2019 年 6 月 27 日（木）実施

クラスの基本とメソッド

第 2 回の教材では，Form1.cs を例にとってクラスの構成要素について解説した。Visual Studio で C#言語によるフォームアプリケーションを作成する際，プロジェクトを新規作成すると自動的に生成される Program.cs は次の様になっている。

【Program.cs】

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Eleventh
{
    static class Program
    {
        /// <summary>
        /// アプリケーションのメイン エントリ ポイントです。
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```

プログラムが実行される際には，Program クラスの Main メソッドが最初に実行される。説明は省くが，Main メソッドの最終行の機能は，現在のスレッドで標準のアプリケーションメッセージループの実行を開始し，指定したフォームを表示するものである。ここで，new 演算子によって，Form1 クラスの**インスタンス**（実体）が生成されるが，new 演算子は新しいインスタンスをメモリ上に割り当て，**コンストラクタ**を呼び出してインスタンスを初期化し，インスタンスへの参照を返す。

Form1.cs には，フォームを定義する Form1 クラスが必須であるが，それ以外のクラスを作成して含めることが出来る。クラスの構成は一般に次の様になる。

```
namespace プロジェクト名
{
    修飾子 class クラス名
    {
        フィールド
        修飾子 コンストラクタ名( 引数リスト )
        {
```

```
            初期化処理
        }
        修飾子 データ型 メソッド名( 引数リスト )
        {
            処理
        }
    }
}
```

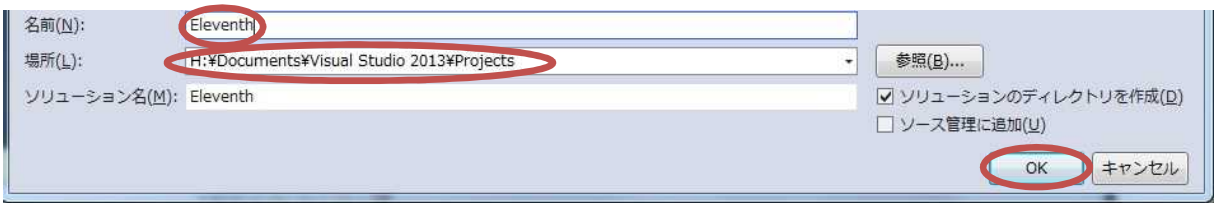
本日の課題

Form1.cs にはイベントハンドラ以外のメソッドを作成し，Form2.cs には Form2 クラス以外のクラスを作成することによって，同一クラス内及び別クラスのメソッドの利用法を学ぶ。

手順

1) プロジェクトの作成

Visual Studio 2013 を起動したら，［ファイル］→［新規作成］→［プロジェクト］と辿って，プロジェクトを作成する。『新しいプロジェクト』ダイアログボックスでは，プログラミング言語を『Visual C#』，プロジェクトテンプレートとしては，『Windows フォームアプリケーション』を選択し，『名前』を「Eleventh」に書き換え，『場所』が「H:¥Documents¥Visual Studio 2013¥Projects」となっていることを確認してから『OK』を押す（詳細は第 1 回の教材を参照）。



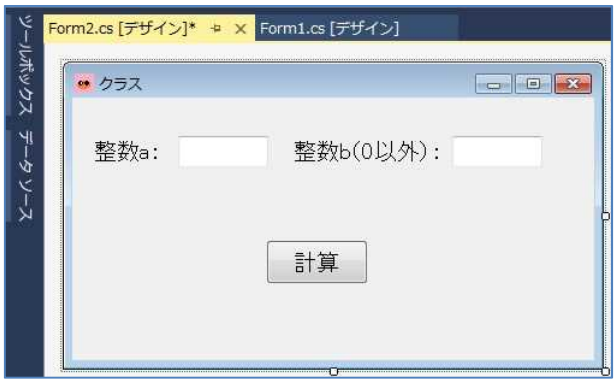
2) コントロールの配置及びフォームの作成

今後，フォーム上に配置するコントロールのプロパティのフォントサイズは全て 14 ポイントに変更するものとする。

Form1 上にラベルを 1 つ，テキストボックスを 1 つ，ボタンを 3 つ貼り付ける。それぞれのプロパティは次の様に設定する。

【Form1】Text　「**メソッド**」
　　Icon　自作のアイコン
【button1】Text　「**入力**」
【button2】Text　「**結果表示**」
【button3】Text　「**次へ**」

