1. **Меню и панели инструментов**

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

Листинг программы:

namespace Task\_1

{

public partial class Form1 : Form

{

private uint \_openDocuments = 1;

public Form1()

{

InitializeComponent();

}

private void CreateNewMdiChild()

{

ChildForm newChild = new ChildForm

{

MdiParent = this,

Text = $"{Text} #{\_openDocuments++}",

};

newChild.Show();

newChild.FormClosed += new FormClosedEventHandler((sender, e) =>

{

\_openDocuments--;

});

}

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

{

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

{

case "NewDoc":

CreateNewMdiChild();

break;

case "Cascade":

LayoutMdi(MdiLayout.Cascade);

break;

case "Title":

LayoutMdi(MdiLayout.TileHorizontal);

break;

}

}

}

}

Таблица 1.1 – Входные и выходные данные

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

Анализ результатов:

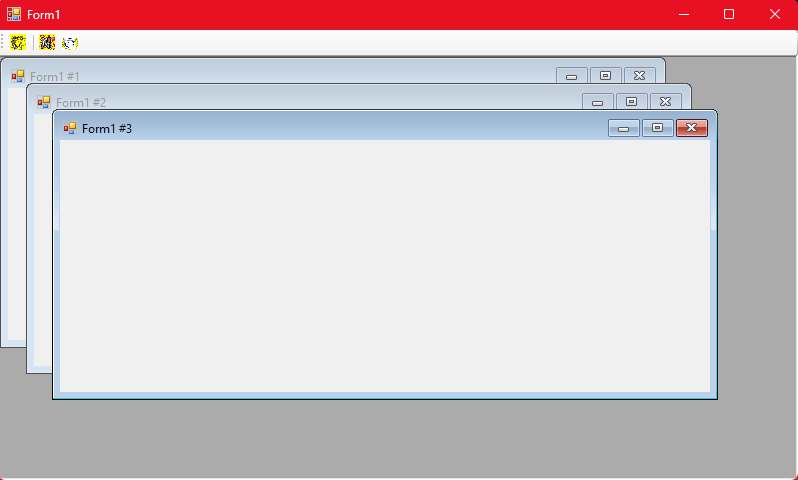


Рисунок 1.1 – Результат работы программы

Источник: собственная разработка

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

Листинг программы:

namespace Task\_2

{

public partial class Form1 : Form

{

private uint \_openDocuments = 1;

public Form1()

{

InitializeComponent();

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

}

private void CreateNewMdiChild()

{

ChildForm newChild = new ChildForm

{

MdiParent = this,

Text = $"{Text} #{\_openDocuments++}",

};

newChild.Show();

newChild.FormClosed += new FormClosedEventHandler((sender, e) =>

{

\_openDocuments--;

});

}

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

{

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

{

case "NewDoc":

CreateNewMdiChild();

break;

case "Cascade":

LayoutMdi(MdiLayout.Cascade);

spWin.Text = "Windows is cascade";

break;

case "Title":

LayoutMdi(MdiLayout.TileHorizontal);

spWin.Text = "Windows is horizontal";

break;

}

}

}

}

Таблица 2.1 – Входные и выходные данные

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

Анализ результатов:



Рисунок 2.1 – Результат работы программы

Источник: собственная разработка

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

Листинг программы:

namespace Task\_3

{

public partial class UserControlTimer : UserControl

{

public UserControlTimer()

{

InitializeComponent();

}

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

{

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

}

public bool TimeEnabled

{

get { return timer1.Enabled; }

set { timer1.Enabled = value; }

}

}

}

Таблица 3.1 – Входные и выходные данные

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

Анализ результатов:

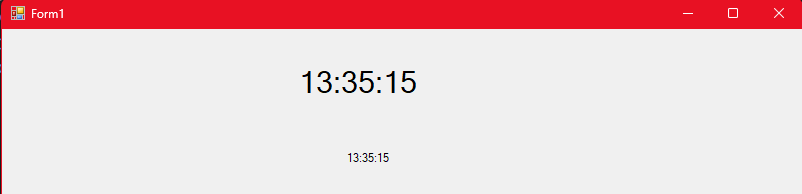


Рисунок 3.1 – Результат работы программы

Источник: собственная разработка

Задание 4. Разработать собственный компонент и продемонстрировать его функциональность в приложении.

