

# 第2回C#講習会

---

2018/5/15

# 前回の復習

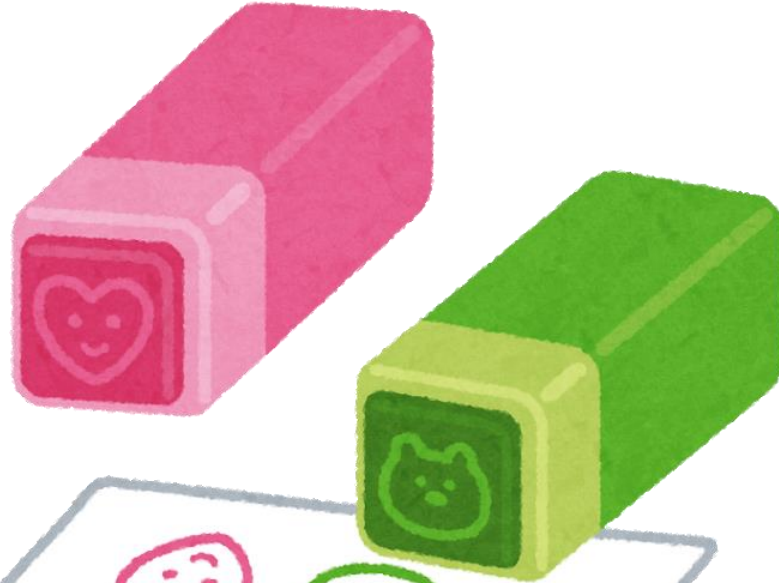
オブジェクト指向ってなあに？

わい「プログラムをオブジェクト(モノ)として考える概念だよ(キラッ)」

おまいら「????????????????????????????????」

# 前回の復習

クラス(設計図)



オブジェクト(モノ)