[プロジェクト] → [Windows フォームの追加]を選択して Form2 を追加し(方法は第 6 回の教材を参照)，ラベルを 2 つ，テキストボックスを 2 つ，ボタンを 1 つ貼り付ける。それぞれのプロパティは次の様に設定する。

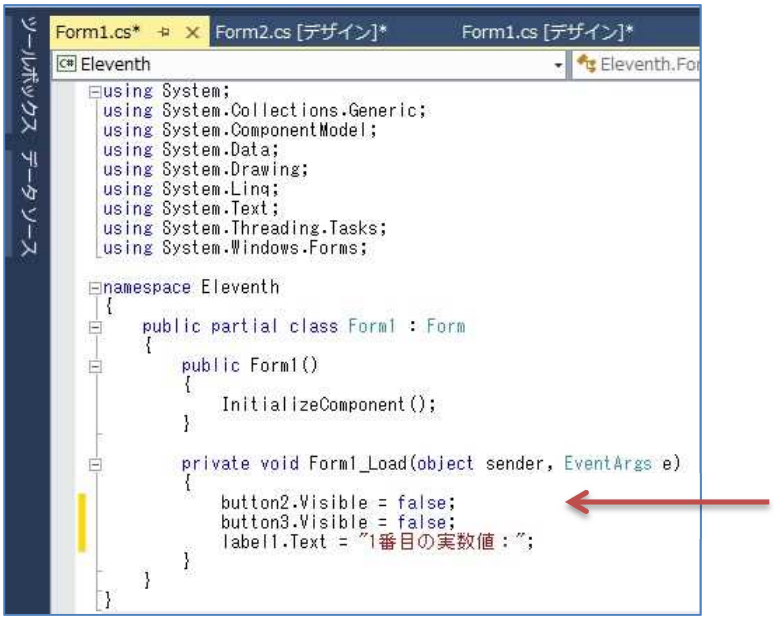


【Form2】Text 「**クラス**」
　　Icon　自作のアイコン
【label1】Text 「**整数 a：**」
【label2】Text 「**整数 b（0 以外）：**」
【button1】Text 「**計算**」

3) コーディング

Form1 のフォームデザイナー上でコントロールが貼られていない箇所をダブルクリックして Form1.cs のプログラムのソースコードを表示する。Form1_Load メソッドのブロック内に Form1 が読み込まれた際の処理として，『button2』及び『button3』の『Visible』プロパティに「false」を設定して非表示にする処理及び『label1』の『Text』プロパティに初期の文字列を設定する処理（赤枠の部分）を記述する。

```
private void Form1_Load(object sender, EventArgs e)
{
    button2.Visible = false;
    button3.Visible = false;
    label1.Text = "1 番目の実数値：";
}
```



