**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#.

Решение:

using System;

using System.Windows.Forms;

namespace MdiApplication

{

public partial class ParentForm : Form

{

private int openDocuments = 0;

public ParentForm()

{

InitializeComponent();

InitializeToolStrip();

// Установите текущую дату в статусе spData

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

}

private void InitializeToolStrip()

{

toolStrip1.ItemClicked += new ToolStripItemClickedEventHandler(toolStrip1\_ItemClicked);

toolStripButton1.Tag = "NewDoc";

toolStripButton1.ToolTipText = "Create new document";

toolStripButton2.Tag = "Cascade";

toolStripButton2.ToolTipText = "Windows cascade";

toolStripButton3.Tag = "Title";

toolStripButton3.ToolTipText = "Windows title";

}

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

{

if (e.ClickedItem?.Tag != null)

{

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

{

case "NewDoc":

ChildForm newChild = new ChildForm

{

MdiParent = this,

Text = "Document " + ++openDocuments

};

newChild.Show();

break;

case "Cascade":

this.LayoutMdi(MdiLayout.Cascade);

spWin.Text = "Windows is cascade";

break;

case "Title":

this.LayoutMdi(MdiLayout.TileHorizontal);

spWin.Text = "Windows is horizontal";

break;

}

}

}

}

}

Анализ результатов:



Рисунок 24.1 – Результат задания

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

Решение:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WinTimer1

{

public partial class UserControlTimer : UserControl

{

public UserControlTimer()

{

InitializeComponent();

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

}

public bool TimeEnabled

{

get { return timer1.Enabled; }

set { timer1.Enabled = value; }

}

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

{

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

}

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

{

// Установка начального времени в Label

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

// Запуск таймера для периодического обновления времени

timer1.Start();

}

}

}

Анализ результатов:

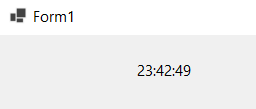


Рисунок 24.2 – Результат задания

Задание 3: Разработать собственный компонент и продемонстрировать его

функциональность в приложении.

using System;

using System.Drawing;

using System.Windows.Forms;

namespace CustomButtonControl

{

public class ColorChangingButton : Button

{

private readonly Random random = new Random();

protected override void OnMouseEnter(EventArgs e)

{

base.OnMouseEnter(e);

BackColor = GetRandomColor();

}

protected override void OnMouseLeave(EventArgs e)

{

base.OnMouseLeave(e);

BackColor = SystemColors.Control;

}

private Color GetRandomColor()

{

return Color.FromArgb(random.Next(256), random.Next(256), random.Next(256));

}

}

}

Анализ результатов:

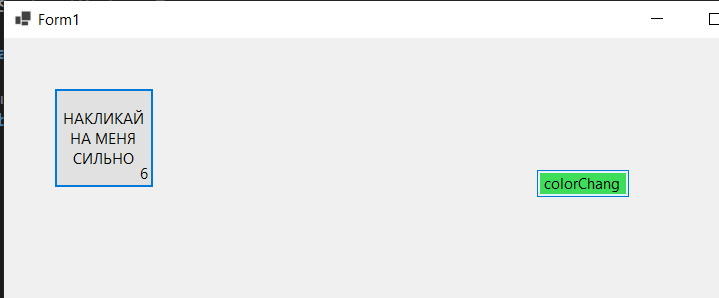


Рисунок 24.3 – Результат задания

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

Решение:

using System.Windows.Forms;

namespace Task4

{

public partial class Form4 : Form

{

private HelpProvider helpProvider;

private Button helpButton;

private ToolTip toolTip;

public Form4()

{

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

InitializeComponent();

MaximizeBox = false;

MinimizeBox = false;

HelpButton = true;

helpProvider = new HelpProvider();

helpProvider.HelpNamespace = @"D:\Practic\Tema24\WinFormsApp3\bin\Debug\net8.0-windows\helpfortask4.txt";

// Создайте кнопку для вызова справки

helpButton = new Button();

helpButton.Text = "Help";

helpButton.Click += new EventHandler(HelpButton\_Click);

this.Controls.Add(helpButton);

helpProvider.SetHelpString(Name, "Введите ваше имя");

helpProvider.SetHelpString(Surname, "Введите вашу фамилию");

helpProvider.SetHelpString(MiddleName, "Введите ваше отчество");

helpProvider.SetHelpString(Email, "Введите ваш адрес электронной почты");

helpProvider.SetHelpString(PhoneNumber, "Введите ваш номер телефона");

helpProvider.SetHelpString(LocationCity, "Выберите ваш город проживания");

helpProvider.SetHelpString(CallOperator, "Выберите вашего оператора связи");

helpProvider.SetHelpString(Male, "Выберите ваш пол");

helpProvider.SetHelpString(Female, "Выберите ваш пол");

helpProvider.SetHelpString(NotWorking, "Выберите ваш опыт работы");

helpProvider.SetHelpString(LessThanOneYear, "Выберите ваш опыт работы");

helpProvider.SetHelpString(FromOneToFiveYears, "Выберите ваш опыт работы");