# 今回やること

せっかくC#を学ぶのだから、  
簡単なアプリケーションを作ってみよう！

**基本構造や標準機能とか小難しいことは後にしよう！**

# 今回やること

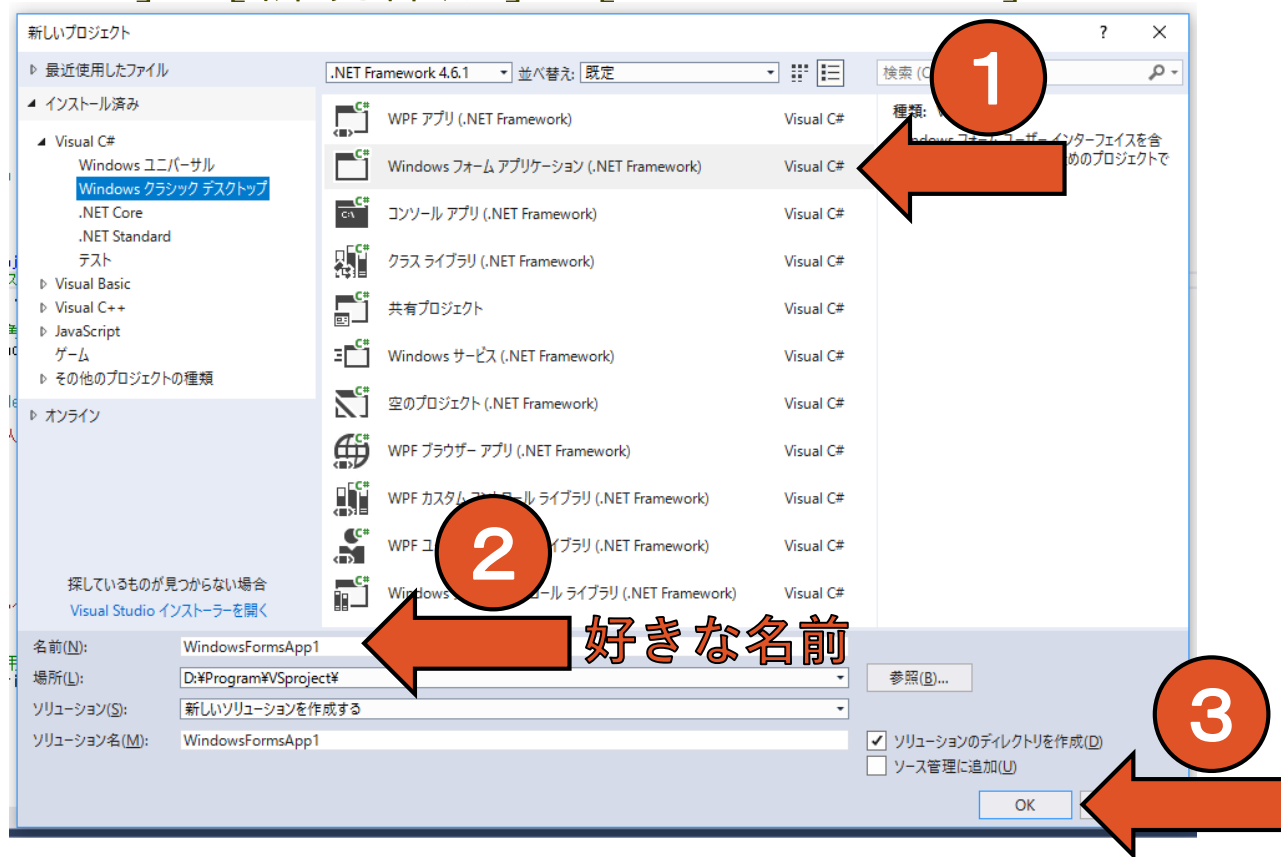
