

2024 年 4 月 25 日 (木) 実施

## イベントハンドラ

### イベントハンドラとは

Windows フォーム上のコントロールに対して「クリックされた」とか「文字列を変更された」とかいったイベントを行った際に、それを受け取って処理を行うメソッドをイベントハンドラと呼ぶ。

### 本日の課題

今回の授業では、**フォームデザイナーで該当するコントロールをダブルクリックして、コードエディタに表示されたイベントハンドラの処理を記述する方法**を学ぶ。

\* Visual Studio ではイベントハンドラは **Form1.cs** とは別のファイルで管理されていて、単に **Form1.cs** に手入力して記述しても機能しないので注意が必要である。

### 手順

#### 1) プロジェクトの作成

第 1 回の教材の pp.3-4 と同様にして、新規のプロジェクトを作成する。前回開いていた **First** を閉じていなかった場合には、『ファイル』→『ソリューションを閉じる』を選択すれば、新規のプロジェクトを作成できる画面が表示される。

今回は、『プロジェクト名』を「**JimboCalc**」(**Jimbo** の箇所は自分の名前に置き換える)とする。また、今後、暫くは今回作成したプロジェクトに機能を追加していく。

#### 2) コントロールの配置・プロパティの設定

まず、フォーム上のコントロールの無い箇所をクリックすることでフォーム (Form1) を選択し、『プロパティウィンドウ』で、次のプロパティを設定する。

(プロパティ名)	(値)
Text	<b>JimboCalc</b> ( <b>Jimbo</b> の箇所は自分の名前に置き換える)
Font	12pt
Size	
Width	265
Height	406

次に、『ツールボックス』の『コモンコントロール』から『Button』を選択して、フォームに貼り付け、次のプロパティを設定する。この時のボタンの名称は **button1** となる。

Text	0
Size	
Width	100

Height	35
Location	
X	12
Y	321

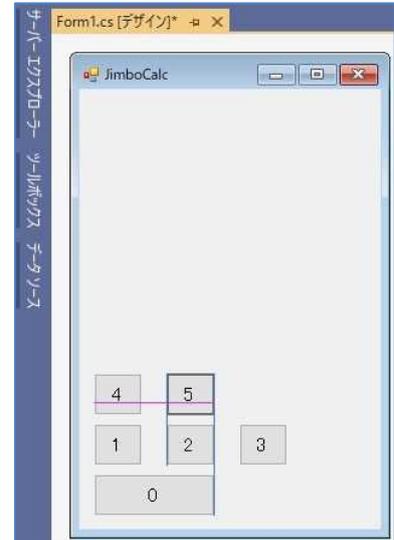
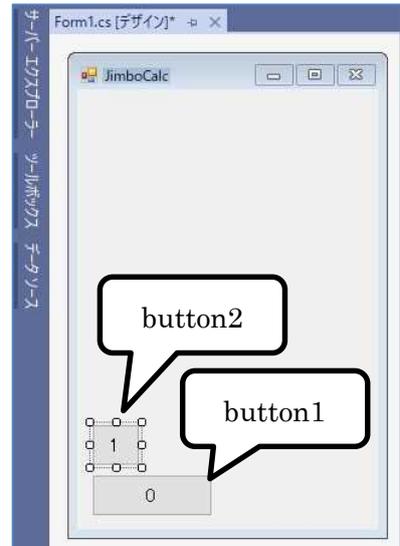
続けて、もう 1 個ボタンを選択して貼り、次のプロパティを設定する。

button2 :

Text	1
Size	
Width	40
Height	35
Location	
X	12
Y	279

この後は同じサイズのボタンを 8 個配置するので、button2 を右クリックして**コピー**し、フォームの何もないところをクリックしてから、右クリックして**貼り付け**、プロパティを設定する。一列貼り付けたら、右図の様に位置合わせで大まかに配置をすることも出来るが、正確に合わせる為には Location プロパティで調整する。

