

松山大学 経営学部

# 情報処理論（応用）



## 第4回 文字列の入力



講師 檀 裕也

<http://www.cc.matsuyama-u.ac.jp/~dan/application/>

2006年10月 3日

# 出席確認

- 出席確認フォームから学籍番号および氏名を送信せよ。

## 出席確認フォーム

<http://www.cc.matsuyama-u.ac.jp/~dan/application/attendance.html>

# 前回の課題

## くじ引きプログラム (ex\_20060929)

- 4つのボタンのうち、あたりは1つ
- “当たり”と“はずれ”で別のメッセージを表示



2006年10月 3日

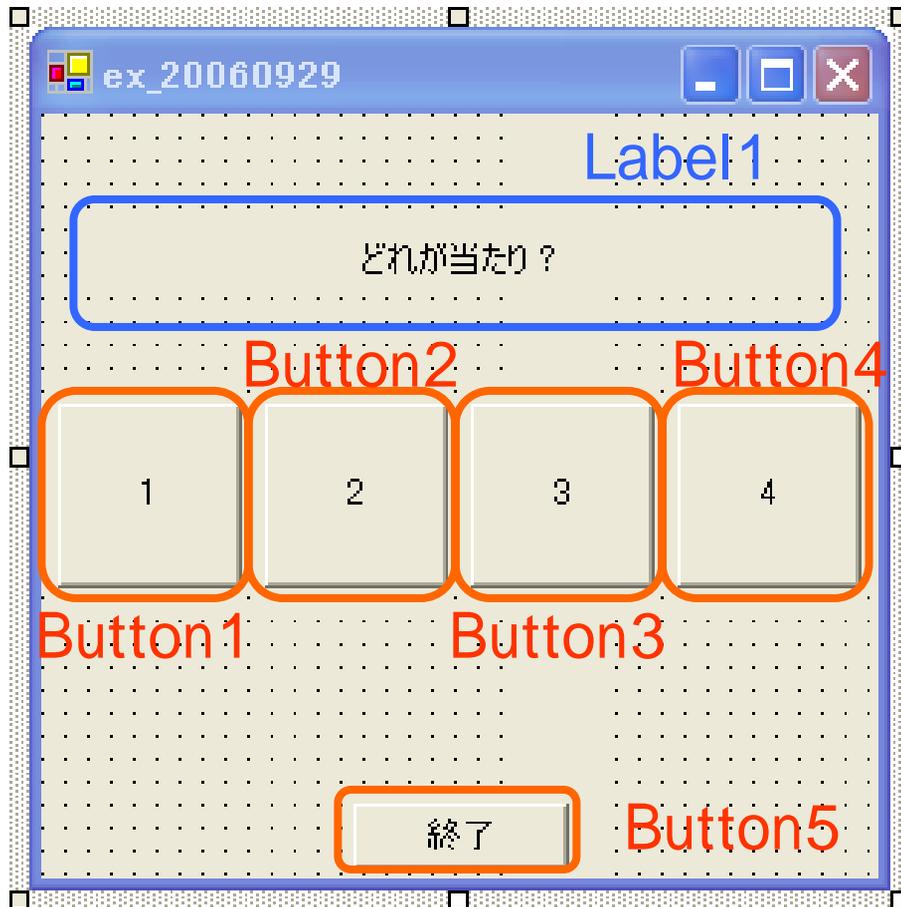


情報処理論 (応用)



3

# 解答例 (フォーム)



- ラベルおよびボタンの名前は、機能が分かりやすいものに変更したほうがよい。

# 解答例 (コード)

- 各ボタンに対応するコードを記述する

```
Public Class Form1
    Inherits System.Windows.Forms.Form

    Windows フォーム デザイナで生成されたコード

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Label1.Text = "はずれ、残念。"
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        Label1.Text = "はずれ、残念。"
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
        Label1.Text = "はずれ、残念。"
    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click
        Label1.Text = "大当たり!"
    End Sub

    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button5.Click
        Me.Dispose()
    End Sub
End Class
```