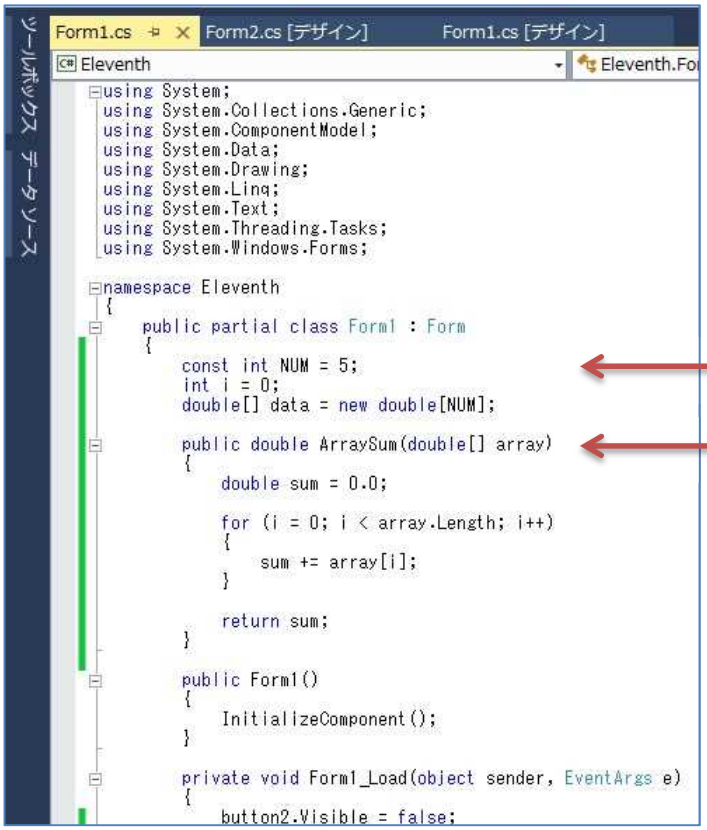
Form1 のフォームデザイナー上で『button1』及び『button2』をダブルクリックして，Form1.cs

のプログラムのソースコードを表示する。先ず，クラス全体に適用可能な定数，変数及び配列の宣言をクラスの冒頭に記述する。NUM，i，data の役割は前回の Form2.cs のプログラムと同様である。

```
const int NUM = 5;
int i = 0;
double[] data = new double[NUM];
```

これに続けて，引き渡された配列の全要素の値の和を求める ArraySum メソッドを記述する。引数の double 型の配列 array はこのメソッドの中が有効範囲の仮引数で，呼び出し側で与えられた実引数への参照が渡される。なお，引数が値型の変数の場合には，実引数の値が渡される。

```
public double ArraySum(double[] array)
{
    double sum = 0.0;

    for (i = 0; i < array.Length; i++)
    {
        sum += array[i];
    }

    return sum;
}
```



array.Length は，配列 array の Length プロパティで，要素数を表す。

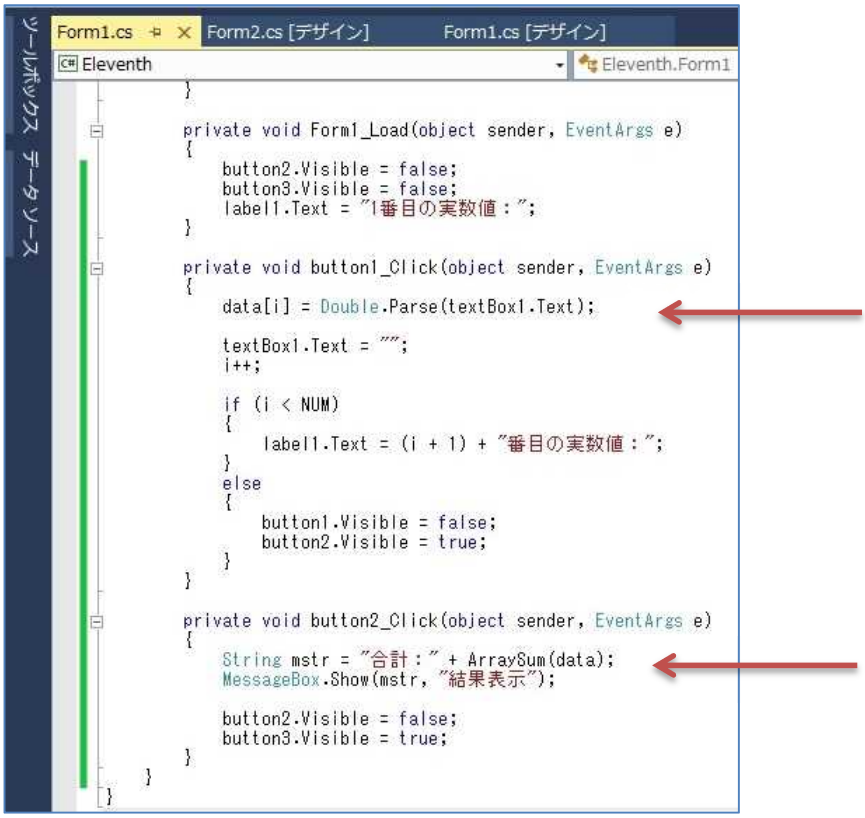
なお，メソッドのデータ型は戻り値（return で呼び出し側に返す値）のデータ型である。