それぞれのボタンのプロパティを一覧表として挙げておく。

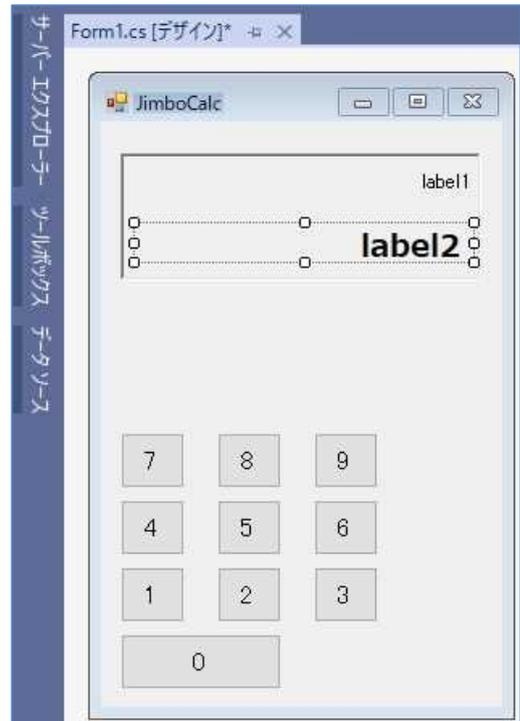
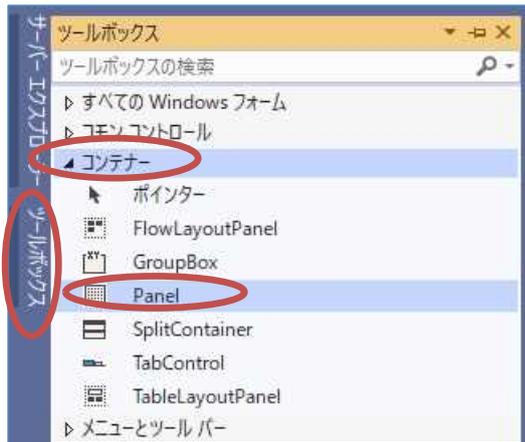


【ボタンのプロパティ】

button3		button4		button5		button6	
Text	2	Text	3	Text	4	Text	5
X	72	X	132	X	12	X	72
Y	279	Y	279	Y	237	Y	237
button7		button8		button9		button10	
Text	6	Text	7	Text	8	Text	9
X	132	X	12	X	72	X	132
Y	237	Y	195	Y	195	Y	195

次に、『ツールボックス』の『コンテナ』から『Panel』を選択して、フォームに貼り付け、次のプロパティを設定する。この時のボタンの名称は panel1 となる。

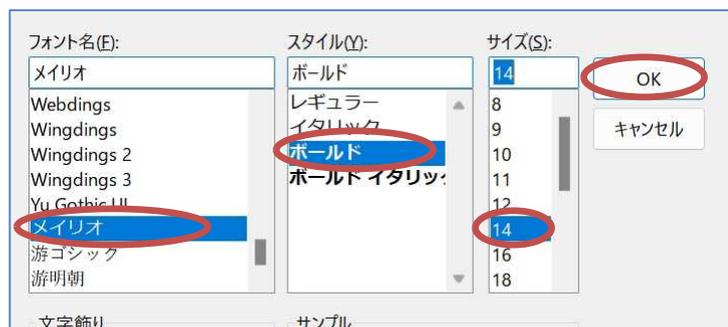
(図は次のページ)



BorderStyle	Fixed3D
Size	
Width	224
Height	80
Location	
X	12
Y	20

更に、『ツールボックス』の『コモンコントロール』から『Label』を選択して、パネルの上に重ねて貼り付けることを 2 回繰り返す。次のプロパティを設定する。この時のラベルの名称はそれぞれ label1, label2 となる (右上の図)。

label1 :		label2 :	
Font	9pt	Font	メイリオ ボールド 14pt
TextAlign	MiddleRight	TextAlign	MiddleRight
AutoSize	false	AutoSize	false
Size		Size	
Width	208	Width	208
Height	12	Height	22
Location		Location	
X	10	X	8
Y	10	Y	43