\_\_人人人人人\_\_  
> 突然の死 <  
—Y^Y^Y^Y^Y—

↑これを作ります↑  
(ふざけてないよ？本当だよ？)

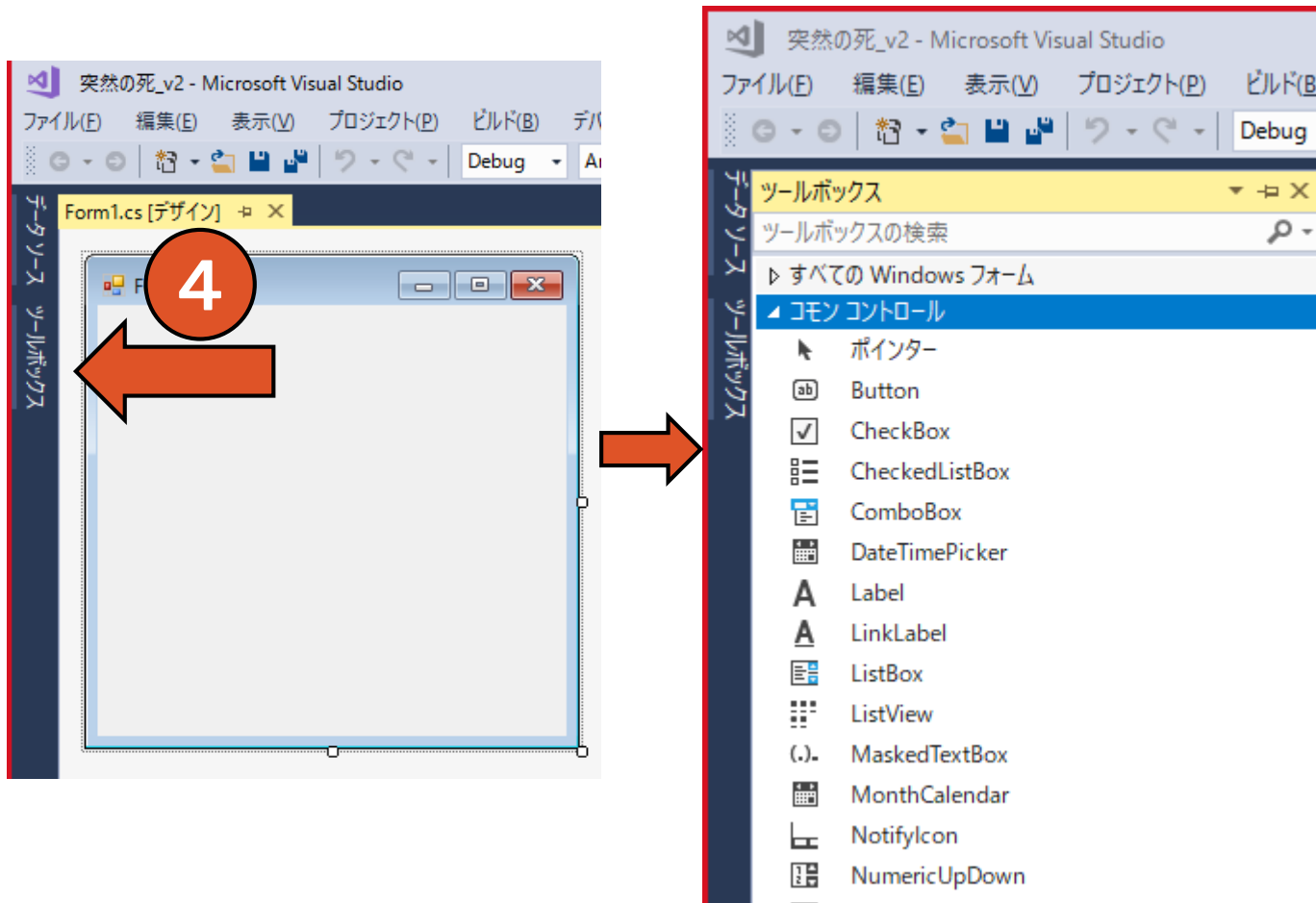
# 作る . . . ! ! ! !