次に，button1_Click メソッド及び button2_Click メソッドのブロック内にそれぞれのボタン

がクリックされた際の処理（赤枠の部分）を記述していく。

```
private void button1_Click(object sender, EventArgs e)
{
    data[i] = Double.Parse(textBox1.Text);

    textBox1.Text = "";
    i++;

    if (i < NUM)
    {
        label1.Text = (i + 1) + "番目の実数値：";
    }
    else
    {
        button1.Visible = false;
        button2.Visible = true;
    }
}
```

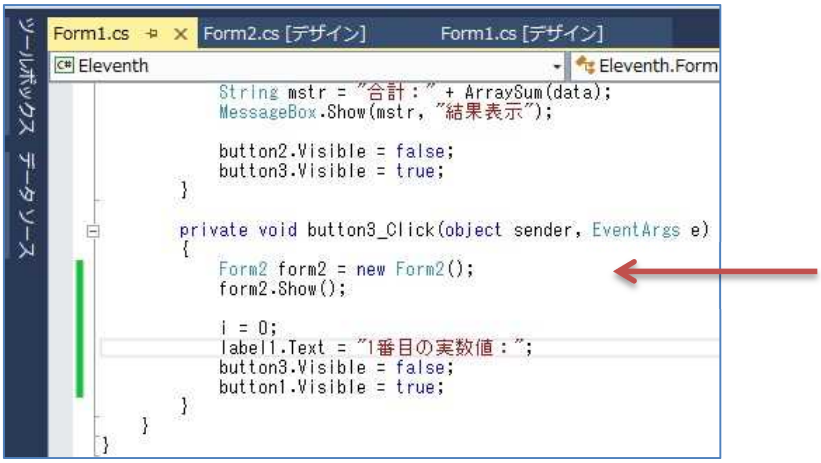
```
private void button2_Click(object sender, EventArgs e)
{
    String mstr = "合計：" + ArraySum(data);
    MessageBox.Show(mstr, "結果表示");

    button2.Visible = false;
    button3.Visible = true;
}
```



button2_Click メソッド中の ArraySum(data)は，実引数 data を引き渡してメソッド呼び出す。その結果として，ArraySum メソッドからは sum の値が返されてくる。

更に，フォームデザイナー上で『button3』をダブルクリックして，コードにイベントハンドラを作成する。button3_Click メソッドのブロック内にボタンがクリックされた際の処理（赤枠の部分）を記述していく。