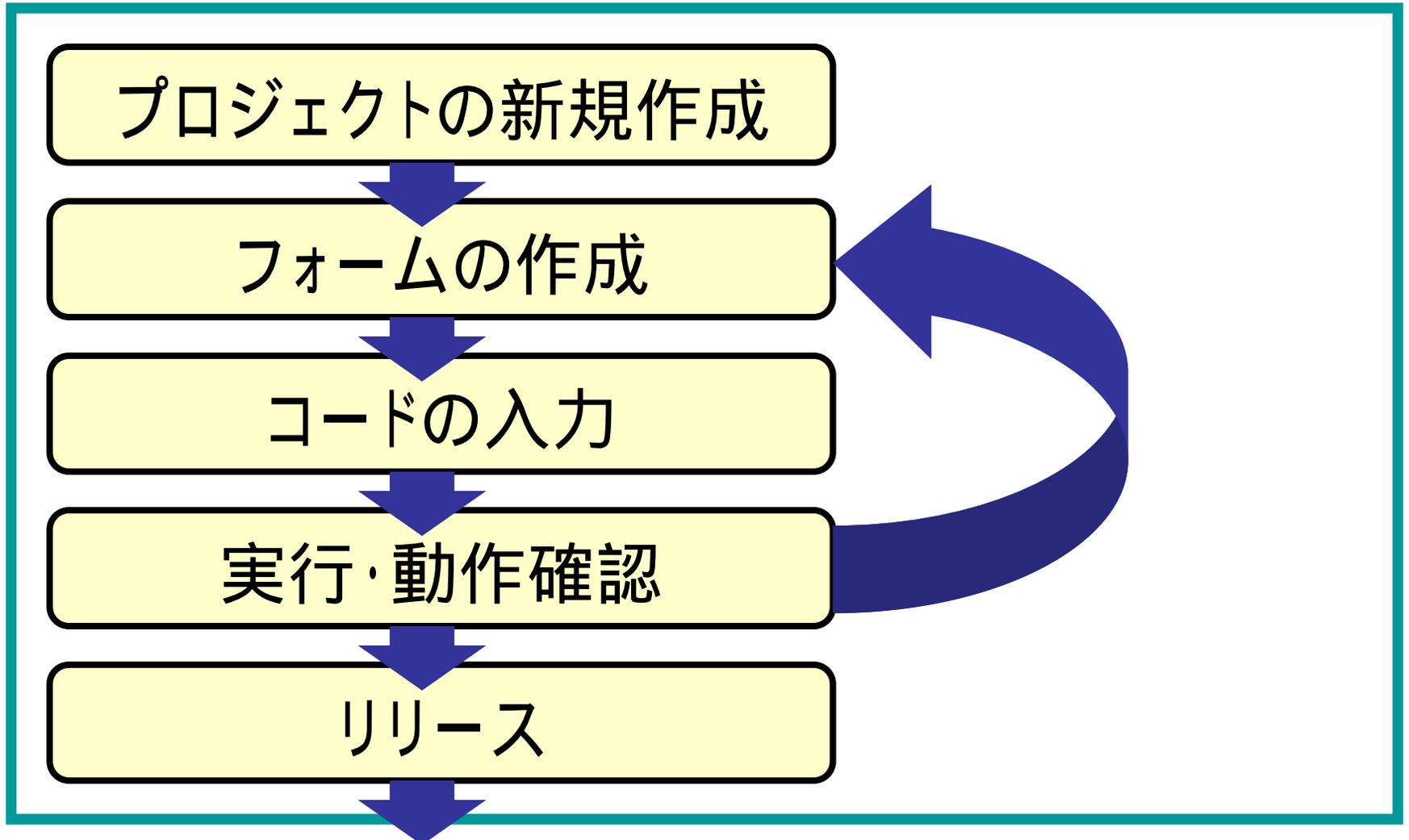
# オブジェクト指向プログラミング

コントロールなどのオブジェクトを中心に、その動作や性質を記述するプログラミングの手法

- オブジェクトの動作を与える **メソッド**
  - オブジェクト.**メソッド**()  
**Me.Dispose()**
- オブジェクトの性質を決める **プロパティ**
  - オブジェクト.**プロパティ**  
**Label1.Text**

プロパティの値は変更したり参照したりできる

# プログラミングの流れ



# 今回の予定

- さまざまなコントロールを使ってみよう！
  - Label コントロール (ラベル)
  - Button コントロール (ボタン)
  - TextBox コントロール (テキストボックス)
- 到達目標
  - テキストボックスによる文字列の入力ができる