Листинг программы:

namespace WFControlLibrarytable

{

public partial class UserControl1: UserControl

{

public UserControl1()

{

InitializeComponent();

}

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

{

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

}

public bool TimeEnabled

{

get { return timer1.Enabled; }

set { timer1.Enabled = value; }

}

}

}

Таблица 4.1 – Входные и выходные данные

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

Анализ результатов:

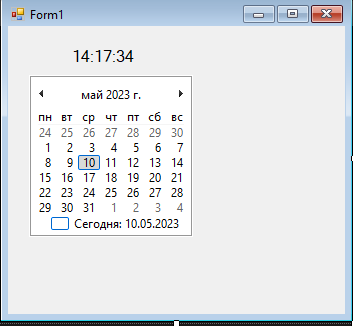


Рисунок 4.1 – Результат работы программы

Источник: собственная разработка

Задание 5. Выполнить лабораторную работу №8 из учебного пособия: Осипов Н.А. Разработка Windows приложений на C#.

Для выполнения работы используйте Задание №4 из темы: «Технология Windows Form. Работа с формами».

Листинг программы:

namespace Task\_4

{

public partial class Form1 : Form

{

public Form1()

{

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

InitializeComponent();

}

public string LastName { get => textBox1.Text; set => LastName = value; }

public string FirstName { get => textBox2.Text; set => FirstName = value; }

public string Patronymic { get => comboBox3.Text; set => Patronymic = value; }

public string Data { get => comboBox1.Text + " " + comboBox2.Text + " " + comboBox3.Text; set => Data = value; }

public string HomeAdress { get => comboBox4.Text; set => HomeAdress = value; }

public string EmailAdress { get => textBox4.Text; set => EmailAdress = value; }

public string PhoneType { get => comboBox5.Text; set => PhoneType = value; }

public string PhoneNumber { get => textBox5.Text; set => PhoneNumber = value; }

public string MinSalary { get => numericUpDown1.Text; set => MinSalary = value; }

public string MaxSalary { get => numericUpDown2.Text; set => MaxSalary = value; }

public string Summary { get => textBox6.Text; set => Summary = value; }

public short experience;

private void radioButton1\_CheckedChanged(object sender, EventArgs e) => experience = 1;

private void radioButton2\_CheckedChanged(object sender, EventArgs e) => experience = 2;

private void radioButton3\_CheckedChanged(object sender, EventArgs e) => experience = 3;

private void radioButton4\_CheckedChanged(object sender, EventArgs e) => experience = 4;

private void radioButton5\_CheckedChanged(object sender, EventArgs e) => experience = 5;

public short schedule;

private void radioButton8\_CheckedChanged(object sender, EventArgs e) => schedule = 1;

private void radioButton9\_CheckedChanged(object sender, EventArgs e) => schedule = 2;

private void radioButton10\_CheckedChanged(object sender, EventArgs e) => schedule = 3;

private void radioButton11\_CheckedChanged(object sender, EventArgs e) => schedule = 4;

public bool lastNameVerification = false;

public bool firstNameVerification = false;

public bool PatronymicVerification = false;

public bool EmailAdressVerification = false;

public bool PhoneNumberVerification = false;

public void Test()

{

if (textBox1.Text != string.Empty)

{

lastNameVerification = true;

for (int i = 0; i < textBox1.Text.Length; i++)

{

if (!Char.IsLetter(textBox1.Text[i]))

{

lastNameVerification = false;

break;

}

}

}

if (textBox2.Text != string.Empty)

{

firstNameVerification = true;

for (int i = 0; i < textBox2.Text.Length; i++)

{

if (!Char.IsLetter(textBox2.Text[i]))

{

firstNameVerification = false;

break;

}

}

}

if (textBox3.Text != string.Empty)

{

PatronymicVerification = true;

for (int i = 0; i < textBox3.Text.Length; i++)

{

if (!Char.IsLetter(textBox3.Text[i]))

{

PatronymicVerification = false;

break;

}

}

}

if (textBox4.Text != string.Empty)

{

EmailAdressVerification = false;

string pattern = @"^(?("")("".+?""@)|(([0-9a-zA-Z]((\.(?!\.))|[-!#\$%&'\\*\+/=\?\^`\{\}\|~\w])\*)(?<=[0-9a-zA-Z])@))" +