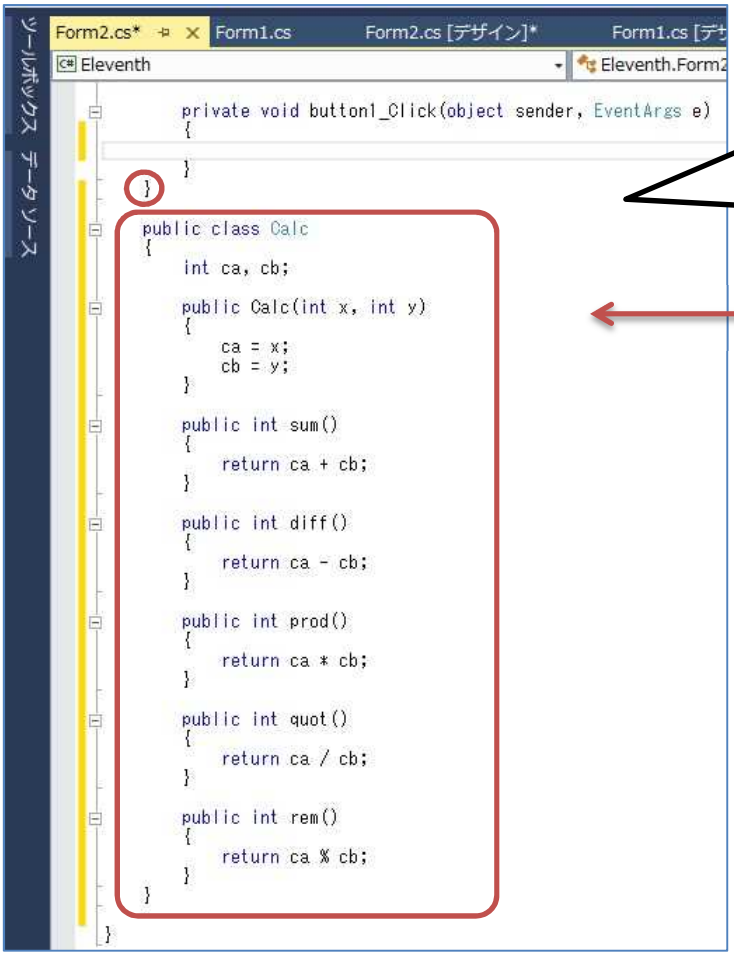
```
        private void button3_Click(object sender, EventArgs e)
        {
            Form2 form2 = new Form2();
            form2.Show();

            i = 0;
            label1.Text = "1 番目の実数値：";
            button3.Visible = false;
            button1.Visible = true;
        }
```



Form2 のフォームデザイナー上で『button1』をダブルクリックして，Form2.cs のプログラムのソースコードを表示する。先ず，Form2 クラスより下に，四則演算及び剰余算のメソッドを含む Calc クラスを記述する。なお，Visual Studio でフォームアプリケーションを作成する場合には，フォームに関するクラスは先頭になければならないという制約があるので注意が必要である。

```
public class Calc
{
    int ca, cb;

    public Calc(int x, int y)
    {
        ca = x;
        cb = y;
    }
```

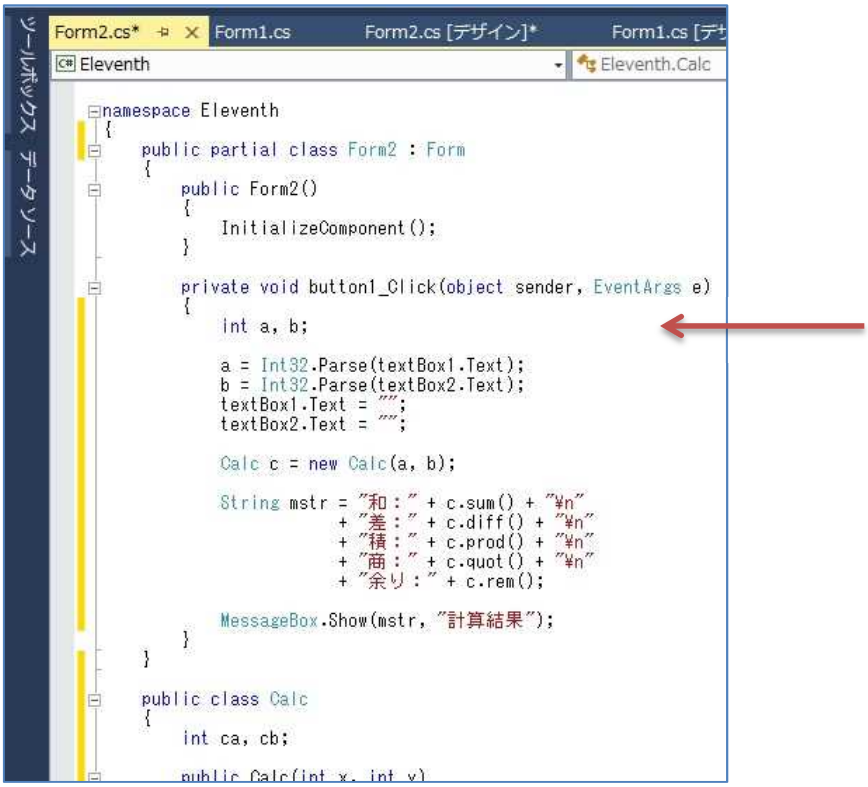
```
        public int sum()
        {
            return ca + cb;
        }

        public int diff()
        {
            return ca - cb;
        }

        public int prod()
        {
            return ca * cb;
        }

        public int quot()
        {
            return ca / cb;
        }

        public int rem()
        {
            return ca % cb;
        }
    }
```