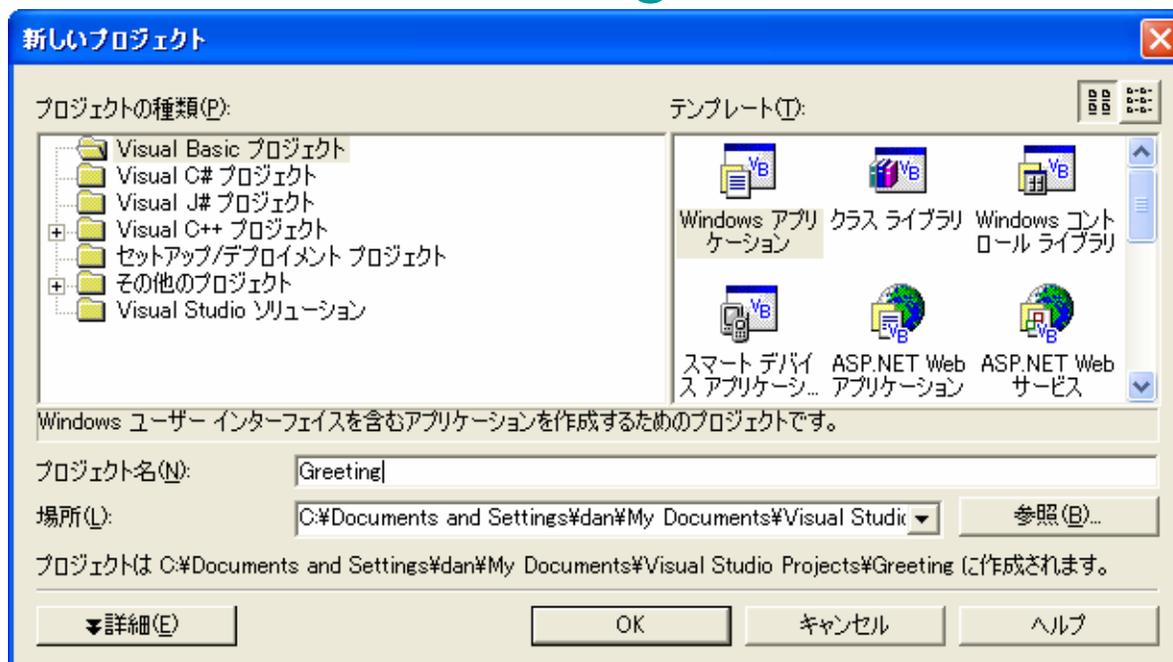
# 例題 3

テキストボックスから入力された文字列を使う

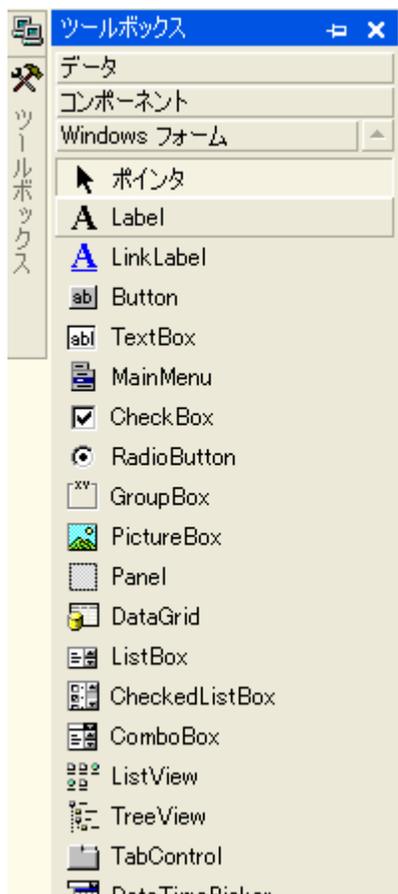
- 手順：
  - プロジェクトの新規作成 (**Greeting**)
  - 画面レイアウト (ラベルとボタンなどの配置)
  - イベントハンドラの記述
  - 実行および動作確認