@"(?(\[)(\[(\d{1,3}\.){3}\d{1,3}\])|(([0-9a-zA-Z][-\w]\*[0-9a-zA-Z]\.)+[a-zA-Z]{2,6}))$";

if (Regex.IsMatch(textBox4.Text, pattern))

{

EmailAdressVerification = true;

}

}

if (textBox5.Text != string.Empty)

{

PhoneNumberVerification = true;

for (int i = 0; i < textBox5.Text.Length; i++)

{

if (textBox5.Text.Length != 9) PhoneNumberVerification = false;

else

{

if (!Char.IsNumber(textBox5.Text[i]))

{

PhoneNumberVerification = false;

break;

}

}

}

}

}

private void button2\_Click(object sender, EventArgs e)

{

Experience1.Checked = false;

Experience2.Checked = false;

Experience3.Checked = false;

Experience4.Checked = false;

Experience5.Checked = false;

Man.Checked = false;

Woman.Checked = false;

checkBox1.Checked = false;

checkBox2.Checked = false;

checkBox3.Checked = false;

checkBox4.Checked = false;

checkBox5.Checked = false;

checkBox6.Checked = false;

textBox1.Text = string.Empty;

textBox2.Text = string.Empty;

textBox3.Text = string.Empty;

textBox4.Text = string.Empty;

textBox5.Text = string.Empty;

textBox6.Text = string.Empty;

comboBox1.Text = string.Empty;

comboBox2.Text = string.Empty;

comboBox3.Text = string.Empty;

comboBox4.Text = string.Empty;

comboBox5.Text = string.Empty;

numericUpDown1.Text = "0";

numericUpDown2.Text = "0";

radioButton8.Checked = false;

radioButton9.Checked = false;

radioButton10.Checked = false;

radioButton11.Checked = false;

}

private void button3\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Вы точно хотите выйти?", "Выход", MessageBoxButtons.YesNo) == DialogResult.Yes)

Application.Exit();

}

private void button1\_Click(object sender, EventArgs e)

{

Test();

if (textBox1.Text != string.Empty && textBox2.Text != string.Empty && textBox3.Text != string.Empty && textBox5.Text != string.Empty &&

comboBox1.Text != string.Empty && comboBox2.Text != string.Empty && comboBox3.Text != string.Empty)

{

if (lastNameVerification && firstNameVerification && PatronymicVerification && EmailAdressVerification && PhoneNumberVerification)

{

File.AppendAllText("output.txt", "\r\n" + LastName + " " + FirstName + " " + Patronymic + "\r\n");

if (Man.Checked) File.AppendAllText("output.txt", "Пол - Мужской\r\n");

if (Woman.Checked) File.AppendAllText("output.txt", "Пол - Женский\r\n");

File.AppendAllText("output.txt", "Дата рождения - " + Data + "\r\n");

File.AppendAllText("output.txt", "Домашний адрес - " + HomeAdress + "\r\n");

File.AppendAllText("output.txt", "Email адрес - " + EmailAdress + "\r\n");

File.AppendAllText("output.txt", "Мобильный телефон - " + PhoneType + " " + "+375" + PhoneNumber + "\r\n");

string lineOne = "Стаж работы - ";

switch (experience)

{

case 1: File.AppendAllText("output.txt", "Прежде не работал" + "\r\n"); break;

case 2: File.AppendAllText("output.txt", lineOne + "Меньше 1 года" + "\r\n"); break;

case 3: File.AppendAllText("output.txt", lineOne + "От 1 года до 5 лет" + "\r\n"); break;

case 4: File.AppendAllText("output.txt", lineOne + "От 5 до 9 лет" + "\r\n"); break;

case 5: File.AppendAllText("output.txt", lineOne + "10 лет и больше" + "\r\n"); break;

}

string lineTwo = "Тип занятости - ";

switch (schedule)

{

case 1: File.AppendAllText("output.txt", lineTwo + "Полная занятость" + "\r\n"); break;

case 2: File.AppendAllText("output.txt", lineTwo + "Частичная занятость" + "\r\n"); break;

case 3: File.AppendAllText("output.txt", lineTwo + "Работа на дому" + "\r\n"); break;

case 4: File.AppendAllText("output.txt", lineTwo + "Посменная работа" + "\r\n"); break;

