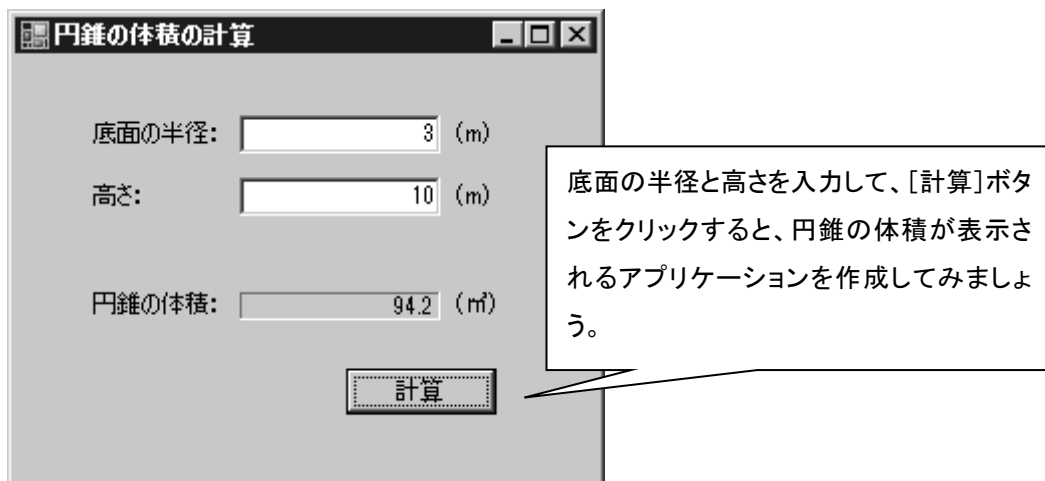


## 練習1. 変数の利用

円錐の体積を計算するアプリケーションを作成します。



### ステップ1. プロジェクトの作成

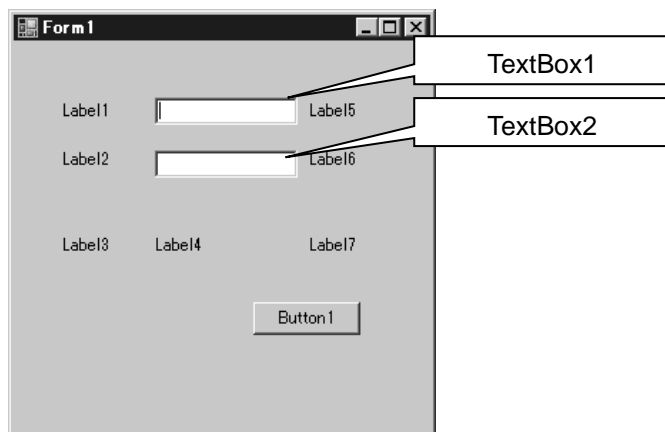
1. Visual Studio 2012 を起動します。
2. 以下の情報をもとに、スタートページまたは、[ファイル]-[新規作成]-[プロジェクト]から新しいプロジェクトを作成します。