# 新しいプロジェクト

- プロジェクトの種類「Visual Basicプロジェクト」
- テンプレート「Windowsアプリケーション」
- プロジェクト名「Greeting」



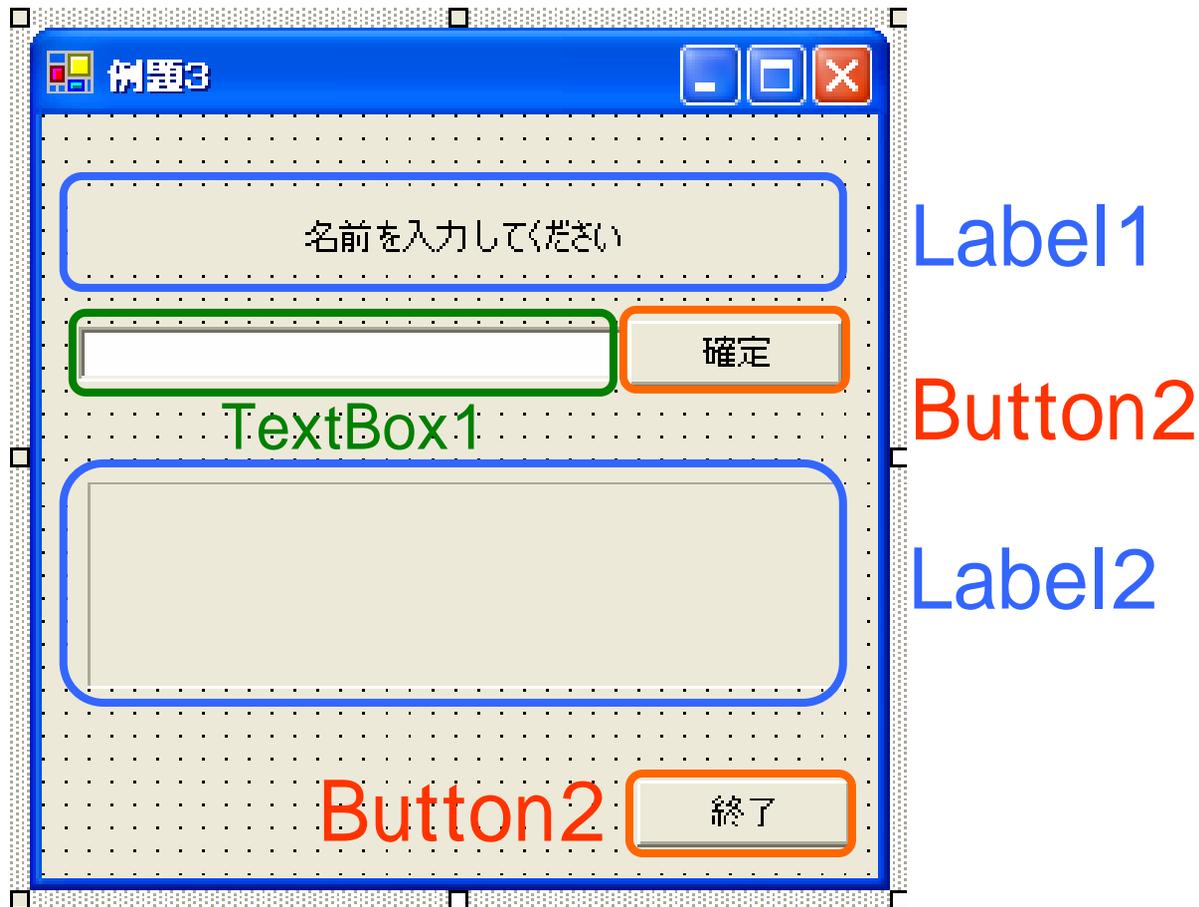
# 画面の構成要素



- ツールボックスの中から必要なコントロールを選択し、フォームに貼り付ける。
  - コントロールをダブルクリック
  - コントロールをフォーム上にドラッグ & ドロップ