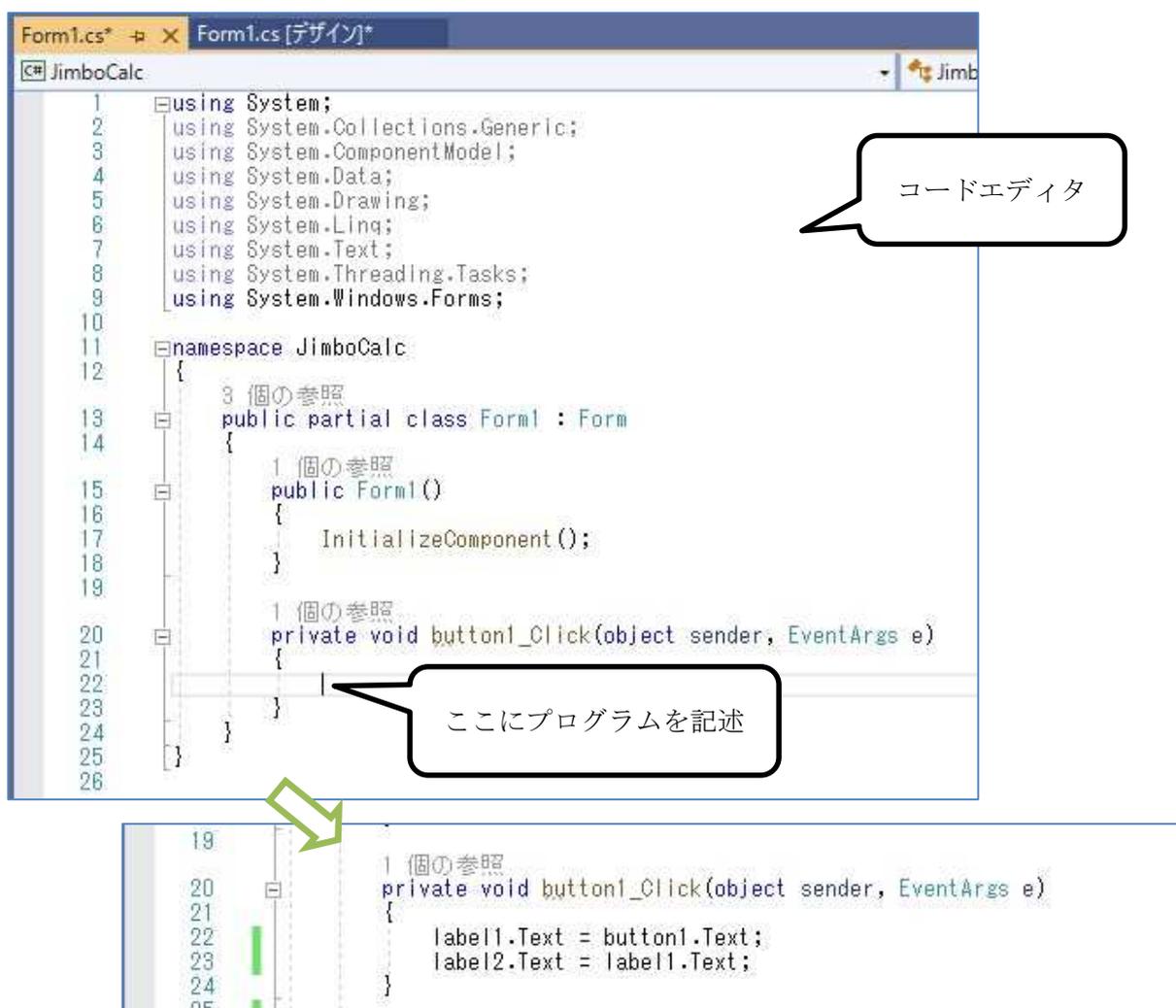
### 3) コーディング

フォームデザイナー上で『0』が表示されているボタン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。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。

