

WINDOWS FORMS

1. На форме разместите компонент DateTimePicker. По выбранному времени в строке состояния должно выводиться сообщение «до полудня» или «после полудня» + При клике на кнопку увеличивать дату на 10 дней в датапикере.

```
public Form1()
{
    InitializeComponent();
    dateTimePicker1.Format = DateTimePickerFormat.Custom;
    dateTimePicker1.CustomFormat = "dd.MM.yyyy HH:mm";
}

private void dateTimePicker1_ValueChanged(object sender, EventArgs e)
{
    DateTime SelectedTime = dateTimePicker1.Value;
    string message = SelectedTime.Hour < 12 ? "До полудня" : "После полудня";
    label1.Text = message;
}

private void button1_Click(object sender, EventArgs e)
{
    dateTimePicker1.Value = dateTimePicker1.Value.AddDays(10);
}
```

2. Создайте текстовое поле для ввода пользователем строки и кнопку по которой строка разбивается на две, равной длины. Результат выведите на форму + Вводишь в текстовое поле строку половина вниз половина остается, и так каждый раз

```
private void button1_Click(object sender, EventArgs e)
{
    string a = textBox1.Text;
    if (!string.IsNullOrEmpty(a))
    {
        int half = a.Length/2;
        string firstHalf = a.Substring(0, half);
        string secondHalf = a.Substring(half);

        textBox1.Text = firstHalf;
        label1.Text = secondHalf;
    }
}
```

3. Разработайте программу на C# WinForms. Разместите combobox со списком дисциплин (ОАП,ООП,БД). При выборе эл-та он должен удаляться из списка. Поменяйте курсор для окна приложения.

```
private void button2_Click(object sender, EventArgs e)
{
    string a = textBox1.Text;
    foreach (char c in a)
```

```

    {
        if (char.IsDigit(c))
        {
            this.Text += c;

        }
        else if (char.IsLetter(c))
        {
            label1.Text += c;
        }
    }
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    comboBox1.Items.RemoveAt(comboBox1.SelectedIndex);
}

```

4. Разработайте программу на C# WinForms, которая содержит два текстовых поля для задания пароля пользователем и его подтверждением. Выполните проверку на равенство паролей, его длину (от 6 до 12 символов) и обязательное наличие символов и цифр в пароле. В случае невыполнения условий сообщите об этом пользователю.

```

private void button3_Click(object sender, EventArgs e)
{
    string password = textBox2.Text;
    string confirmPassword = textBox3.Text;

    if (password.Length < 6 || password.Length > 12)
    {
        MessageBox.Show("neverno1");
    }
    else if (!password.Any(char.IsLetter) && !password.Any(char.IsDigit))
    {
        MessageBox.Show("neverno2");
    }
    else if (confirmPassword != password)
    {
        MessageBox.Show("neverno3");
    }
    //или Regex regex = new Regex(@"^(?=[a-zA-Z])(?=.*\d){6,12}$");
    // if (!regex.IsMatch(password))
}

```

5. Разработайте программу на C# WinForms. Форма содержит четыре радиокнопки, соответствую 4 функциям: sin, cos, tan, ctg, поле ввода для x, и поле вывода для ответа (+стереть все). Посчитать значения выбранной функции +. Поиск процента от введенного числа. 30 50 70 + Сделать перевод в различные системы счисления (2,8,16)

```

private void button1_Click(object sender, EventArgs e)
{

```

```

try
{
    double x = double.Parse(textBox1.Text);
    double result = 0;

    if (radioButton1.Checked)
    {
        result = Math.Sin(x);
    }
    else if (radioButton2.Checked)
    {
        result = Math.Cos(x);
    }
    else if (radioButton3.Checked)
    {
        result = Math.Tan(x);
    }
    else if (radioButton4.Checked)
    {
        result = 1 / Math.Tan(x);
    }
    textBox2.Text = result.ToString();
}
catch
{
    MessageBox.Show("Введите корректное значение для x.", "Ошибка", MessageBoxButtons.OK,
    MessageBoxIcon.Error);
}
}

private void button2_Click(object sender, EventArgs e)
{
    textBox1.Clear();
    textBox2.Clear();
    radioButton2.Checked = true;
    radioButton2.Checked = false;
    radioButton3.Checked = false;
    radioButton4.Checked = false;
}

```

```

private void button3_Click(object sender, EventArgs e)
{
    try
    {
        double chislo = double.Parse(textBox3.Text);
        double result2 = 0;
        if (radioButton8.Checked)
        {
            result2 = chislo * 0.3;
        }
        if (radioButton7.Checked)
        {
            result2 = chislo * 0.5;
        }
        if (radioButton6.Checked)
        {
            result2 = chislo * 0.7;
        }

        textBox5.Text = result2.ToString();
    }
    catch
    {
        MessageBox.Show("Введите корректное значение ", "Ошибка", MessageBoxButtons.OK,