さっそく作っていこう

右上の[ファイル]→[新規作成]→[プロジェクト]から



# 作る ■ ■ ■ ! ! ! !



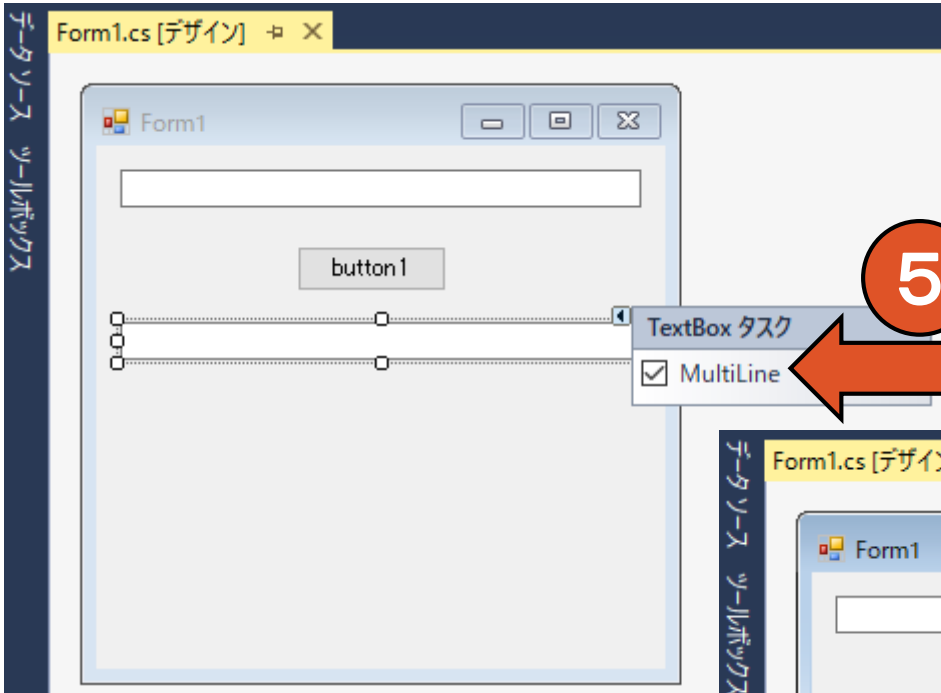
ここでは画面設計をするよ。

[コモンコントロール]の中から

- TextBox × 2
- Button × 1

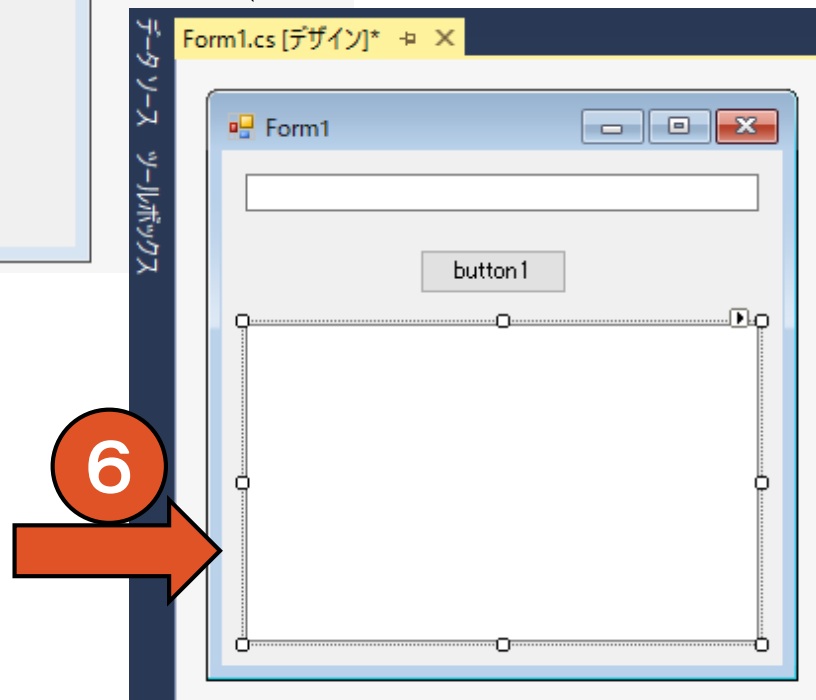
好きなように置いてみよう！！

# 作る . . . ! ! ! !



2つの内、片方の[TextBox]を  