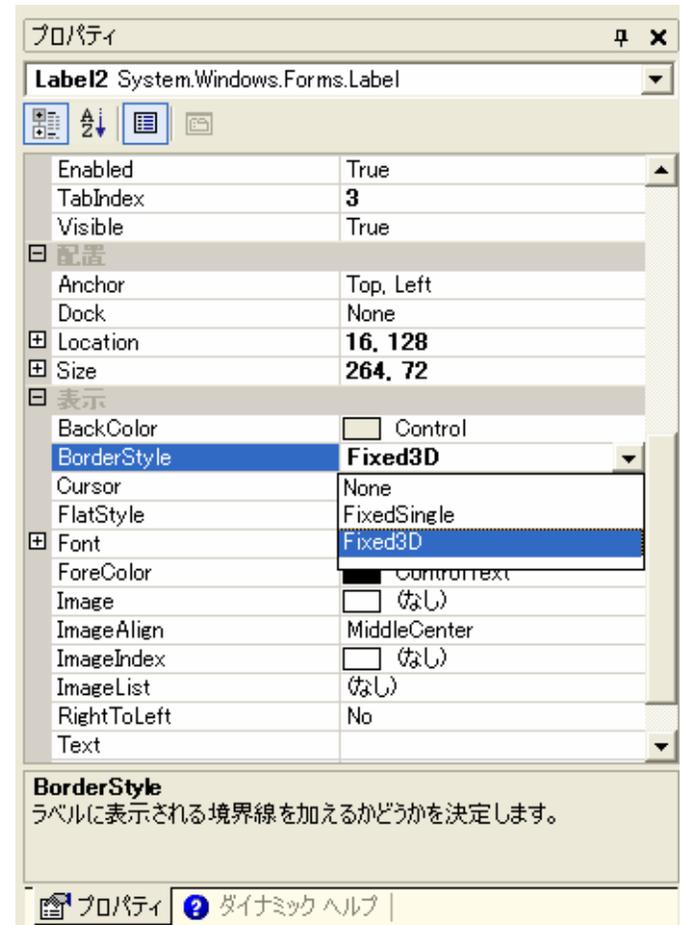
# フォームの構成要素

- ラベル、ボタンおよびテキストボックスを配置



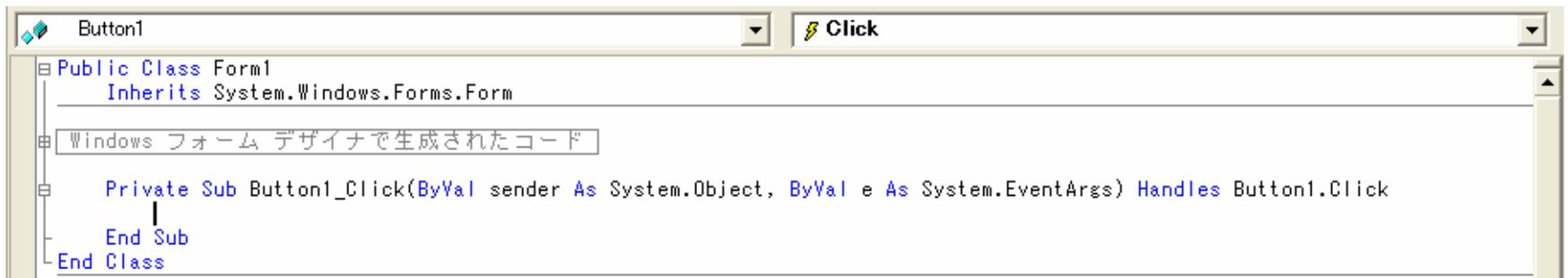
# ラベルのプロパティ

- Label2 の BorderStyle プロパティを「Fixed3D」に設定する



# コードの記述

- フォーム上のボタンをダブルクリックすると、プログラム実行中にボタンがクリックされたときの動作を VB.NET の文法で記述できる



The screenshot shows a Visual Studio code editor window. The title bar indicates the file is 'Button1' and the event is 'Click'. The code is as follows:

```
Public Class Form1
    Inherits System.Windows.Forms.Form

    Windows フォーム デザイナで生成されたコード

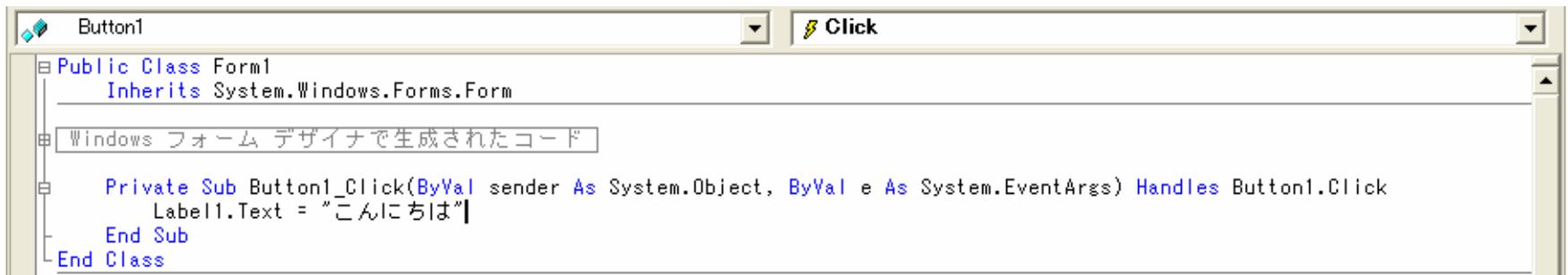
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        |
    End Sub
End Class
```

