystemColors.WindowText;
15.                 StatusBar sb = new StatusBar();
16.                 sb.Parent = this;
17.                 sb.ShowPanels = true;
18.                 StatusBarPanel sbp1 = new StatusBarPanel();
19.                 sbp1.Text = "Panel1";
20.                 StatusBarPanel sbp2 = new StatusBarPanel();
21.                 sbp2.Text = "Panel2";
22.                 sb.Panels.Add(sbp1);
23.                 sb.Panels.Add(sbp2);
24.           }
25.    }
```

16. piemērs

```
1.     using System;
2.     using System.Drawing;
3.     using System.Drawing.Drawing2D;
4.     using System.Windows.Forms;
5.
6.     class Bounce : Form
7.     {
8.           const int iTimerInterval = 100;
9.           const int iBallSize = 16;
10.          const int iMoveSize = 4;
11.
12.          Bitmap bitmap;
13.          int xCenter, yCenter;
14.          int cxRadius, cyRadius, cxMove, cyMove, cxTotal, cyTotal;
15.
16.          public static void Main()
17.          {
18.                Application.Run(new Bounce());
19.          }
20.
21.          public Bounce()
22.          {
23.                Text = "Bounce";
24.                ResizeRedraw = true;
25.                BackColor = Color.White;
26.
27.                Timer timer = new Timer();
28.                timer.Interval = iTimerInterval;
29.                timer.Tick += new EventHandler(TimerOnTick);
30.                timer.Start();
31.
32.                OnResize(EventArgs.Empty);
33.          }
34.          protected override void OnResize(EventArgs ea)
35.          {
36.                Graphics grfx = CreateGraphics();
37.                grfx.Clear(BackColor);
38.
39.                float fRadius = Math.Min(ClientSize.Width / grfx.DpiX,
40.                      ClientSize.Height / grfx.DpiY) / iBallSize;
41.
42.                cxRadius = (int) (fRadius * grfx.DpiX);
43.                cyRadius = (int) (fRadius * grfx.DpiY);
44.
45.                grfx.Dispose();
46.                cxMove = Math.Max(1, cxRadius / iMoveSize);
47.                cyMove = Math.Max(1, cyRadius / iMoveSize);
48.
```

```
49.                 cxTotal = 2 * (cxRadius + cxMove);
50.                 cyTotal = 2 * (cyRadius + cyMove);
51.
52.                 bitmap = new Bitmap(cxTotal, cyTotal);
53.
54.                 grfx = Graphics.FromImage(bitmap);
55.                 grfx.Clear(BackColor);
56.
57.                 DrawBall(grfx, new Rectangle(cxMove, cyMove,
58.                       2 * cxRadius, 2 * cyRadius));
59.
60.                 grfx.Dispose();
61.
62.                 xCenter = ClientSize.Width / 2;
63.                 yCenter = ClientSize.Height / 2;
64.           }
65.
66.           protected void DrawBall(Graphics grfx, Rectangle rect)
67.           {
68.                 GraphicsPath path = new GraphicsPath();
69.                 path.AddEllipse(rect);
70.
71.                 PathGradientBrush pgbrush = new PathGradientBrush(path);
72.                 pgbrush.CenterPoint = new PointF((rect.Left + rect.Right) / 3,
73.                       (rect.Top + rect.Bottom) / 3);
74.
75.                 pgbrush.CenterColor = Color.White;
76.                 pgbrush.SurroundColors = new Color[] { Color.Red };
77.
78.                 grfx.FillRectangle(pgbrush, rect);
79.           }
80.
81.           void TimerOnTick(object obj, EventArgs ea)
82.           {
83.                 Graphics grfx = CreateGraphics();
84.
85.                 grfx.DrawImage(bitmap, xCenter - cxTotal / 2,
86.                       yCenter - cyTotal / 2, cxTotal, cyTotal);
87.
88.                 grfx.Dispose();
89.
90.                 xCenter += cxMove;
91.                 yCenter += cyMove;
92.
93.                 if ((xCenter + cxRadius >= ClientSize.Width) ||
94.                       (xCenter - cxRadius <= 0))
95.                       cxMove = -cxMove;
96.
97.                 if ((yCenter + cyRadius >= ClientSize.Height) ||
98.                       (yCenter - cyRadius <= 0))
99.                       cyMove = -cyMove;
100.          }
101.    }
```