```

```
MessageBoxIcon.Error);
    }
}
```

```
private void button4_Click(object sender, EventArgs e)
{
    try
    {
        int xyi = int.Parse(textBox6.Text);
        string xyilo = string.Empty;

        if (radioButton10.Checked)
        {
            xyilo = Convert.ToString(xyi, 2);
        }
        if (radioButton9.Checked)
        {
            xyilo = Convert.ToString(xyi, 8);
        }
        if (radioButton5.Checked)
        {
            xyilo = Convert.ToString(xyi, 16);
        }
        textBox4.Text = xyilo;
    }
    catch
    {
        MessageBox.Show("Введите корректное значение ", "Ошибка", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}
```

6. Найти расстояние между точками используя 1 класс (Приложение создает три точки на двумерной плоскости, подсчитывает общее количество созданных точек, выводит информацию о каждой точке, и вычисляет расстояния между ними.)

```
class Point
{
    int x;
    int y;
    int N;
    static int count=0;
    public Point(int x, int y )
    {
        this.x = x;
        this.y = y;
        count++;
        N = count;
    }
    public override string ToString()
    {
        return "Nomer=" + N + " X=" + x + " Y=" + y;
    }
    public static int Count()
    {
        return count;
    }
    public static double rasst(Point p1, Point p2)
```

```

    {
        return Math.Sqrt(Math.Pow(p2.x - p1.x, 2) + Math.Pow(p2.y - p1.y, 2));
    }
    public double rasst(Point p1)
    {
        return Math.Sqrt(Math.Pow(this.x - p1.x, 2) + Math.Pow(this.y - p1.y, 2));
    }
}
static void Main(string[] args)
{
    Point p1 = new Point(22, 11);
    Point p2 = new Point(3, 10);
    Point p3 = new Point(15, 1);
    Console.WriteLine(Point.Count());
    Console.WriteLine("{0}, {1}, {2}", p1.ToString(), p2.ToString(), p3.ToString());
    Console.WriteLine(Point.rasst(p1, p2));
    Console.WriteLine(p2.rasst(p3));
}

```

7. Различные операции LINQ (Language Integrated Query) для работы с коллекциями данных в C#.

- Найти элементы массива names, которые начинаются с буквы "A".
- Получить длину каждого элемента массива names.
- Разбить каждый элемент массива names на массив символов.
- Взять первые 2 элемента массива names.
- Взять элементы массива names, пока их длина меньше 5.
- Пропустить первые 2 элемента массива names.
- Пропустить элементы массива names, которые начинаются с буквы "A".
- Соединить первый элемент массива names с элементами после первых трех.
- Отсортировать элементы массива names по их длине.
- Отсортировать элементы массива names по их длине, а затем в обратном порядке по значению в пределах одной длины.
- Создать последовательность из 15 чисел, начиная с 34.
- Создать последовательность, содержащую число 10, повторяющееся 5 раз.
- Использование точечной нотации
- Создать массив целых чисел.
- Изменить элемент, в источнике данных

```

string[] names = { "Anna", "Alex", "Bob", "Alice", "John", "Amy", "Michael" };

//Найти элементы массива names, которые начинаются с буквы "A".
var a = names.Where(name => name.StartsWith("A")).ToList();
//Пропустить элементы массива names, которые начинаются с буквы "A".
var b = names.SkipWhile(name => name.StartsWith("A")).ToList();
//Получить длину каждого элемента массива names.
var c = names.Select(Name => Name.Length).ToList();
//Разбить каждый элемент массива names на массив символов.
var d = names.Select(name => name.ToCharArray()).ToList();
//Взять первые 2 элемента массива names.
var e = names.Take(2).ToList();
//Взять элементы массива names, пока их длина меньше 5.
var f = names.Select(name => name.Length < 5).ToList();

```

```
//Пропустить первые 2 элемента массива names.
var g = names.Skip(2).ToList();
//Соединить первый элемент массива names с элементами после первых трех.
var h = names.First() +string.Join("",names.Skip(3));
//Отсортировать элементы массива names по их длине.
var i = names.OrderBy(name => name.Length).ToList();
//Отсортировать элементы массива names по их длине, а затем в обратном порядке по значению в
пределах одной длины.
var k = names.OrderByDescending(name => name.Length < 5 ).ToList();
//Создать последовательность из 15 чисел, начиная с 34.
var l = Enumerable.Range(34,15).ToList();
//Создать последовательность, содержащую число 10, повторяющееся 5 раз.
var m = Enumerable.Repeat(10,5).ToList();
//Использование точечной нотации
var o = names
    .Where(name => name.StartsWith("A"))
    .OrderBy(name => name)
    .ToList();
//Создать массив целых чисел.
int[] numbers = Enumerable.Range(1, 10).ToArray();
//Изменить элемент, в источнике данных
var change = names.Select(name => name == "Alice"? "Papa": name).ToList();
```

8. Создайте приложение на C# с использованием Windows Forms, в котором будет реализован следующий функционал:

- Добавьте на форму два списка (ListBox), два текстовых поля (TextBox) и кнопку (Button).
- Напишите обработчик события для кнопки, который выполняет следующие действия:
 - Генерирует случайное количество чисел (от 10 до 20) в диапазоне от 100 до 1000.
 - Добавляет эти числа в listBox1, а их индексы в listBox2.
 - Определяет и отображает максимальное число и его индекс в textBox1 и textBox2.

```
private void button1_Click(object sender, EventArgs e)
{
    Random random = new Random();
    int count = random.Next(10, 21);

    List<int> numbers = new List<int>();

    listBox1.Items.Clear();
    listBox2.Items.Clear();
    textBox1.Clear();
    textBox2.Clear();

    for (int i = 0; i < count; i++)
    {
        int number = random.Next(100, 1001);
        numbers.Add(number);

        listBox1.Items.Add(number);

        listBox2.Items.Add(i);
    }

    int maxNumber = numbers.Max();
    int maxIndex = numbers.IndexOf(maxNumber);
```

```
textBox1.Text = maxNumber.ToString();
textBox2.Text = maxIndex.ToString();
}
```

9. Список со странами добавить контекстное меню, чтобы можно было их отсортировать и удалить. Тут сделан как клик правой кнопки мыши

```
public Form1()
{
    InitializeComponent();
    string[] countries = { "Италия", "Франция", "Германия", "Беларусь", "Канада", "Латвия" };
    listBox1.Items.AddRange(countries);

    ToolStripMenuItem Delete = new ToolStripMenuItem("Delete");
    ToolStripMenuItem Sort = new ToolStripMenuItem("Sort");

    contextMenuStrip1.Items.AddRange( new[] { Delete, Sort });

    listBox1.ContextMenuStrip = contextMenuStrip1;
    Delete.Click += DeleteMenu_click;
    Sort.Click += SortMenu_click ;
}

void SortMenu_click(object sender, EventArgs e)
{
    listBox1.Sorted = true;
}

void DeleteMenu_click(object sender, EventArgs e)
{
    listBox1.Items.Remove(listBox1.SelectedItem);
}
```

10. При клике на кнопку отображается картинка + Есть 4 бокса для картинки, надо сделать чтобы тыкаешь на картинку которая сейчас видна (остальные нет) и становилась видна другая.

```
//сначала какую картинку хотим изменить, а потом сама картинка

private void pictureBox1_Click(object sender, EventArgs e)
{
    pictureBox2.Image = pictureBox1.Image;
    pictureBox1.Image = null;
}

private void pictureBox2_Click(object sender, EventArgs e)
{
    pictureBox3.Image = pictureBox2.Image;
    pictureBox2.Image = null;
}

private void pictureBox3_Click(object sender, EventArgs e)
{
    pictureBox4.Image = pictureBox3.Image;
    pictureBox3.Image = null;
}

private void pictureBox4_Click(object sender, EventArgs e)
{
}
```

```
pictureBox1.Image = pictureBox4.Image;
pictureBox4.Image = null;
}
```

11. Разработайте программу на C# WinForms. В 2-х listview заполняются индексы и значения массива (случайным образом). По нажатию на кнопку выполняется поиск значения и индекса максимального элемента и результаты выводятся в textbox-ы.

```
private void button1_Click(object sender, EventArgs e)
{
    listView1.Items.Clear();
    listView2.Items.Clear();

    Random random = new Random();
    var maxInt = 0;
    var maxIndex = 0;

    for (int i = 0; i < 10; i++)
    {
        int randomInt = random.Next(0, 300);

        ListViewItem item1 = new ListViewItem(randomInt.ToString());
        ListViewItem item2 = new ListViewItem(i.ToString());

        listView1.Items.Add(item1);
        listView2.Items.Add(item2);

        if (randomInt > maxInt)
        {
            maxInt = randomInt;
            maxIndex = i;
        }
    }
    textBox1.Text = maxInt.ToString();
    textBox2.Text = maxIndex.ToString();
}
```

12. Надо сделать чтобы при клике на окно оно изменяло свой размер

```
private void Form1_MouseDown(object sender, MouseEventArgs e)
{
    this.Size = new Size(this.Width + 100, this.Height + 100);
}

private void Form1_MouseUp(object sender, MouseEventArgs e)
{
    this.Size = new Size(Width - 100, Height - 100);
}
```

13. Есть список чисел, его выбираешь и выводится факториал числа

```
private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    int selectedNumber = int.Parse(listBox1.SelectedItem.ToString());
    long factorial = CountFactorial(selectedNumber);
    textBox1.Text = factorial.ToString();
}
```



```

}

private long CountFactorial(int number)
{
    long factorial = 1;
    for (int i = 1; i <= number; i++)
    {
        factorial *= i;
    }
    return factorial;
}

```

14. При клике на разные кнопки получать разные значения: год дата и время(минуты) есть еще вариант с выводом в строку состояния, и еще есть где выводится и обновляется дата и время которое сейчас

```

private void button1_Click(object sender, EventArgs e)
{
    label1.Text = DateTime.Now.Day.ToString() + " День";
}

private void button2_Click(object sender, EventArgs e)
{
    label1.Text = DateTime.Now.ToString("mm")+ " Минуты";
}

private void button3_Click(object sender, EventArgs e)
{
    label1.Text = DateTime.Now.Year.ToString() + " Год";
}

private void timer1_Tick(object sender, EventArgs e)
{
    string CurrentDate = DateTime.Now.ToString();
    this.Text = CurrentDate;
}

```

15. Есть лист бокс, при выборе элемента выводится его значение, так же есть кнопка сброса

```

private void listBox2_SelectedIndexChanged(object sender, EventArgs e)
{
    if (listBox2.SelectedItem != null)
    {
        int selected = int.Parse(listBox2.SelectedItem.ToString());
        textBox2.Text = selected.ToString();
    }
}

private void button4_Click(object sender, EventArgs e)
{
    textBox2.Clear();
    listBox2.ClearSelected();
}

```

16. Рисование окружностей ?