[MultiLine]にしよう。

※複数行に渡っての文字列に対応するよ

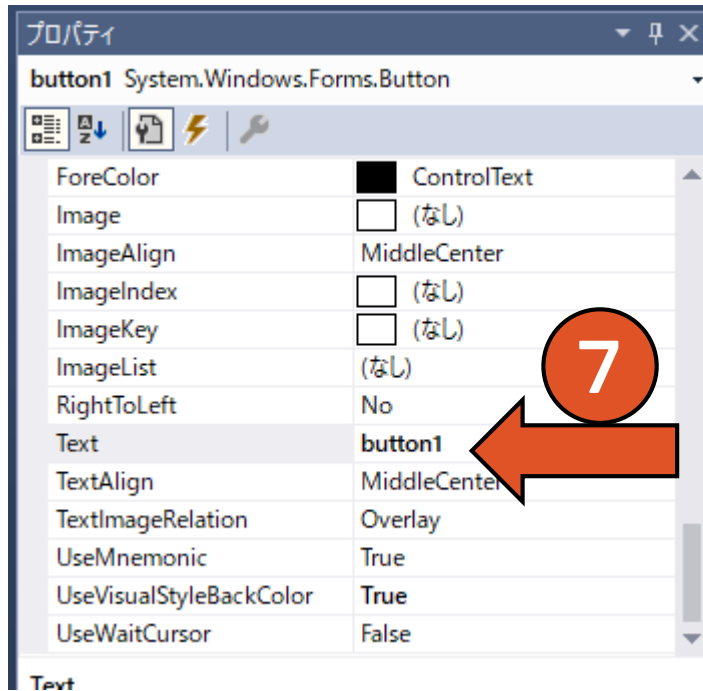


[MultiLine]にした[TextBox]を  
グーッと下に伸ばそう

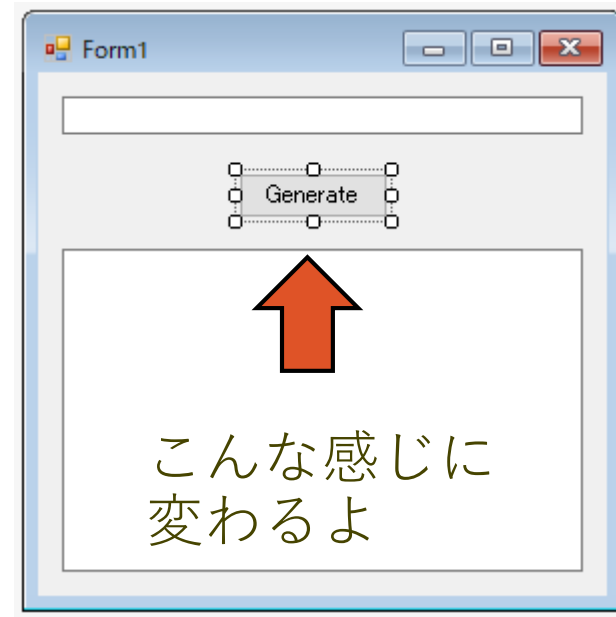
余裕があったら、  
好きな風に大きさや配置を変えて  
みよう！



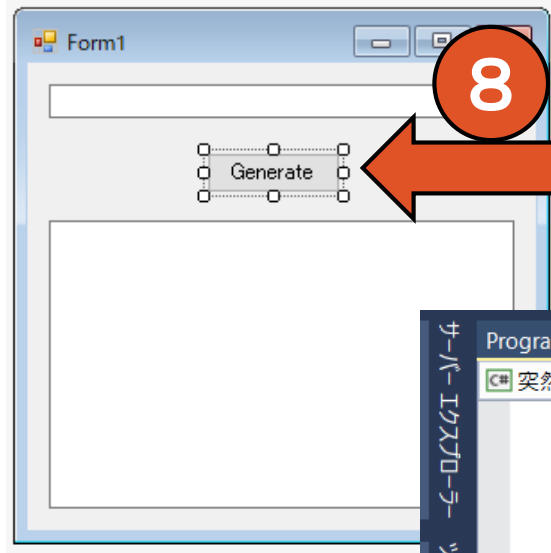
# 作る . . . ! ! ! !