helpProvider.SetHelpString(FromFiveToNineYears, "Выберите ваш опыт работы");

helpProvider.SetHelpString(MoreThanTenYears, "Выберите ваш опыт работы");

helpProvider.SetHelpString(DriverCategoryA, "Выберите категории водительских прав");

helpProvider.SetHelpString(DriverCategoryB, "Выберите категории водительских прав");

helpProvider.SetHelpString(DriverCategoryC, "Выберите категории водительских прав");

helpProvider.SetHelpString(DriverCategoryD, "Выберите категории водительских прав");

helpProvider.SetHelpString(FullTime, "Выберите ваш график работы");

helpProvider.SetHelpString(PartTime, "Выберите ваш график работы");

helpProvider.SetHelpString(WorkAtHome, "Выберите ваш график работы");

helpProvider.SetHelpString(TimeWork, "Выберите ваш график работы");

helpProvider.SetHelpString(MinSalary, "Введите минимальную зарплату");

helpProvider.SetHelpString(MaxSalary, "Введите максимальную зарплату");

toolTip = new ToolTip();

toolTip.SetToolTip(Name, "Введите ваше имя");

toolTip.SetToolTip(Surname, "Введите вашу фамилию");

toolTip.SetToolTip(MiddleName, "Введите ваше отчество");

toolTip.SetToolTip(Email, "Введите ваш адрес электронной почты");

toolTip.SetToolTip(PhoneNumber, "Введите ваш номер телефона");

toolTip.SetToolTip(LocationCity, "Выберите ваш город проживания");

toolTip.SetToolTip(CallOperator, "Выберите вашего оператора связи");

toolTip.SetToolTip(Male, "Выберите ваш пол");

toolTip.SetToolTip(Female, "Выберите ваш пол");

toolTip.SetToolTip(NotWorking, "Выберите ваш опыт работы");

toolTip.SetToolTip(LessThanOneYear, "Выберите ваш опыт работы");

toolTip.SetToolTip(FromOneToFiveYears, "Выберите ваш опыт работы");

toolTip.SetToolTip(FromFiveToNineYears, "Выберите ваш опыт работы");

toolTip.SetToolTip(MoreThanTenYears, "Выберите ваш опыт работы");

toolTip.SetToolTip(DriverCategoryA, "Выберите категории водительских прав");

toolTip.SetToolTip(DriverCategoryB, "Выберите категории водительских прав");

toolTip.SetToolTip(DriverCategoryC, "Выберите категории водительских прав");

toolTip.SetToolTip(DriverCategoryD, "Выберите категории водительских прав");

toolTip.SetToolTip(FullTime, "Выберите ваш график работы");

toolTip.SetToolTip(PartTime, "Выберите ваш график работы");

toolTip.SetToolTip(WorkAtHome, "Выберите ваш график работы");

toolTip.SetToolTip(TimeWork, "Выберите ваш график работы");

toolTip.SetToolTip(MinSalary, "Введите минимальную зарплату");

toolTip.SetToolTip(MaxSalary, "Введите максимальную зарплату");

}

// Обработчик события нажатия на кнопку справки

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

{

Help.ShowHelp(this, helpProvider.HelpNamespace);

}

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

{

TextBox[] textBoxes = { Name, Surname, MiddleName, Email, PhoneNumber };

ComboBox[] comboBoxes = { LocationCity, CallOperator };

RadioButton[] radioButtons = { Male, Female };

RadioButton[] workExpRadioButtons = { NotWorking, LessThanOneYear, FromOneToFiveYears, FromFiveToNineYears, MoreThanTenYears };

CheckBox[] categoryCheckboxes = { DriverCategoryA, DriverCategoryB, DriverCategoryC, DriverCategoryD };

RadioButton[] workGraphicRadioButtons = { FullTime, PartTime, WorkAtHome, TimeWork };

NumericUpDown[] numericUpDowns = { MinSalary, MaxSalary };

string GetSelectedRadioButtonText(RadioButton[] radioButtons)

{

foreach (RadioButton radioButton in radioButtons)

{

if (radioButton.Checked)

{

return radioButton.Text;

}

}

return "";

}

string GetSelectedCheckBoxText(CheckBox[] checkBoxes)

{

string result = "";

foreach (CheckBox checkBox in checkBoxes)

{

if (checkBox.Checked)

{

result += checkBox.Text;

}

}

return result;

}

string[] textBoxValues = textBoxes.Select(textBox => textBox.Text).ToArray();

string[] comboBoxValues = comboBoxes.Select(comboBox => comboBox.Text).ToArray();

string sex = GetSelectedRadioButtonText(radioButtons);

string dateOfBirth = DateBirth.Text;

string workExp = GetSelectedRadioButtonText(workExpRadioButtons);

if (workExp == "")

{

MessageBox.Show("Не выбран опыт работы");

return;

}

string driveCategory = GetSelectedCheckBoxText(categoryCheckboxes);

string workGraphic = GetSelectedRadioButtonText(workGraphicRadioButtons);

if (workGraphic == "")

{

MessageBox.Show("Выберите график работы");

return;