Form2 クラスの終わりの中括弧『 } 』の後ろで改行して，その下に記述する。

次に，コードにイベントハンドラを作成する。button1_Click メソッドのブロック内にボタンがクリックされた際の処理（赤枠の部分）を記述していく。Calc クラスのコンストラクタは int 型の引数を 2 個持つので，new Calc(a, b)の様にコンストラクタの引数に合わせた実引数を与えてインスタンスを生成しなければならない。Calc クラスのメンバーにアクセスするには，インスタンス変数 c とメンバー名とを演算子『.』で繋ぐ（第 6 回の教材 p.1 を参照）。

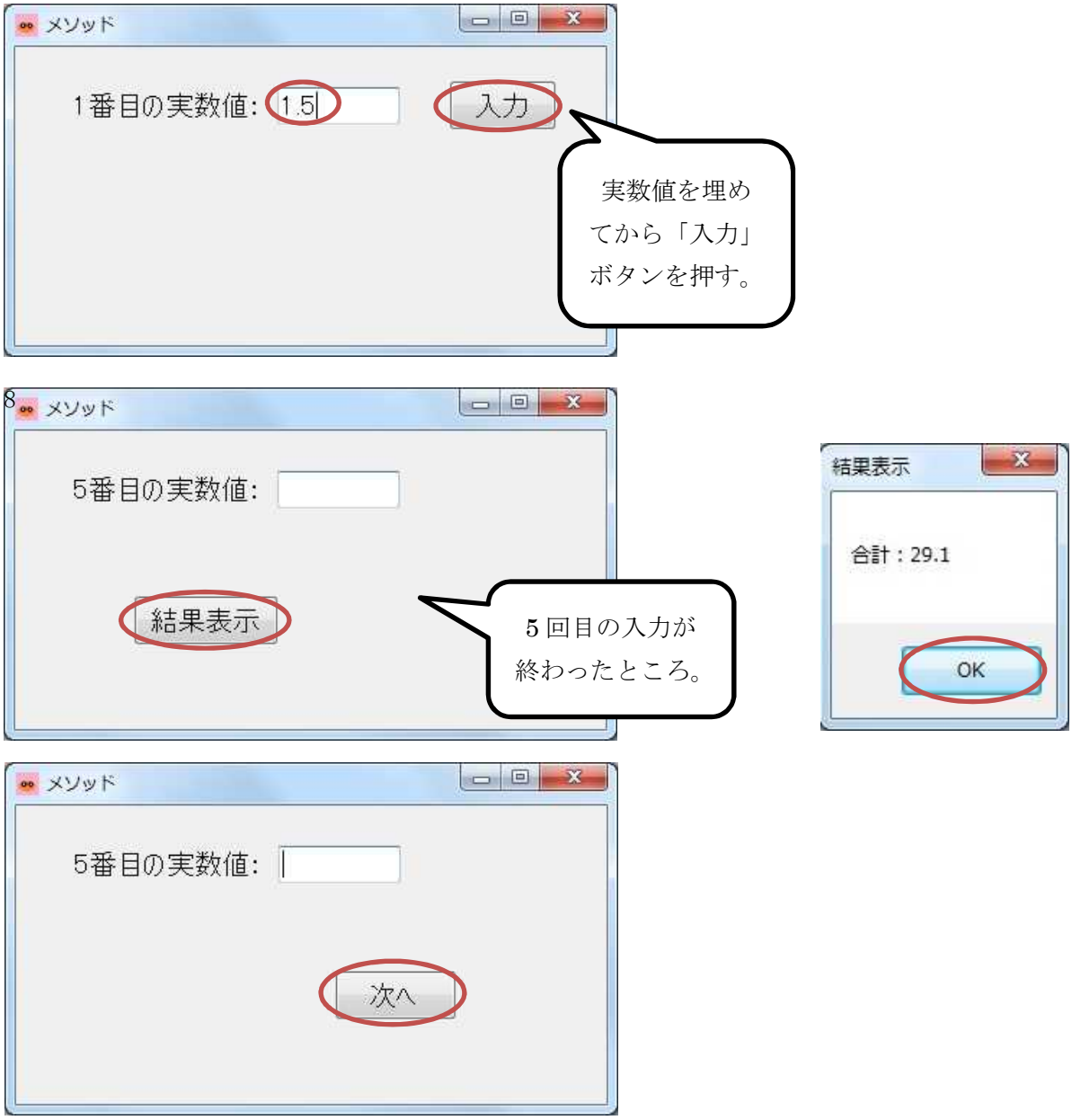
```
private void button1_Click(object sender, EventArgs e)
{
    int a, b;

    a = Int32.Parse(textBox1.Text);
    b = Int32.Parse(textBox2.Text);
    textBox1.Text = "";
    textBox2.Text = "";

    Calc c = new Calc(a, b);

    String mstr = "和：" + c.sum() + "¥n"
                + "差：" + c.diff() + "¥n"
                + "積：" + c.prod() + "¥n"
                + "商：" + c.quot() + "¥n"
                + "余り：" + c.rem();

    MessageBox.Show(mstr, "計算結果");
}
```



