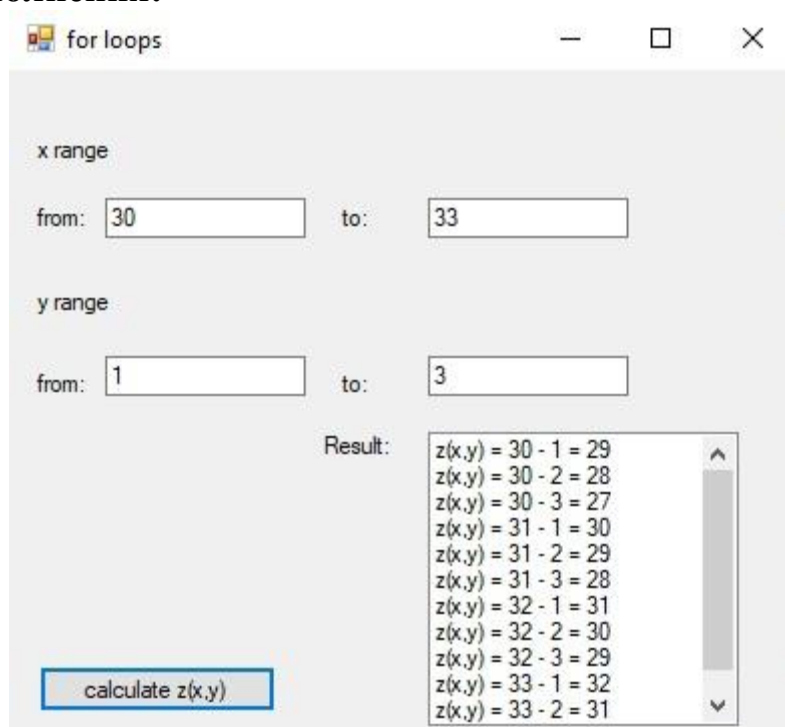


## Задание №1

**Выполнить:** Создайте приложение *Windows Forms Application* для расчета значения функции  $z(x,y) = x - y$ . Диапазон значений для  $x$  и  $y$  вводятся в текстовые поля.

[Solution and Project name: Lab5\_1, form name Lab5\_1.cs]

### Пример выполнения:



элемент управления	свойство	свойство
	name	text
form	Lab5	цикл for
button	btnCalc	Вычислить z(x,y)
textbox 1	txtX1	
textbox 2	txtX2	
textbox 3	txtY1	
textbox 4	txtY2	
ListBox	lstResult	
Labels		

1. *Внимание!* Задавайте свойство *name* для элементов управления так, как указано в таблице.
2. Создайте проект (Файл → Новый → Проект → *Windows Forms Application*), и задайте имя проекта — *Lab5*; свойство *name* формы должно быть *Lab5* (Окно Свойства → (*Name*)).
3. Расположите элементы на форме так, как это показано на рисунке.

4. Запрограммируйте событие *Click* кнопки. Для этого дважды щелкните по ней на дизайне формы. Откроется код для события. Интервалы для переменных  $x$  и  $y$  будут вводиться в текстовые поля (text box). Добавим переменные, которые проассоциируем с данными текстовыми полями. Не забудем перевести вводимые значения в целочисленный тип — *int*:

```
private void button1_Click(object sender, EventArgs e)
{
    // здесь ваш код:
    int fromX = int.Parse(txtX1.Text);
    int toX = int.Parse(txtX2.Text);
    int fromY = int.Parse(txtY1.Text);
    int toY = int.Parse(txtY2.Text);
    //...
}
```

5. Сейчас можно создать вложенный цикл для расчета функции  $z(x,y) = x - y$ . Внешний цикл будет перебирать значения переменной  $x$ , а внутренний —  $y$ . Продолжите код обработки события *click* кнопки:

```
//...
for (int x = fromX; x <= toX; x++)
{
    for (int y = fromY; y <= toY; y++)
    {
        // ...
    }
}
```

6. Результаты вычислений будут добавляться в элемент *Список* — *list box*; каждый полученный результат — на отдельной строке списка. Будем использовать метод *Add* коллекции *Items* элемента *Список*:

```
//...
for (int y = fromY; y <= toY; y++)
{
    // здесь ваш код:
    lstResult.Items.Add($"z(x,y) = {x} - {y} = {x - y}");
    // конец вашего кода
}
```