}

if (checkBox1.Checked) File.AppendAllText("output.txt", "Личный автомобиль - Есть\r\n");

else File.AppendAllText("output.txt", "Личного автомобиля - Нету\r\n");

if (checkBox2.Checked)

{

File.AppendAllText("output.txt", "Водительское удостоверение - Есть ");

if (checkBox3.Checked) File.AppendAllText("output.txt", " A");

if (checkBox4.Checked) File.AppendAllText("output.txt", " B");

if (checkBox5.Checked) File.AppendAllText("output.txt", " C");

if (checkBox6.Checked) File.AppendAllText("output.txt", " D");

File.AppendAllText("output.txt", "\r\n");

}

else File.AppendAllText("output.txt", "Водительское удостоверение - Нету\r\n");

if (MinSalary != "0" && MaxSalary != "0") File.AppendAllText("output.txt", "ЗП от:" + MinSalary + " до:" + MaxSalary + "\r\n");

if (textBox6.Text != String.Empty) File.AppendAllText("output.txt", "Краткое резюме:\r\n" + Summary + "\r\n");

File.AppendAllText("output.txt", "=================================\r\n");

MessageBox.Show("Сохранено!", "Сохранение!", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

string lineThree = "Введены неверно следущие поля:\r\n";

if (!lastNameVerification) { lineThree += "- Фамилия\r\n"; }

if (!firstNameVerification) { lineThree += "- Имя\r\n"; }

if (!PatronymicVerification) { lineThree += "- Отчество\r\n"; }

if (!EmailAdressVerification) { lineThree += "- Адресс э. почты\r\n"; }

if (!PhoneNumberVerification) { lineThree += "- Номер телефона\r\n"; }

MessageBox.Show(lineThree, "Неверно заполнены поля!", MessageBoxButtons.OK);

}

}

else

{

string lineFour = "Не заполнены следущие поля:\r\n";

if (textBox1.Text == string.Empty) { lineFour += "- Фамилия\r\n"; }

if (textBox2.Text == string.Empty) { lineFour += "- Имя\r\n"; }

if (textBox3.Text == string.Empty) { lineFour += "- Отчество\r\n"; }

if (comboBox1.Text == string.Empty) { lineFour += "- День рождения\r\n"; }

if (comboBox2.Text == string.Empty) { lineFour += "- Месяц рождения\r\n"; }

if (comboBox3.Text == string.Empty) { lineFour += "- Год рождения\r\n"; }

if (textBox5.Text == string.Empty) { lineFour += "- Адрес эл. почты\r\n"; }

MessageBox.Show(lineFour, "Не заполнены поля!", MessageBoxButtons.OK);

}

}

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

{

Help.ShowHelp(this, helpProvider1.HelpNamespace);

}

}

}

Таблица 5.1 – Входные и выходные данные

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

Анализ результатов:

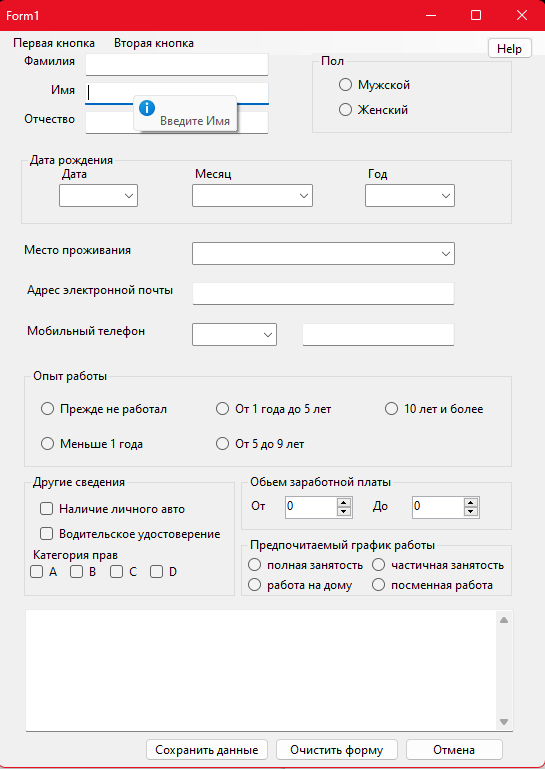


Рисунок 5.1 – Результат работы программы

Источник: собственная разработка