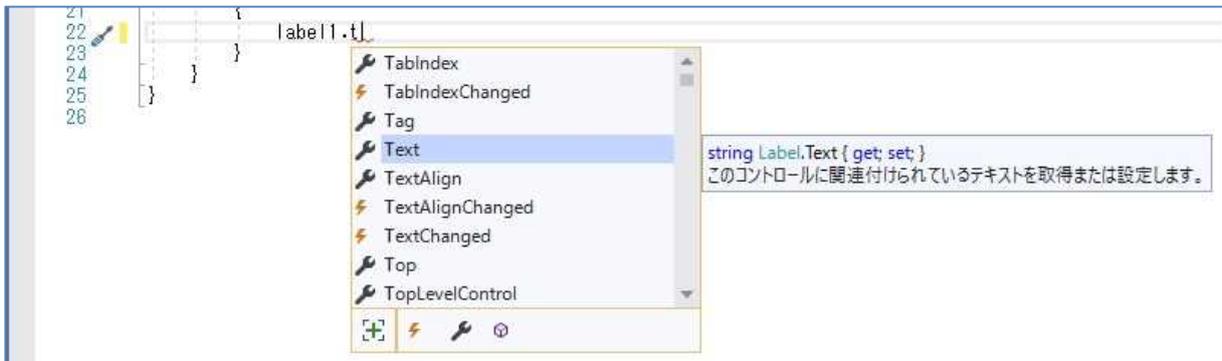
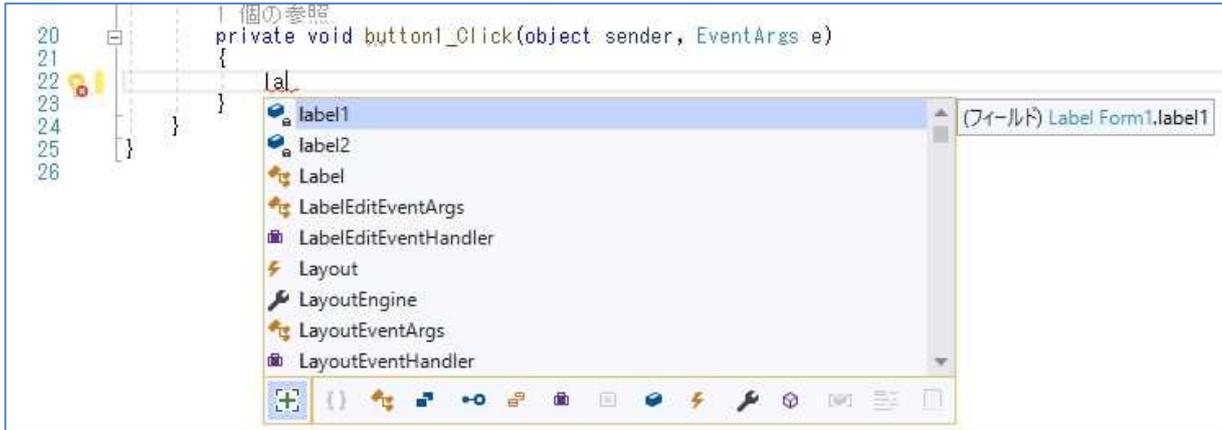
```
private void button1_Click(object sender, EventArgs e)
{
    label1.Text = button1.Text;
    label2.Text = label1.Text;
}
```

ここでは、ボタンの Text プロパティの値を label1 という名前のラベルの Text プロパティに当てはめて設定している。また、label1 の Text プロパティの値を label2 の Text プロパティに代入している。

コードエディタでは文字列を途中まで打つと候補が現れる補完機能がある。適切なものを選んで Enter キーを打つとプログラムにそれが入力される。



(補完機能の図は次のページ)