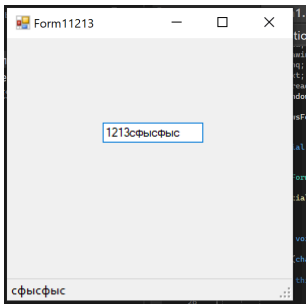
Это приложение на Windows Forms создает случайное количество окружностей, которые отображаются на форме. Вот что здесь происходит:

- В момент загрузки формы случайно определяется количество окружностей `col`
- В событии `Paint` формы для каждой окружности определяются координаты и они отображаются на форме.

- Метод `setcoord` вычисляет координаты следующей окружности, учитывая предыдущие координаты и размеры окружностей. Он также обрабатывает переполнение по горизонтали, чтобы новые окружности начинались с новой строки.

- Окружности рисуются с помощью объекта `Graphics` и метода `FillEllipse`, используя ранее вычисленные координаты.

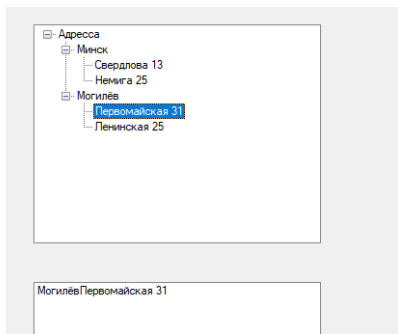
17. Есть текстовый ввод цифры они добавляются к названию формы, если буквы то ниже текстового



```
private void textBox1_TextChanged(object sender, EventArgs e)
{
    toolStripStatusLabel1.Text = " ";
    this.Text = "";

    string a = textBox1.Text;
    foreach (char c in a)
    {
        if (char.IsDigit(c))
        {
            toolStripStatusLabel1.Text += c;
        }
    }
    foreach (char c in a)
    {
        if (char.IsLetter(c))
        {
            this.Text += c;
        }
    }
}
```

18. Такая штука с адресами и добавлением их в список TreeView



```
InitializeComponent();
//1
TreeNode main1 = new TreeNode("1");
TreeNode main2 = new TreeNode("2");
//2
TreeNode node1 = new TreeNode("1_1");
TreeNode node2 = new TreeNode("1_2");

TreeNode node3 = new TreeNode("2_1");
TreeNode node4 = new TreeNode("2_2");

main1.Nodes.Add(node1);
main1.Nodes.Add(node2);

main2.Nodes.Add(node3);
main2.Nodes.Add(node4);

treeView1.Nodes.Add(main1);
treeView1.Nodes.Add(main2);

private void treeView1_AfterSelect(object sender, TreeViewEventArgs e)
{
    textBox2.Text = treeView1.SelectedNode.FullPath;
}
```

19. treeview Но для твоего компа, то есть должны отображаться все диски, папки и файлы

```
TreeNode rootNode = new TreeNode("My Computer");

DriveInfo[] drives = DriveInfo.GetDrives();

foreach (DriveInfo drive in drives)
{
    TreeNode driveNode = new TreeNode(drive.Name);
    AddDirectories(drive.RootDirectory, driveNode);
    rootNode.Nodes.Add(driveNode);
}

treeView2.Nodes.Add(rootNode);
private void AddDirectories(DirectoryInfo directory, TreeNode parentNode)
{

```

```

try
{
    DirectoryInfo[] directories = directory.GetDirectories();

    foreach (DirectoryInfo dir in directories)
    {
        TreeNode node = new TreeNode(dir.Name);
        AddDirectories(dir, node);
        parentNode.Nodes.Add(node);
    }
}
catch (Exception)
{}
}

```

20. Добавление в листбокс элементов и вывод их кол-ва

```

private void button1_Click(object sender, EventArgs e)
{
    listBox1.Items.Add("element");
    label2.Text = listBox1.Items.Count.ToString();
}

```

21. Создать MDI интерфейс, когда пишешь текст в одном окне, он меняется во всем остальных **не получается сделать (**

22. Папка называется drag and drops, не поняла в чем прикоп просто перетаскивать содержимое текстовых полей

```

private void TextBox_MouseMove(object sender, MouseEventArgs e)
{
    TextBox textBox = sender as TextBox;
    if (!string.IsNullOrEmpty(textBox.Text))
    {
        textBox.DoDragDrop(textBox.Text, DragDropEffects.Copy);
    }
}

private void TextBox_DragEnter(object sender, DragEventArgs e)
{
    if (e.Data.GetDataPresent(DataFormats.Text))
    {
        e.Effect = DragDropEffects.Copy;
    }
}

private void TextBox_DragDrop(object sender, DragEventArgs e)
{
    TextBox textBox = sender as TextBox;
    string draggedText = (string)e.Data.GetData(DataFormats.Text);
    textBox.Text = draggedText;
}