フォーム上の[button1]を1回クリックしたら  
左下のプロパティの[Text]という所を好きな風に変えてみよう。



# 作る . . . ! ! ! !



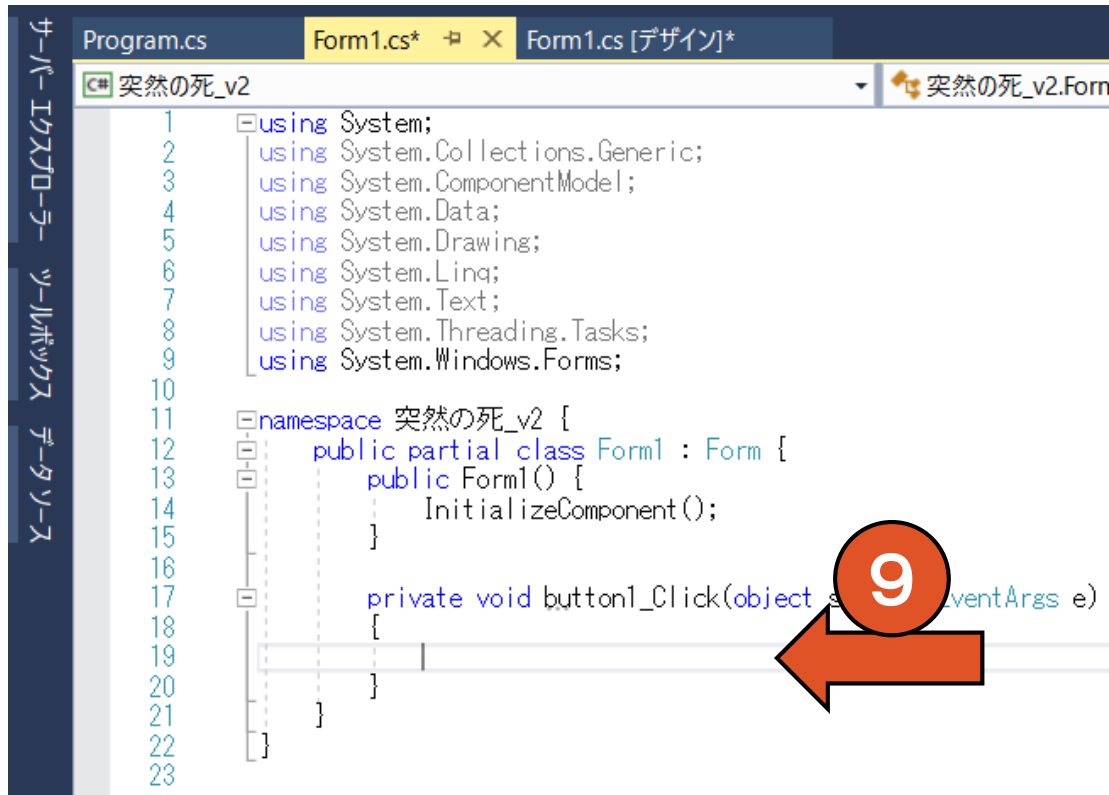
ダブルクリック

```
Program.cs  Form1.cs*  Form1.cs [デザイン]*
突然の死_v2
突然の死_v2.Form1
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11 namespace 突然の死_v2 {
12     public partial class Form1 : Form {
13         public Form1() {
14             InitializeComponent();
15         }
16
17         private void button1_Click(object sender, EventArgs e)
18         {
19
20         }
21     }
22 }
23
```

すると左の画像がように、プログラムが自動生成されるよ。

# 作る . . . ! ! ! !

⑨の部分に次のプログラムを入力しよう。



```
Program.cs | Form1.cs* | Form1.cs [デザイン]*
C# 突然の死_v2
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace 突然の死_v2 {
12     public partial class Form1 : Form {
13         public Form1() {
14             InitializeComponent();
15         }
16
17         private void button1_Click(object sender, EventArgs e)
18         {
19
20         }
21     }
22 }
23
```

// 一番上にある入力用のテキストボックスから文字列を取得

```
string srcText = textBox1.Text;
```

// 入力されたテキストを半角は1文字、全角2文字として取得

```
int size = Encoding.GetEncoding("Shift_JIS").GetByteCount(srcText);
```

