**24 МЕНЮ И ПАНЕЛИ ИНСТРУМЕНТОВ**

Задание №1. Выполнить упражнения 3 и 4 из лабораторной работы №2 учебного пособия: Осипов Н.А. Разработка Windows приложений на C#.

Листинг программы:

private int openDocument = 0;

public Form1()

{

InitializeComponent();

spdata.Text = Convert.ToString(System.DateTime.Today.ToLongDateString());

}

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

{

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

{

case "NewDoc":

Child\_Form newChild = new Child\_Form();

newChild.MdiParent = this;

++openDocument;

newChild.Text = newChild.Text + " " + openDocument.ToString();

newChild.Show();

break;

case "Cascade":

this.LayoutMdi(System.Windows.Forms.MdiLayout.Cascade);

spWin.Text = "Windows is cascade";

break;

case "Title":

this.LayoutMdi

(System.Windows.Forms.MdiLayout.TileHorizontal);

spWin.Text = "Windows is horizontal";

break;

}

}

Таблица 24.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Дочерняя форма |

Результат работы программы представлен на рисунке 24.1.

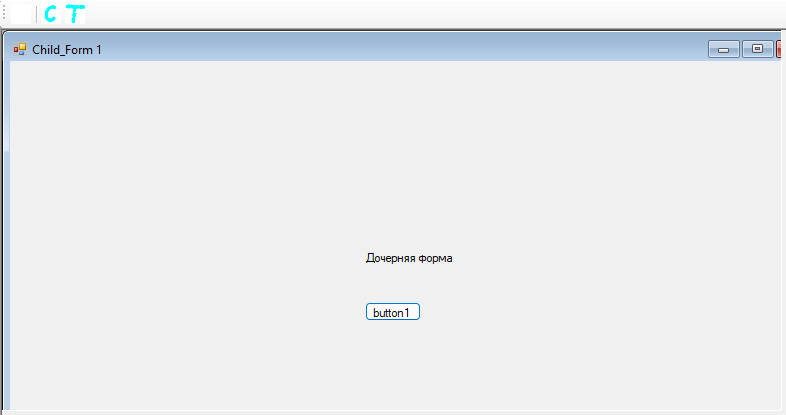


Рисунок 24.1 – Результат работы программы

Источник: собственная разработка

Задание №2. Выполнить лабораторную работу №3 из учебного пособия: Осипов Н.А. Разработка Windows приложений на C#.

Листинг программы:

public bool TimeEnabled

{

get { return timer1.Enabled; }

set { timer1.Enabled = value; }

}

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

{

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

}

Таблица 24.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 1, 10, 1, 1, 10 | График функции |

Результат работы программы представлен на рисунке 24.2.



Рисунок 24.2 – Результат работы программы

Источник: собственная разработка

Задание №3. Разработать собственный компонент и продемонстрировать его функциональность в приложении.

Листинг программы:

public int value = 10;

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

{

value += 1;

this.Refresh();

}

public bool TimeEnabled

{

get { return timer1.Enabled; }

set { timer1.Enabled = value; }

}

protected override void OnPaint(PaintEventArgs pe)

{

base.OnPaint(pe);

Graphics g = pe.Graphics;

g.FillRectangle(Brushes.Black, 0, 0, value, value);

}

Таблица 24.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 10 | Прямоугольник |

Результат работы компонента представлен на рисунке 24.3.

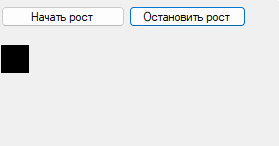


Рисунок 24.3 – Результат работы компонента

Источник: собственная разработка

Задание №4. Выполнить лабораторную работу №8 из учебного пособия: Осипов Н.А. Разработка Windows приложений на C#.