```

23. По двойному щелчку элемент из одного списка добавлять в другой

```
private void listBox1_DoubleClick(object sender, EventArgs e)
{
    string selected = listBox1.SelectedItem.ToString();
    listBox2.Items.Add(selected);
    listBox1.Items.Remove(selected);
}
```

24. При клике на кнопку добавлять введенное в текст бокс в листбокс

```
private void button1_Click(object sender, EventArgs e)
{
    string inputText = textBox4.Text;
    if (!string.IsNullOrEmpty(inputText))
    {
        listBox2.Items.Add((string)inputText);
    }
}
```

25. Перенести текст с формы на форму там опять MDI

26. ЭУ numericupdown. Сделать так чтобы были доступны значения только 1, 2, 3. И при выборе значения чтобы картинка менялась. Для каждого значения - свои картинки.

```
private void numericUpDown1_ValueChanged(object sender, EventArgs e)
{
    int selectedValue = (int)numericUpDown1.Value;
    switch (selectedValue)
    {
        case 1:
            pictureBox1.Image = new Bitmap("C:\\Users\\PC\\Pictures\\1.jpg");
            break;
        case 2:
            pictureBox1.Image = new
            Bitmap("C:\\Users\\PC\\Pictures\\dbae4a45fc0c86bedb4485e261e834b7.jpg");
            break;
        case 3:
            pictureBox1.Image = new Bitmap("C:\\Users\\PC\\Pictures\\Снимок экрана 2024-02-19 134812.png");
            break;
    }
}
```

27. Открытие и сохранение картинки как новой

```
private void button2_Click(object sender, EventArgs e)
{
    OpenFileDialog dialog = new OpenFileDialog();

    if (dialog.ShowDialog() == DialogResult.OK)
    {
        pictureBox2.ImageLocation = dialog.FileName;
    }
}
```

```

    }
}

private void button3_Click(object sender, EventArgs e)
{
    SaveFileDialog saveFileDialog = new SaveFileDialog();

    if (saveFileDialog.ShowDialog() == DialogResult.OK)
    {
        pictureBox2.Image.Save(saveFileDialog.FileName);
    }
}

```

28. добавить в комбобокс элементы. выбранный элемент вывести в строку состояния

```

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    string choosedEI = comboBox1.SelectedItem.ToString();
    toolStripStatusLabel1.Text = choosedEI;
}

```

29. Счетчик сколько раз кликнул по кнопке

```

private int counter = 0;
private void button4_Click(object sender, EventArgs e)
{
    counter++;
    label1.Text = counter.ToString();
}

```

30. Есть 2 трек бара найти разность между их значениями

```

private void trackBar1_Scroll(object sender, EventArgs e)
{
    int number1 = trackBar1.Value;
    int number2 = trackBar2.Value;
    CountRaznica(number1, number2);
}

private void trackBar2_Scroll(object sender, EventArgs e)
{
    int number1 = trackBar1.Value;
    int number2 = trackBar2.Value;
    CountRaznica(number1, number2);
}

private void CountRaznica(int number1, int number2)
{
    int raznica = (number1 - number2);
    label2.Text = raznica.ToString();
}

```

31. Добавить открытие диалоговых окон для шрифта и цвета текста

```

private void button1_Click(object sender, EventArgs e)

```

```

{
    ColorDialog colorDialog = new ColorDialog();
    if (colorDialog.ShowDialog() == DialogResult.OK)
    {
        richTextBox1.SelectionColor = colorDialog.Color;
    }
}

private void button2_Click(object sender, EventArgs e)
{
    FontDialog fontDialog = new FontDialog();
    if (fontDialog.ShowDialog() == DialogResult.OK)
    {
        richTextBox1.SelectionFont = fontDialog.Font;
    }
}

```

32. Есть чекбокс надо чтобы про таймеру менялись выбранные

```

int currentCheckBox = 1;
private void timer1_Tick(object sender, EventArgs e)
{
    switch (currentCheckBox)
    {
        case 1:
            checkBox1.Checked = true;
            checkBox2.Checked = false;
            checkBox3.Checked = false;
            break;
        case 2:
            checkBox1.Checked = false;
            checkBox2.Checked = true;
            checkBox3.Checked = false;
            break;
        case 3:
            checkBox1.Checked = false;
            checkBox2.Checked = false;
            checkBox3.Checked = true;
            break;
    }

    currentCheckBox++;
    if (currentCheckBox > 3)
    {
        currentCheckBox = 1;
    }
}

```

Файлы

1. На WinForms сделать сохранение данных в формат xml.

```

private List<Data> dataList = new List<Data>();
public class Data