**Label12.Text = “こんにちは、 ” & TextBox1.Text & “さん ! ”**

# ラベルの文字列を書き換える

- ラベル (**Label1**) の Text プロパティに文字列を与えるコード

**Label1.Text = "こんにちは"**



```
Button1 Click
Public Class Form1
    Inherits System.Windows.Forms.Form
    Windows フォーム デザイナで生成されたコード
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Label1.Text = "こんにちは"
    End Sub
End Class
```

**Label1.Text** と記述して **Label1** の **Text** プロパティを表す

# 文字列の連結

& 演算子を使って、文字列を連結できる

【構文】 文字列 & 文字列

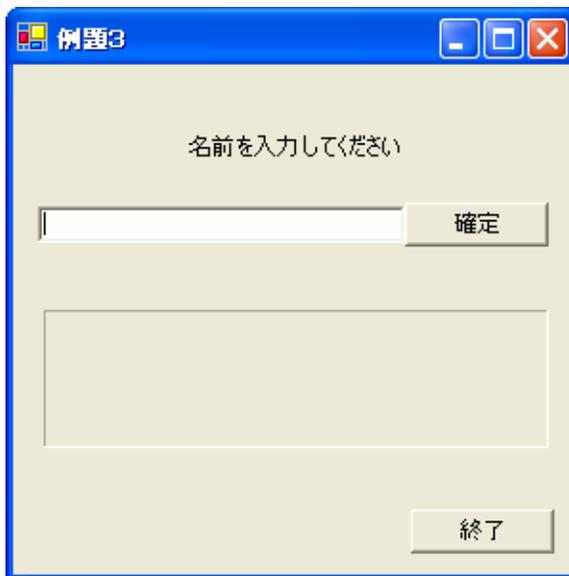
- 例えば “次長” & “課長” は “次長課長” となる
- & 演算子で3個以上の文字列も連結可能
- “12” & “34” は？ “46” か “1234” か？

`Label12.Text = “こんにちは、” & TextBox1.Text & “さん！”`

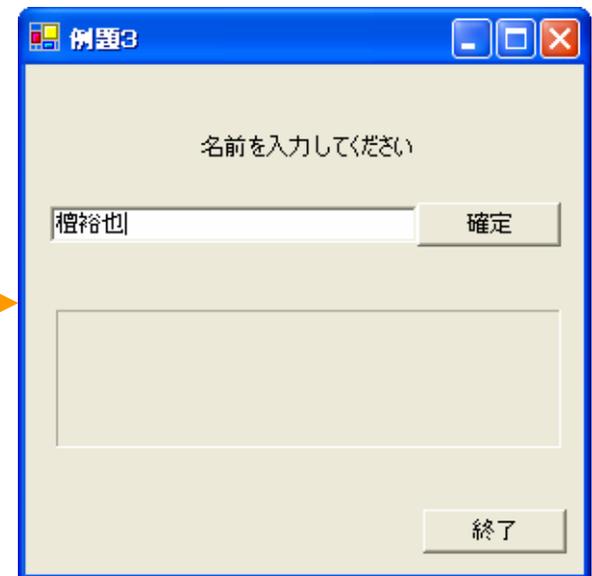
# 実行 (デバッグ)