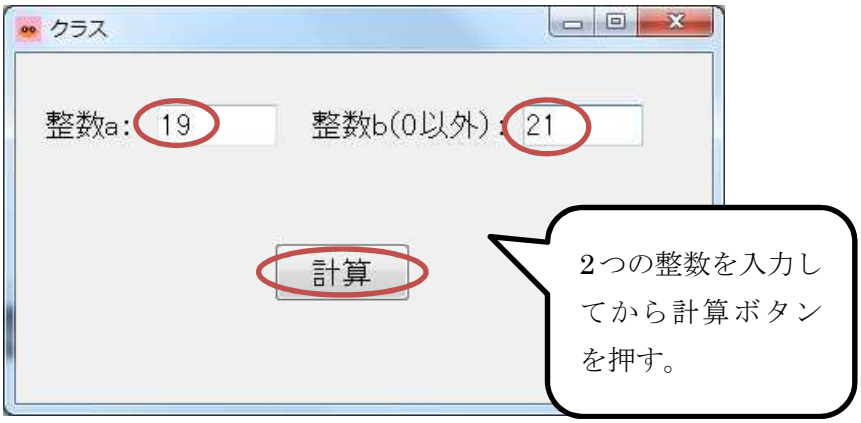
4）プログラムの実行・最終確認

『すべてを保存』ボタンを押してから，『開始』ボタンを押して，プログラムを実行する。
エラーが出ている場合には，修正してから保存，開始と進む。

今回のプログラムでは，実数値をテキストボックスに入力して入力ボタンを押すという操作を 5 回繰り返すと結果表示のボタンが現れる。（図はその一例）



メッセージボックスの OK ボタンを押すと，フォームに『次へ』ボタンが現れる。これを押すと，Form2 が表示される。Form2 では 2 つの整数を与えて，四則演算及び剰余算を行った結果を表示する。（次のページの図はその一例）



確認を終えたら，プログラムを終了する。

【ファイルが保存されている場所】　H:¥Documents¥Visual Studio 2013¥Projects¥Eleventh¥Eleventh

提出物：

1）フォームのデザインファイル Form1.Designer.cs をメールに添付して提出する。
2）フォームを含むソースファイル Form1.cs をメールに添付して提出する。
3）フォームのデザインファイル Form2.Designer.cs をメールに添付して提出する。
4）フォームを含むソースファイル Form2.cs をメールに添付して提出する。
5）質問を記述したファイル Questions_11th.txt に解答を書き込んで保存し，メールに添付して提出する。