```

```

{
    public string TextBox1Text { get; set; }
    public string TextBox2Text { get; set; }
    public string TextBox3Text { get; set; }
}
private void addButton_Click(object sender, EventArgs e)
{
    Data data = new Data
    {
        TextBox1Text = textBox1.Text,
        TextBox2Text = textBox2.Text,
        TextBox3Text = textBox3.Text
    };

    dataList.Add(data);
}

private void saveButton_Click(object sender, EventArgs e)
{
    SaveFileDialog saveFileDialog = new SaveFileDialog();

    if (saveFileDialog.ShowDialog() == DialogResult.OK)
    {
        DataManager.SaveData(dataList, saveFileDialog.FileName);
    }
}

private void loadButton_Click(object sender, EventArgs e)
{
    OpenFileDialog openFileDialog = new OpenFileDialog();

    if (openFileDialog.ShowDialog() == DialogResult.OK)
    {
        dataList = DataManager.LoadData(openFileDialog.FileName);

        string loadedData = "";
        foreach (Data data in dataList)
        {
            loadedData += $"{data.TextBox1Text}, {data.TextBox2Text},{data.TextBox3Text}\n";
        }
        MessageBox.Show(loadedData, "Загруженные данные");
    }
}

public static class DataManager
{
    public static void SaveData(List<Data> dataList, string filePath)
    {
        XmlSerializer serializer = new XmlSerializer(typeof(List<Data>));
        using (StreamWriter writer = new StreamWriter(filePath))
        {
            serializer.Serialize(writer, dataList);
        }
    }

    public static List<Data> LoadData(string filePath)
    {
        XmlSerializer serializer = new XmlSerializer(typeof(List<Data>));
        using (StreamReader reader = new StreamReader(filePath))
        {
            return (List<Data>)serializer.Deserialize(reader);
        }
    }
}

```


2. Запись в файл и из файла (в примере текстовый) но есть еще вариант с xml

```
private List<Data> dataList = new List<Data>();
public class Data
{
    public string TextBox1Text { get; set; }
    public string TextBox2Text { get; set; }
    public string TextBox3Text { get; set; }
}
private void addButton_Click(object sender, EventArgs e)
{
    Data data = new Data
    {
        TextBox1Text = textBox1.Text,
        TextBox2Text = textBox2.Text,
        TextBox3Text = textBox3.Text
    };

    dataList.Add(data);
}

private void saveButton_Click(object sender, EventArgs e)
{
    SaveFileDialog saveFileDialog = new SaveFileDialog();

    if (saveFileDialog.ShowDialog() == DialogResult.OK)
    {
        DataManager.SaveData(dataList, saveFileDialog.FileName);
    }
}

private void loadButton_Click(object sender, EventArgs e)
{
    OpenFileDialog openFileDialog = new OpenFileDialog();

    if (openFileDialog.ShowDialog() == DialogResult.OK)
    {
        dataList = DataManager.LoadData(openFileDialog.FileName);

        string loadedData = "";
        foreach (Data data in dataList)
        {
            loadedData += $"{data.TextBox1Text}, {data.TextBox2Text}, {data.TextBox3Text}\n";
        }
        MessageBox.Show(loadedData, "Загруженные данные");
    }
}

public static class DataManager
{
    public static void SaveData(List<Data> dataList, string filePath)
    {
        XmlSerializer serializer = new XmlSerializer(typeof(List<Data>));
        using (StreamWriter writer = new StreamWriter(filePath))
        {
            serializer.Serialize(writer, dataList);
        }
    }

    public static List<Data> LoadData(string filePath)
    {
        XmlSerializer serializer = new XmlSerializer(typeof(List<Data>));
        using (StreamReader reader = new StreamReader(filePath))
        {
            return (List<Data>)serializer.Deserialize(reader);
        }
    }
}
```

```
}
```

WPF

Простые

1. Разработайте программу на C# WPF. На форме разместите ползунок и текст к нему привязанный, Размер текста изменяется перемещением ползунка. Какой-то текст

```
<Slider x:Name="SliderValue" HorizontalAlignment="Left" Margin="52,78,0,0"
VerticalAlignment="Top" Width="251" />

<TextBlock HorizontalAlignment="Left" Margin="52,40,0,0" TextWrapping="Wrap" Text="TextBlock"
VerticalAlignment="Top" FontSize="{Binding ElementName=SliderValue,Path=Value}"/>
```

2. Разработайте программу на C# WPF. Создайте стиль для CheckBox с установкой параметров шрифта. Используя наследование стилей создайте другой стиль с определением цвета. Примените оба стиля.

```
<Window.Resources>
  <Style x:Key="BaseStyle" TargetType="CheckBox">
    <Setter Property="FontFamily" Value="Arial"/>
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="FontWeight" Value="Bold"></Setter>
  </Style>
  <Style x:Key="SecondStyle" BasedOn="{StaticResource ResourceKey=BaseStyle}"
TargetType="CheckBox">
    <Setter Property="Background" Value="Blue"/>
    <Setter Property="Foreground" Value="DarkBlue"/>
  </Style>
</Window.Resources>

<CheckBox Content="CheckBox" Style="{StaticResource BaseStyle}"
<CheckBox Content="CheckBox" Style="{StaticResource SecondStyle}"
```