- [デバッグ] [開始]でプログラムを実行する
- ツールバーの**開始ボタン**をクリックしてもよい



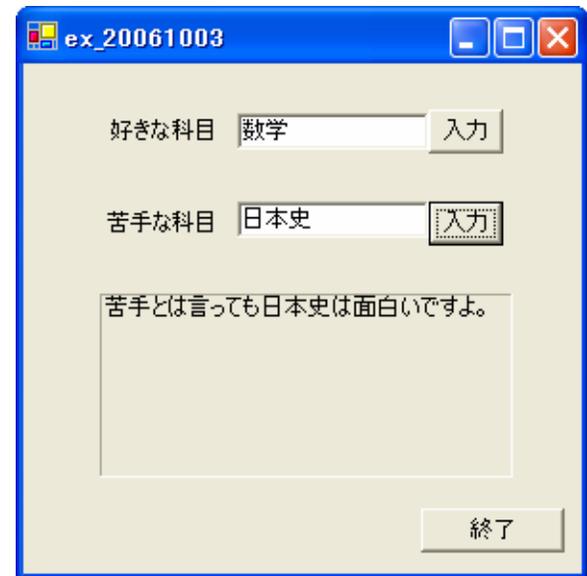
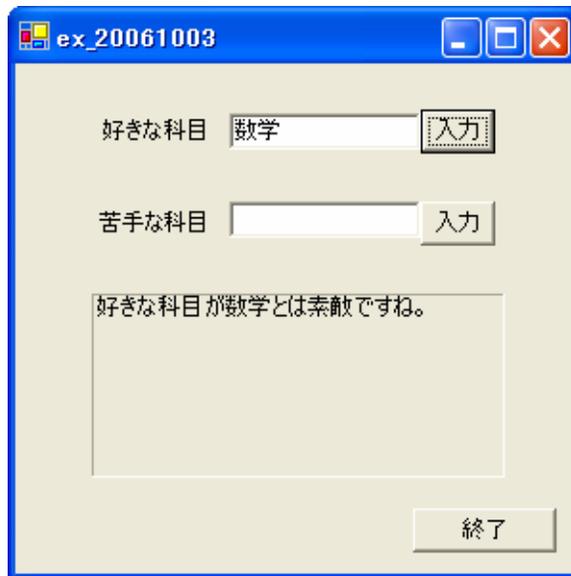
文字列を入力



# 課題

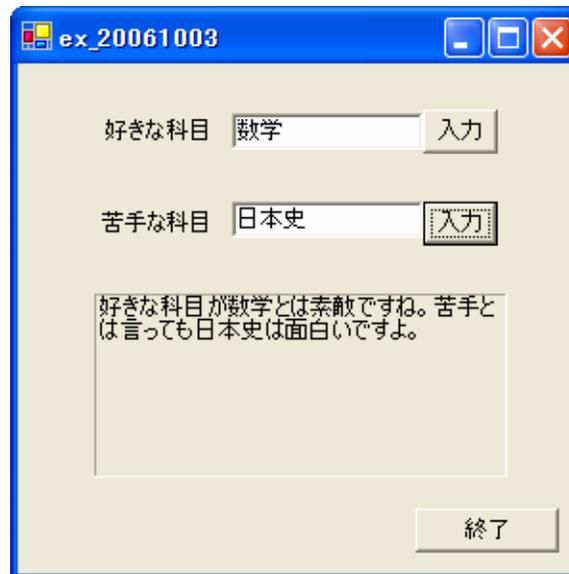
## 対話プログラム (ex\_20061003)

- 好きな科目を入力させ メッセージを表示する
- 苦手な科目を入力させ メッセージを表示する



# さらに...

- 前の文章に続けて次の文章が表示されるにはどのようにすればよいか考えてみよ。
- また、メッセージを消去するボタンを追加せよ。



# まとめ

- TextBox コントロールを利用し、キーボードから入力された文字列を処理するプログラムを作成した
  - TextBox.Text プロパティの利用
  - 文字列の連結演算子 &

# 次回予定

- 条件による場合分け（条件分岐）
  - 比較演算子
  - 数値と文字列