Листинг программы:

public Form1()

{

Localizer.LocaleChanged += Localizer\_LocaleChanged;

Thread.CurrentThread.CurrentUICulture = Thread.CurrentThread.CurrentCulture;

InitializeComponent();

}

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

{

YearDropBox.Items.Clear();

var thisYear = DateTime.Now.Year;

for (int i = 1900; i <= thisYear; i++)

{

YearDropBox.Items.Add(i.ToString());

}

}

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

{

ClearFields(this);

}

private void button6\_Click(object sender, EventArgs e)

{

Application.Exit();

}

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

{

File.Delete(@"hr\_info.txt");

string path = @"hr\_info.txt";

SaveInfo(this, path);

}

private static void ClearFields(Control parent)

{

foreach (Control c in parent.Controls)

{

if (c is TextBox)

{

((TextBox)c).Clear();

}

if (c.HasChildren)

{

ClearFields(c);

}

if (c is CheckBox)

{

((CheckBox)c).Checked = false;

}

if (c is RadioButton)

{

((RadioButton)c).Checked = false;

}

if (c is ComboBox)

{

((ComboBox)c).Text = "";

}

}

}

private static void SaveInfo(Control parent, string path)

{

foreach (Control c in parent.Controls)

{

if (c is TextBox txt)

{

File.AppendAllText(path, txt.Name + ": " + txt.Text + "\n");

}

if (c.HasChildren)

{

if (!(c is NumericUpDown))

{

SaveInfo(c, path);

}

}

if (c is CheckBox)

{

if (((CheckBox)c).Checked)

{

File.AppendAllText(path, c.Text + ": " + ((CheckBox)c).Checked.ToString() + "\n");

}

}

if (c is RadioButton)

{

if (((RadioButton)c).Checked)

{

File.AppendAllText(path, c.Text + ": " + ((RadioButton)c).Checked.ToString() + "\n");

}

}

if (c is ComboBox)

{

File.AppendAllText(path, c.Name + ": " + ((ComboBox)c).Text + "\n");

}

if (c is NumericUpDown)

{

File.AppendAllText(path, c.Name + ": " + ((NumericUpDown)c).Text + "\n");

}

}

}

public static class Localizer

{

public static event EventHandler LocaleChanged;

public static void SetLocale(string name)

{

Thread.CurrentThread.CurrentUICulture = new System.Globalization.CultureInfo(name);

LocaleChanged?.Invoke(null, EventArgs.Empty);

}

}

private void Localizer\_LocaleChanged(object sender, EventArgs e)

{

Controls.Clear();

InitializeComponent();

}

private void button7\_Click(object sender, EventArgs e)

{

Help.ShowHelp(this, helpProvider1.HelpNamespace);

}

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

{

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

Localizer.SetLocale("ru-RU");

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

Localizer.SetLocale("en-EN");

}

}

Таблица 24.4 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Набор данных | Набор данных |

Результат работы программы представлен на рисунке 24.4.

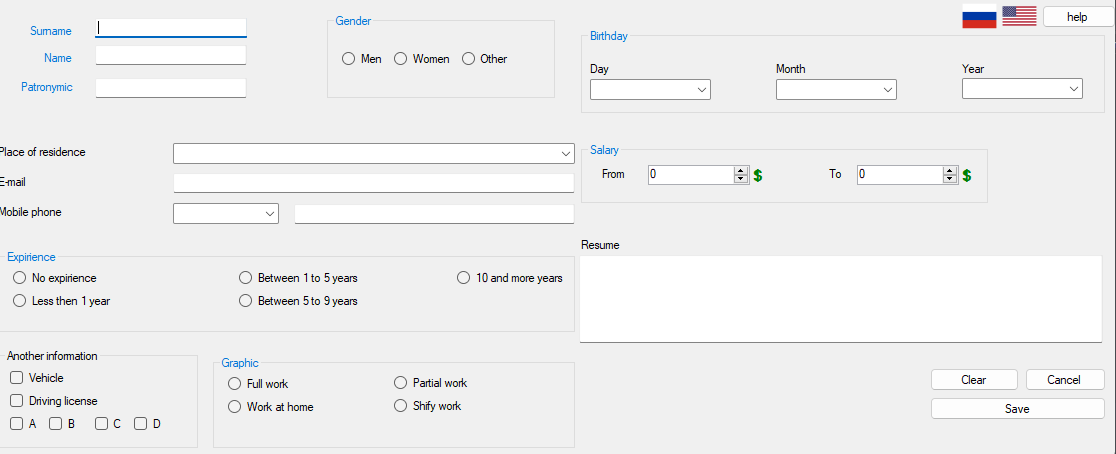


Рисунок 24.4 – Результат работы программы

Источник: собственная разработка