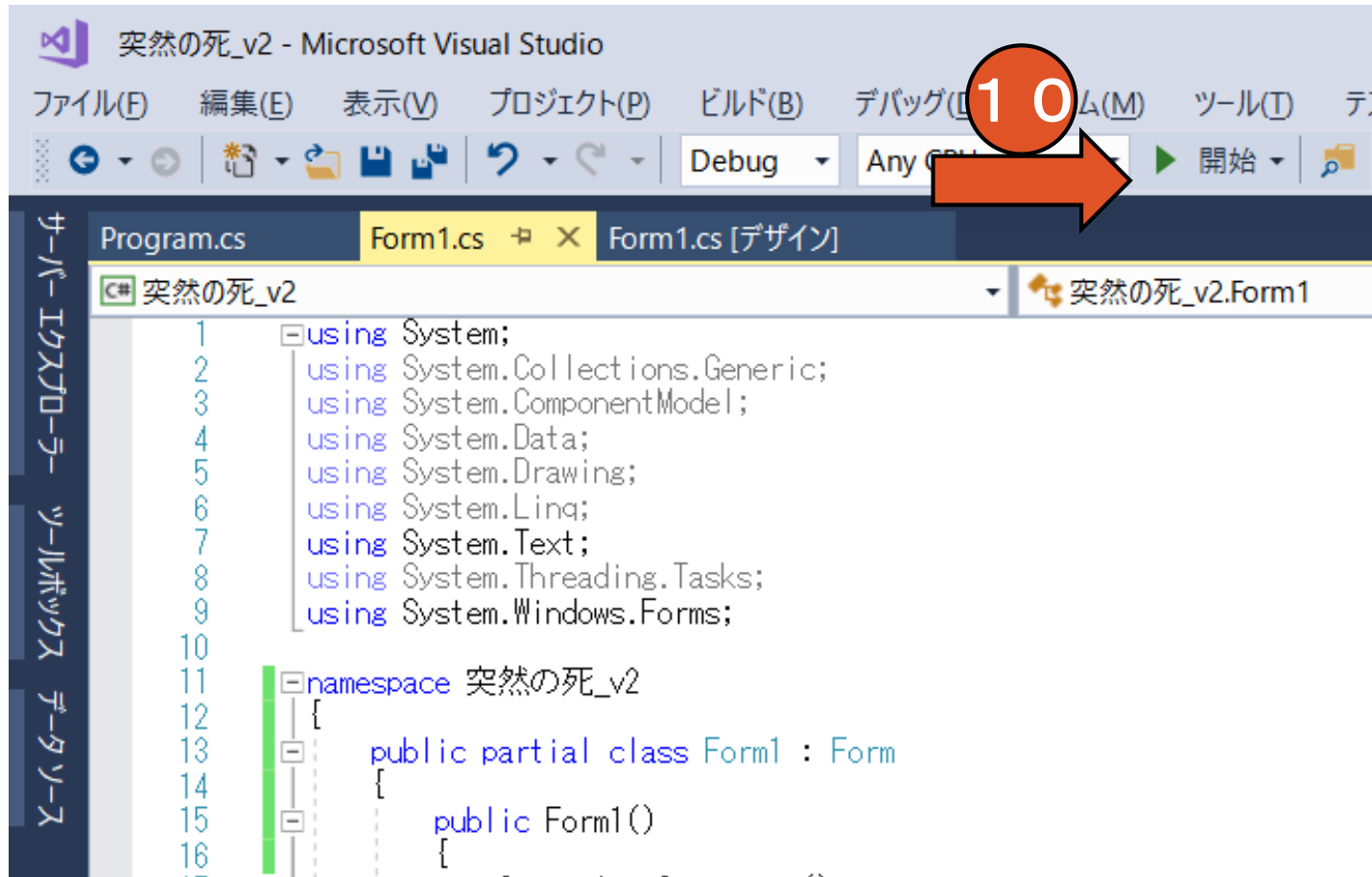
// トゲ吹き出しの作成

```
var cnv = new StringBuilder();
cnv.Append("__");
cnv.Insert(cnv.Length, "人", size / 2);
cnv.Append("__");
cnv.AppendLine();
cnv.Append("> ");
cnv.Append(srcText);
cnv.Append("<");
cnv.AppendLine();
cnv.Append(" ");
cnv.Insert(cnv.Length, "Y^", size / 2);
cnv.Append(" ");
```

// 作成したテキストを一番下の出力用テキストボックスに設定

```
textBox2.Text = cnv.ToString();
```

# 作る . . . ! ! ! !



最後にビルド&実行をして  
動作を確認してみよう

# プログラム解説

講習会の時の自分、まかせた

# さいごに

実はこのアプリ...完全なものではありません。

複数行に渡って入力できなかったり、

長すぎると形が破綻したり

etc...

**自分なりに改良してみよう！**