**24 Меню и панели инструментов**

Задание 1. Создание и использование элемента управления ToolStrip. Использование элемента управления StatusStrip.

Листинг программы:

namespace Task6

{

public partial class ParentForm : Form

{

private int openDocs = 0;

public ParentForm()

{

InitializeComponent();

spData.Text = Convert.ToString(DateTime.Today.ToLongDateString());

}

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

{

this.Close();

}

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

{

this.LayoutMdi(MdiLayout.Cascade);

spWin.Text = "Window is Cascade";

}

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

{

this.LayoutMdi(MdiLayout.TileHorizontal);

spWin.Text = "Window is TileHorizontal";

}

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

{

ChildForm childForm = new ChildForm();

childForm.MdiParent = this;

childForm.Text = childForm.Text + " " + ++openDocs;

childForm.Show();

}

private void toolStrip1\_ItemClicked(object sender, ToolStripItemClickedEventArgs e)

{

switch (e.ClickedItem.Tag.ToString())

{

case "NewDoc":

ChildForm newChild = new ChildForm();

newChild.MdiParent = this;

newChild.Show();

newChild.Text = newChild.Text + " " +

++openDocs;

break;

case "Cascade":

this.LayoutMdi(MdiLayout.Cascade);

spWin.Text = "Window is Cascade";

break;

case "Title":

this.LayoutMdi

(MdiLayout.TileHorizontal);

spWin.Text = "Window is TileHorizontal";

break;

}

}

}

}

Таблица 24.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| ⠀ |  |

Анализ результатов:

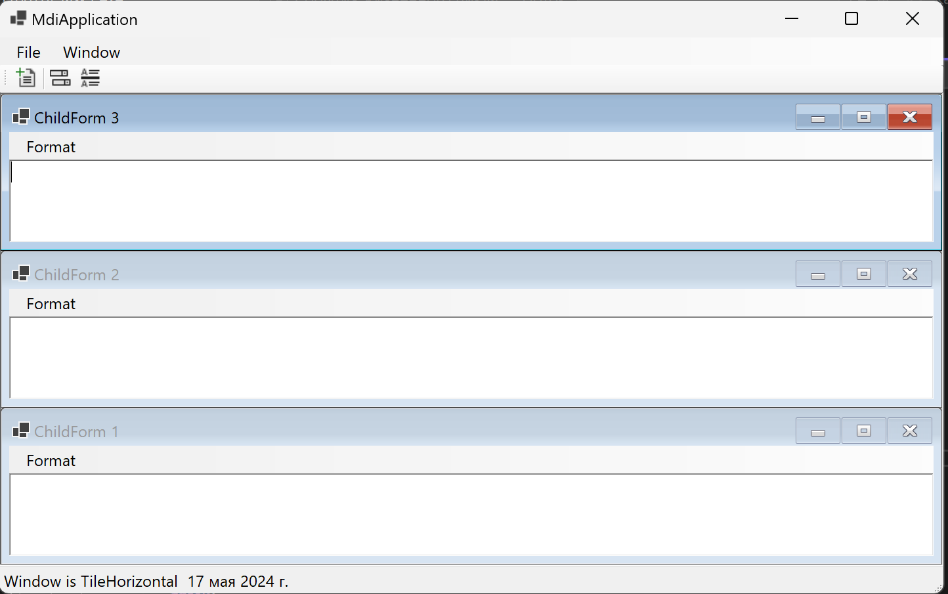


Рисунок 24.1 – Результат работы программы

Задание 2. Создать составной, специализированный и расширенный элемент управления.

Листинг программы:

namespace Task2

{

public partial class UserControlTimer : UserControl

{

public UserControlTimer()

{

InitializeComponent();

}

public bool TimeEnabled

{

get { return timer1.Enabled; }

set { timer1.Enabled = value; }

}

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

{

label1.Text = DateTime.Now.ToLongTimeString();

}

}

}

namespace Task2

{

public partial class UserControlTimer2 : UserControl

{

public UserControlTimer2()

{

InitializeComponent();

}

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

{

Refresh();

}

protected override void OnPaint(PaintEventArgs pe)

{

base.OnPaint(pe);

Graphics g = pe.Graphics;

g.FillRectangle(Brushes.Blue, 0, 0, Width, Height);

pe.Graphics.DrawString(DateTime.Now.ToLongTimeString(), Font, new SolidBrush(ForeColor), 0, 0);

}

}

}

namespace Task2

{

public class ClickButton : Button

{

int mClicks;

public int Clicks

{

get { return mClicks; }

}

protected override void OnClick(EventArgs e)

{

mClicks++;

base.OnClick(e);

}

protected override void OnPaint(PaintEventArgs pevent)

{

base.OnPaint(pevent);

Graphics g = pevent.Graphics;

SizeF stringsize;

stringsize = g.MeasureString(Clicks.ToString(), Font, Width);

g.DrawString(Clicks.ToString(), Font, SystemBrushes.ControlText, Width - stringsize.Width - 3, Height - stringsize.Height - 3);

}

}

}

Таблица 24.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| ⠀ |  |

Анализ результатов:

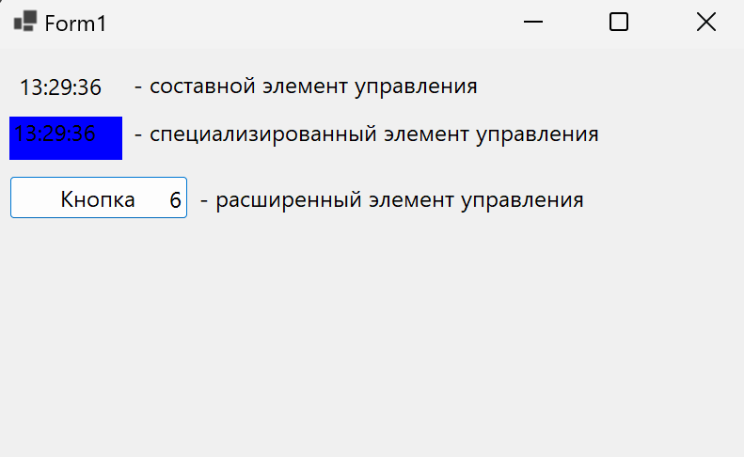


Рисунок 24.2 – Результат работы программы