8. Запустите приложение, введите значения в текстовые поля и щелкните по кнопке. Давайте представим, что вводимые значения были от большего числа к меньшему. В таком случае компилятор выдаст ошибку.
9. Добавьте следующий код перед циклом, чтобы проверить вводимый диапазон:

```

if (fromX > toX)
{
    MessageBox.Show("интервал должен быть от меньшего к большему");
    txtX1.Text = "";
    txtX2.Text = "";
}

```

Метод Show класса MessageBox используется для вывода диалоговых окон.

10. Мы проверили интервал для x. То же самое сделайте для y самостоятельно.
11. Запустите приложение.

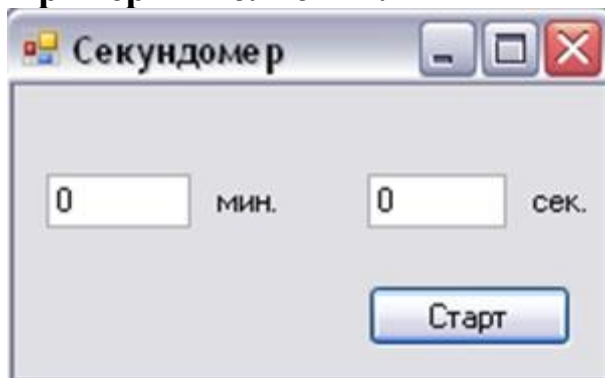
## Задание №2

Разработайте приложение «таймер» или «*Секундомер*», которое при нажатии на кнопку начинает или приостанавливает отсчет времени в секундах.

**Целью данного практического занятия** является ознакомление с объектом timer, рассмотрение его свойств и метода tick.

[Solution and Project name: Lab5\_2, form name Lab5\_2.cs]

### Пример выполнения:



элемент управления	свойство name
форма	
текстовое поле (textbox)	txtSeconds
текстовое поле (textbox)	txtMinutes
таймер (timer)	tmrSecundomer
кнопка (button)	btnStart

### Выполнение:

1. Создайте на новой форме два текстовых окна: для секунд (*txtSeconds*) и минут (*txtMinutes*).
2. Создайте элемент управления *Timer* (*tmrSecundomer*).

Этот элемент не отображается на форме, а потому размещается в нижней части экрана. *Timer* необходим, чтобы выполнять какие-либо действия через определенные промежутки времени, поэтому он невидим при работе приложения. Промежуток времени устанавливается свойством таймера *Interval*, которое измеряется в миллисекундах. Чтобы отсчитывать секунды необходимо установить значение **1000** для этого свойства.

3. Установите значение *1000* для свойства *Interval* элемента *Timer*.
4. Запрограммируйте событие *Tick* для таймера так, чтобы значение текстового окна секунд увеличивалось на единицу. Для этого дважды щелкните *Timer*:

```
private void TmrSecundomer_Tick(object sender, EventArgs e)
{
    // ваш код начинается здесь
    int tmp = Int32.Parse(txtSeconds.Text);
    tmp += 1;
    txtSeconds.Text = tmp.ToString();

    //...
}
```

Событие *Tick* происходит по истечении времени в миллисекундах, указанного в свойстве *Interval*.

5. Запрограммируйте кнопку (*btnStart*), которая будет запускать или останавливать Ваш секундомер. Для запуска и остановки отсчета времени используйте тот факт, что изначально таймер заблокирован, т.е. его свойство *Enabled* равно *false*. Чтобы запустить таймер необходимо установить это свойство в значение *true* и наоборот.
6. Запустите приложение. Введите в текстовое поле 0 или другое число. Щелкните по кнопке *Start* и убедитесь, что значение окна секунд стало увеличиваться на единицу.
7. Измените приложение так, чтобы время изменялось и в секундах и в минутах.
8. Добавьте кнопку *Сброс*, при нажатии на которую время на секундомере обнулялось бы.
9. Добавьте текстовое поле и найдите, как вывести в с# текущее время.