3. Создать два стиля для кнопки. Организовать циклическую смену стилей по нажатию на кнопку. (При нажатии клавиши увеличивается счетчик, по клику на кнопку переключаются её стили, а по клику на элемент меню отображается счетчик нажатий, меняя цвет текста в зависимости от значения.)

```
<Style x:Key="ButtonStyle1" TargetType="Button">
  <Setter Property="Background" Value="LightBlue"/>
  <Setter Property="Foreground" Value="Black"/>
  <Setter Property="FontSize" Value="16"/>
  <Setter Property="Padding" Value="10"/>
</Style>

<Style x:Key="ButtonStyle2" TargetType="Button">
  <Setter Property="Background" Value="LightGreen"/>
  <Setter Property="Foreground" Value="Black"/>
```

```

<Setter Property="FontSize" Value="16"/>
<Setter Property="Padding" Value="10"/>

<Button x:Name="NiceButton" Content="Button" HorizontalAlignment="Left" Margin="77,319,0,0"
VerticalAlignment="Top" Height="61" Width="145" Click="NiceButton_Click"/>
<Menu Margin="0,0,0,399">
    <MenuItem Header="Нажми на меня" Click="MenuItem_Click"/>
</Menu>
<Label x:Name="CounterLabel" Content="Counter" HorizontalAlignment="Left" Margin="265,319,0,0"
VerticalAlignment="Top"/>

private int clickCount = 0;
private bool isStyle1 = true;
public MainWindow()
{
    InitializeComponent();
}

private void NiceButton_Click(object sender, RoutedEventArgs e)
{
    clickCount++;
    if (isStyle1)
    {
        NiceButton.Style = FindResource("ButtonStyle2") as Style;
    }
    else
    {
        NiceButton.Style = FindResource("ButtonStyle1") as Style;
    }
    isStyle1 = !isStyle1;
}

private void MenuItem_Click(object sender, RoutedEventArgs e)
{
    CounterLabel.Content = $"Количество нажатий: {clickCount}";
    if (clickCount % 2 == 0)
    {
        CounterLabel.Foreground = Brushes.Red;
    }
    else
    {
        CounterLabel.Foreground = Brushes.Green;
    }
}

```

4. Разработайте программу на C# WPF. Создайте триггер, который при наличии в двух текстовых полях отличного от нуля значения, выводит результат деления первого на второе.

хуйня ебаная гори в аду

нужно посмотреть как еще триггеры можно использовать

уже не нужно

Базы данных

1. Разработайте программу на C# WPF. На окне разместите кнопку «Загрузить новый курс» по нажатию на которую данные из таблицы (содержит фамилию и номер курса) БД SQL Server первого курса отображаются в форме. Использовать классы Connection и Command.

```
private void LoadData()
{
    try
    {
        using (SqlConnection connection = new
SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionDB"].ConnectionString))
        {
            connection.Open();
            string query = "SELECT * FROM STUDENTS WHERE Kurs = 1";

            using (SqlCommand command = new SqlCommand(query, connection))
            {
                SqlDataReader dataReader = command.ExecuteReader();
                DataTable dataTable = new DataTable();
                dataTable.Load(dataReader);

                dataGrid.ItemsSource = dataTable.DefaultView;
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Ошибка загрузки данных: {ex.Message}");
    }
}
```

2. Разработайте программу на C# WPF. Разместите кнопку «Загрузить» по нажатию на которую данные из массива объектов (содержит номер карты, имя и фамилию и срок действия) загружаются в таблицу card базы данных SQL.

```
private void LoadData()
{
    var cards = new[]
    {
        new { CardNumber = "1234", FirstName = "John", LastName = "Doe", ExpiryDate = "12/25" },
        new { CardNumber = "2345", FirstName = "Jane", LastName = "Smith", ExpiryDate = "11/24" },
        new { CardNumber = "3456", FirstName = "Alice", LastName = "Johnson", ExpiryDate = "10/23" }
    };

    try
    {
        using (SqlConnection connection = new
SqlConnection(ConfigurationManager.ConnectionStrings["Connection"].ConnectionString))
        {
            connection.Open();

            foreach (var card in cards)
            {
                string query = "INSERT INTO CARDS (Card_number, FirstName, SecondName, ExpiryDate)
VALUES (@CardNumber, @FirstName, @LastName, @ExpiryDate)";
            }
        }
    }
}
```

```

        using (SqlCommand command = new SqlCommand(query, connection))
        {
            command.Parameters.AddWithValue("@CardNumber", card.CardNumber);
            command.Parameters.AddWithValue("@FirstName", card.FirstName);
            command.Parameters.AddWithValue("@LastName", card.LastName);
            command.Parameters.AddWithValue("@ExpiryDate", card.ExpiryDate);

            command.ExecuteNonQuery();
        }
    }

    MessageBox.Show("Данные успешно загружены.");
}
}
catch (Exception ex)
{
    MessageBox.Show($"Ошибка загрузки данных: {ex.Message}");
}
}
}

```

3. Разработайте программу на C#. Разместите поле ввода для цены и кнопку «Посчитать» по нажатию на которую считается общая сумма вкладов пользователей из таблицы users базы данных SQL (содержат Id, name, sum). Используйте ADO.net **.(дефолт короче)**

4. Разработайте программу на C# WPF. В окне разместите кнопку «Найти» и текстовое поле с подписью «Год» по нажатию на которую данные из таблицы Goods базы данных SQL (содержит название товара, id и год выпуска), которые соответствуют введенному пользователем году должны появиться в окне в виде таблицы. Использовать классы Connection и Command.

5. Разработайте программу на C#. Разместите кнопки «Загрузить имя» и «Загрузить возраст» по нажатию на которые соответствующие данные из таблицы (содержит имя, курс и возраст) БД SQL Server отображаются в окне. Используйте Entity Framework.