}

string salaryFrom = Convert.ToString(MinSalary.Value);

string salaryTo = Convert.ToString(MaxSalary.Value);

AboutMe.AppendText($"Фамилия: {textBoxValues[1]}\n");

AboutMe.AppendText($"Имя: {textBoxValues[0]}\n");

AboutMe.AppendText($"Отчество: {textBoxValues[2]}\n");

AboutMe.AppendText($"Пол: {sex}\n");

AboutMe.AppendText($"Дата рождения: {dateOfBirth}\n");

AboutMe.AppendText($"Город проживания: {comboBoxValues[0]}\n");

AboutMe.AppendText($"Электронная почта: {textBoxValues[3]}\n");

AboutMe.AppendText($"Телефон: {textBoxValues[4]}\n");

AboutMe.AppendText($"Оператор связи: {comboBoxValues[1]}\n");

AboutMe.AppendText($"Опыт работы: {workExp}\n");

AboutMe.AppendText($"Личный автомобиль: {(HaveAPersonalCar.Checked ? "Да" : "Нет")}\n");

AboutMe.AppendText($"Водительское удостоверение: {(DriverLicense.Checked ? "Да" : "Нет")}\n");

AboutMe.AppendText($"Категории водительских прав: {driveCategory}\n");

AboutMe.AppendText($"Зарплата от: {salaryFrom}\n");

AboutMe.AppendText($"Зарплата до: {salaryTo}\n");

AboutMe.AppendText($"График работы: {workGraphic}\n");

}

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

{

TextBox[] textBoxes = { Name, Surname, MiddleName, Email, PhoneNumber };

ComboBox[] comboBoxes = { LocationCity, CallOperator };

RadioButton[] radioButtons = { Male, Female };

RadioButton[] workExpRadioButtons = { NotWorking, LessThanOneYear, FromOneToFiveYears, FromFiveToNineYears, MoreThanTenYears };

CheckBox[] categoryCheckboxes = { DriverCategoryA, DriverCategoryB, DriverCategoryC, DriverCategoryD };

RadioButton[] workGraphicRadioButtons = { FullTime, PartTime, WorkAtHome, TimeWork };

NumericUpDown[] numericUpDowns = { MinSalary, MaxSalary };

foreach (TextBox textBox in textBoxes)

{

textBox.Text = "";

}

foreach (ComboBox comboBox in comboBoxes)

{

comboBox.SelectedIndex = -1;

}

foreach (RadioButton radioButton in radioButtons)

{

radioButton.Checked = false;

}

foreach (CheckBox checkBox in categoryCheckboxes)

{

checkBox.Checked = false;

}

foreach (NumericUpDown numericUpDown in numericUpDowns)

{

numericUpDown.Value = numericUpDown.Minimum;

}

AboutMe.Clear();

}

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

{

Application.Exit();

}

}

}

Анализ результатов:

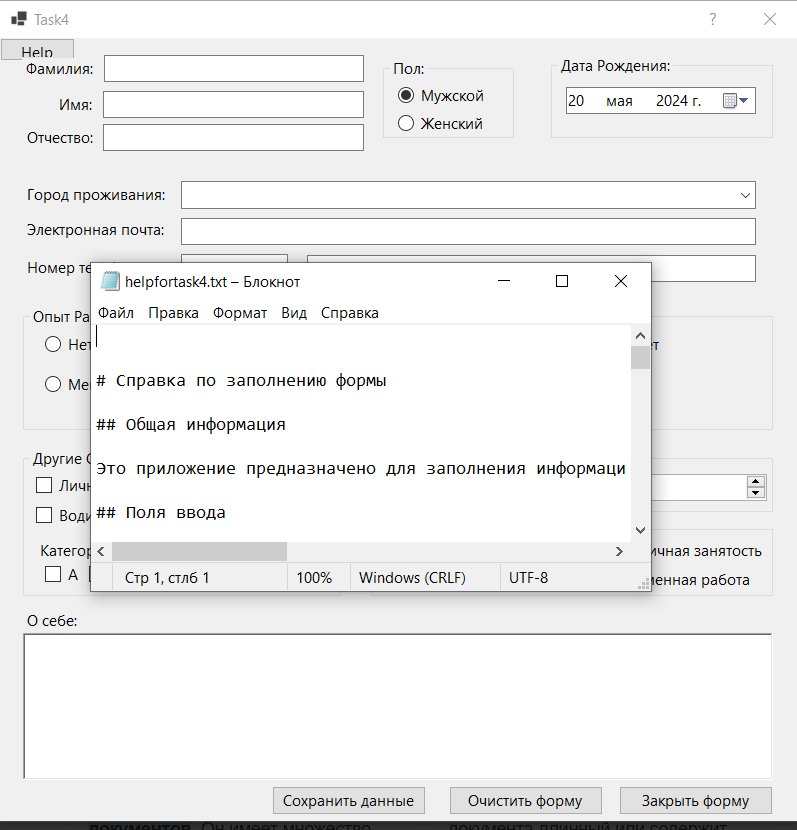


Рисунок 24.4 – Результат задания