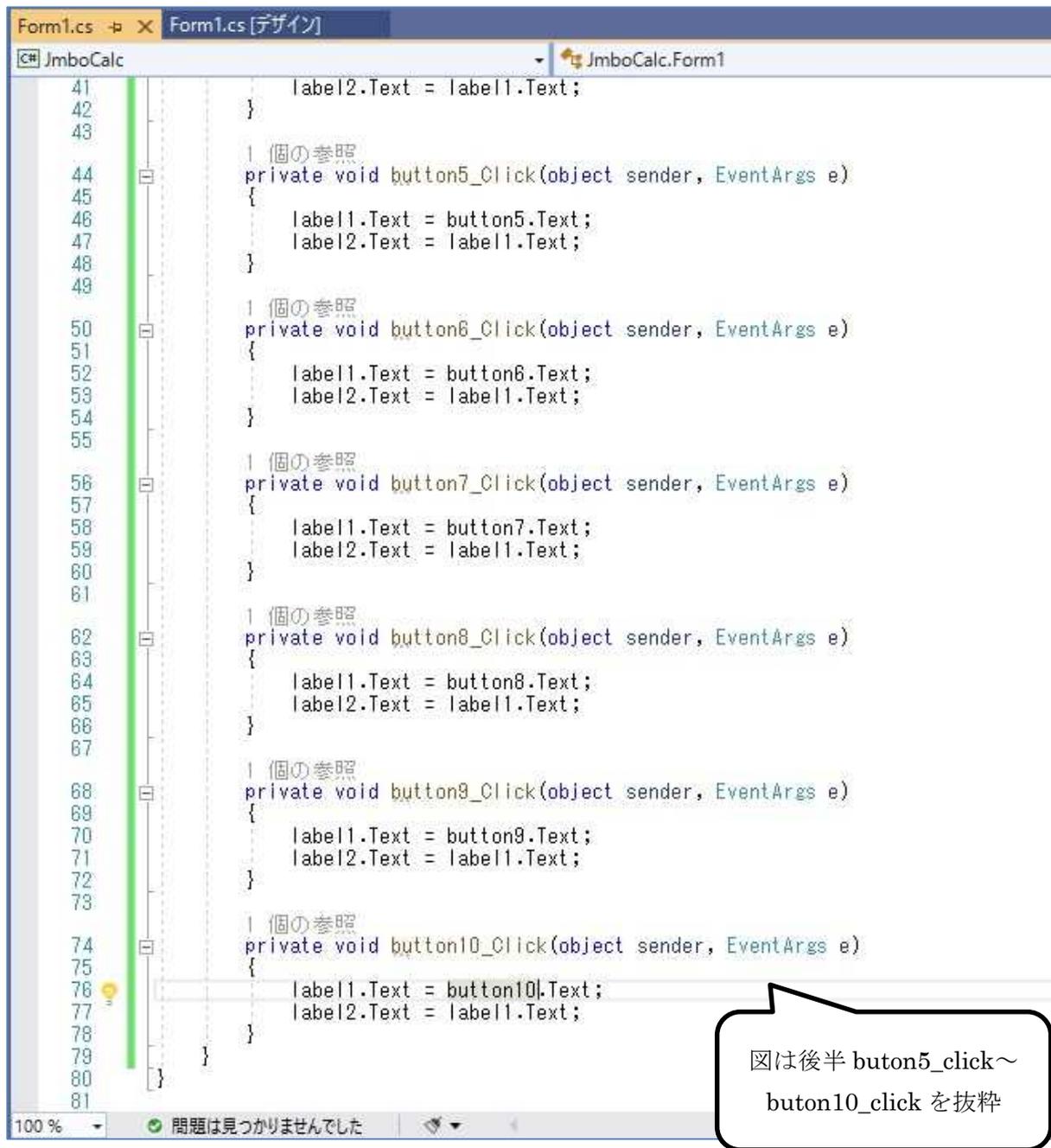


ここで、『Form1.cs の保存』ボタンを押してから、『開始』ボタンを押して、プログラムを実行してみる。



確認が済んだら、閉じるボタンを押して、一旦プログラムを終了する。

更に、フォームデザイナー上で『1』から『9』までのボタンを1個ずつダブルクリックしてはコードエディタでイベントハンドラの処理を記述することを繰り返す。その際、**buton2\_click** ならば、**button2.Text** を用いるといったことに注意する。



ここで、『Form1.cs の保存』ボタンを押してから、『開始』ボタンを押して、プログラムを実行して、『1』から『9』までのボタンの動作を確かめる。

(最終的な図は次のページ)



これに相当する図(スクリーンショット)を自分のアプリの実行時に作成して課題提出用にする。(Alt キーを押しながら PrtScr キーを打って画面のイメージを取り込み, Windows アクセサリのペイントで jpg ファイルとして保存する。)

次回はこのプロジェクトの作業の続きを行うので、『ファイル』→『終了』を選択し、ダイアログの閉じるボタン、開発環境の閉じるボタンを押して終了する。



なお、提出ファイルの場所は作成したプロジェクト毎にフォルダが異なるので、C ドライブの [ユーザー] → [ユーザー名] → [source] → [repos] → JimboCalc → JimboCalc と辿る。

\* Jimbo は自分の名前に置き換えて考える。

**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) コードエディタで編集したソースファイル **Form1.cs** をメールに添付して提出する。
- 3) 実行の最終結果のスクリーンショット **第 2 回実行結果.jpg** (.png も可) をメールに添付して提出する。
- 4) 質問を記述したファイル **Prog2\_Questions\_2nd.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。