6. Разработайте программу на C#. Разместите поле ввода и разработать кнопку для поиска по определенному критерию. Это с бд!

7. entity code first создать связанные таблицы, наложить ограничения, добавить несколько записей через элементы управления

8. Сделать удаление добавление в бд

Файлы

1. Разработайте программу на C# WPF. В окне разместите таблицу связанную с XML/Json (на выбор) файлом. Используйте архитектуру MVVM.

2. Пусть дан файл следующего содержимого. В окне разместите текстовое поле и кнопку. При задании определенного числа (год рождения) в окно выводится информация о студентах заданного года. Предусмотреть обработку ошибок ввода.

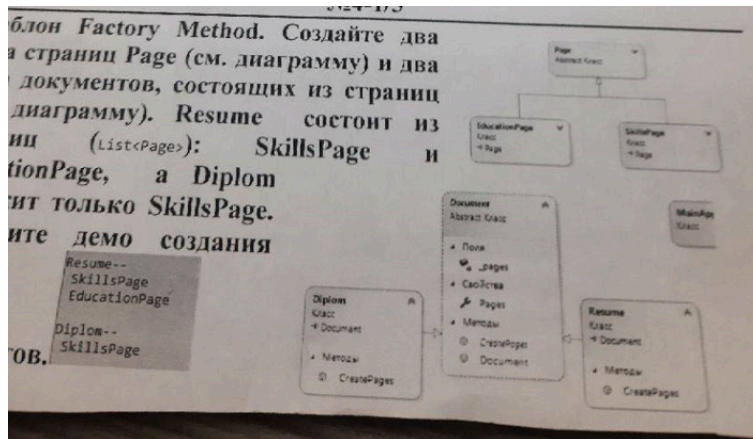
"department":

```
""студент":  
[  
  "_name": "Ольга",  
  "_group": "1",  
  "_age": "18"  
  "_name": "Никита",  
  "_group": "2"  
  {  
    age": "19  
    "_name": "Павел",  
    "_group":  
    "_age": "20"  
    "_name": "  
  "FIT"
```

3. Сделать тип сериализацию xml или json и вывести в таблицу. Реализовать архитектуру MVVM

Паттерны

1. Шаблон Factory Method. Создайте два типа страниц Page (см. диаграмму) и два типа документов, состоящих из страниц (см. диаграмму). Resume состоит из страницы (ListPage): SkillsPage и EducationPage, а Diplom содержит только SkillsPage. Напишите демо создания документов.



2. Создайте потокобезопасный класс BstuNet на Singleton основе паттерна Singleton. Класс содержит список Instance: Singleton серверов. Сервер имеет имя и IP адрес. В Singleton()

#Instance():

Singleton

конструкторе создается, например, три сервера.

Продемонстрируйте получение доступа к определенному серверу несколько раз и докажите, что это один и тот же объект.

3. Реализовать паттерн заместитель (Proxy). Определить методы для сложения и вычитания чисел. Реализация паттерна Proxy по диаграмме классов (в листе была диаграмма, по ней и составлено).

Как работать с бд

1. Создать файл бд

2. Создать подключение в App.config

```
<connectionStrings>
  <add name="ConnectionToDB" название подключения />
</connectionStrings>
```

```
connectionString="Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=D:\Уник\ООП\Лаб\Lab8_1\Librar
y.mdf;Integrated Security=True" /> - эта строка берется из свойств файла бд
</connectionStrings>
```

3. Создаем необходимые таблицы

Добавить таблицу, в проектировании создаем и потом обновить

4. Добавить строки в таблицу сначала можно через запрос

```
INSERT INTO [dbo].[Table] ([SecondName], [Kurs])
VALUES (N'Иванов', 1);

INSERT INTO [dbo].[Table] ([SecondName], [Kurs])
VALUES (N'Петров', 2);

INSERT INTO [dbo].[Table] ([SecondName], [Kurs])
VALUES (N'Сидоров', 3);

select * from [dbo].[Table]
```

помоимо можно как-то проще прописывать

5. Добавить датагрид для отображения и кнопки

```
<DataGrid Name="dataGrid" AutoGenerateColumns="True"...
```

6. Подключение к бд в самом приложении

```
private SqlConnection sqlConnection = null;
private void Window_Loaded(object sender, RoutedEventArgs e)
{
    sqlConnection = new
SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionToDB"].ConnectionS
tring);
    sqlConnection.Open();
    if (sqlConnection.State == ConnectionState.Open)
    {
        MessageBox.Show("Подключение с бд установлено.");
    }
}
```

7. Метод для загрузки данных в датагрид + ЗАПРОС

```
private void LoadData()
{
    try
```



```
{
    using (SqlConnection connection = new
SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionDB"].ConnectionStri
ng))
    {
        connection.Open();
        string query = "SELECT * FROM STUDENTS WHERE Kurs = 1";

        using (SqlCommand command = new SqlCommand(query, connection))
        {
            SqlDataReader dataReader = command.ExecuteReader();
            DataTable dataTable = new DataTable();
            dataTable.Load(dataReader);

            dataGrid.ItemsSource = dataTable.DefaultView;
        }
    }
}
catch (Exception ex)
{
    MessageBox.Show($"Ошибка загрузки данных: {ex.Message}");
}
}
```