プロジェクトの言語	VisualBasic
テンプレート	Windows フォームアプリケーション
プロジェクト名	MyCalc

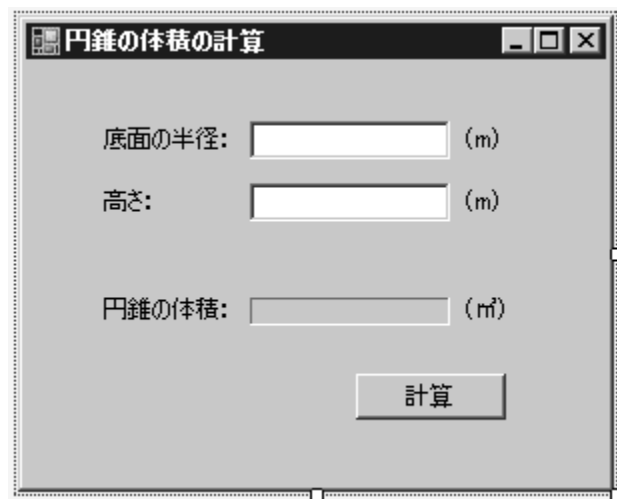
## ステップ2. ユーザーインターフェースの作成

1. 以下の画面イメージを参考に、カテゴリ[コモンコントロール]のコントロールを配置します。

- Label をフォームに7つ配置します。
- TextBox をフォームに2つ配置します。
- Button をフォームに1つ配置します。



2. プロパティウィンドウで、各コントロールのプロパティを変更して、以下のようなユーザーインターフェイスに変更します。



## 第2章 演習

コントロール名	プロパティ名	値
TextBox1	(Name)	RadiusTextBox
	TextAlign	Right
TextBox2	(Name)	HeightTextBox
	TextAlign	Right
Label1	Text	底面の半径:
Label2	Text	高さ:
Label3	Text	円錐の体積:
Label4	(Name)	VolumeLabel
	Text	(空白)
	AutoSize	False
	BorderStyle	Fixed3D
	TextAlign	MiddleRight
Label5	Text	(m)
Label6	Text	(m)
Label7	Text	(m <sup>3</sup> )
Button1	(Name)	CalcButton
	Text	計算
Form1	Text	円錐の体積の計算

### ステップ3. プログラムコードの記述

1. CalcButton ボタンの CalcButton\_Click イベントハンドラーを表示し、以下のコードを、ヒントを参考に完成させます。

● コード

①	
②	
	<code>Dim volume As Single = 0</code>
③	
	<code>height = HeightTextBox.Text</code>
④	
⑤	

## ● ヒント

①以下の変数を定義します。

名前	型	初期値
radius	Single	0

②以下の変数を定義します。

名前	型	初期値
height	Single	0

③変数 radius に RadiusTextBox.Text の値を代入します。

④変数 volume に円錐の体積を計算して代入します。計算式は以下です。

$$[\text{底面の半径}]^2 \times 3.14 \times [\text{高さ}] \div 3$$

⑤VolumeLabel.Text に変数 volume の値を代入します。

2. アプリケーションを実行します。
3. 底面の半径と高さを入力して計算ボタンをクリックします。
4. 円錐の体積が計算され、正しい結果が出力されることを確認します。
5. アプリケーションの実行を終了します。

## 練習2. 定数の利用

練習1で作成したアプリケーションを、定数を利用した処理に修正します。

### ステップ1. プログラムコードの変更とアプリケーションのテスト

1. CalcButton ボタンの CalcButton\_Click イベントハンドラーを表示し、以下のコードを、ヒントを参考に完成させます。

- コード

```
Dim radius As Single = 0
Dim height As Single = 0
Dim volume As Single = 0
```

①

②

```
radius = RadiusTextBox.Text
height = HeightTextBox.Text
```

③

```
VolumeLabel.Text = volume
```

- ヒント

①以下の定数を定義します。

名前	型	初期値
pai	Single	3.14

②以下の定数を定義します。

名前	型	初期値
coefficient	Integer	3

③体積の計算式内のリテラルを定数に置き換えます。

2. アプリケーションを実行します。

## 第2章 演習

3. 動作確認をおこない、アプリケーションの実行を終了します。

### **参考: 定数を利用する利点**

プログラムコードに直接データを記述するのではなく、定数を利用しているので、プログラムコードが理解しやすくなりました。また、後からデータが変更になったとしても、修正箇所は定数にデータを代入する部分だけで済みます。

## 練習3. (オプション) Option Strict と型変換

練習2で作成したアプリケーションのプログラムコードを変更し、Option Strict ステートメントの設定とデータ型の変換を確認します。

### ステップ1. Option ステートメントの挿入

1. MyCalcプロジェクトのForm1.vb ファイルをコードエディタで開き、以下のコードを、ヒントを参考に完成させます。

● コード

```
Option   
  
Public Class Form1  
  
    Private Sub CalcButton_Click(sender As Object, e As EventArgs  
    ) Handles CalcButton.Click  
        Dim radius As Single = 0
```

● ヒント

①ファイルレベルで暗黙的な型変換をおこなわない設定を記述します。

2. アプリケーションを実行します。

Q デバッグ実行をおこなうことができますか？それはなぜですか？

### ステップ2. データ型の明示的な変換

1. エラーが消えるように型変換を実装します。

※自由にコーディングしてください解答例は次のページに載せてあります

● コード

```
Private Sub CalcButton_Click(sender As Object, e As EventArgs
    ) Handles CalcButton.Click
    Dim radius As Single = 0
    Dim height As Single = 0
    Dim volume As Single = 0

    Const pai As Single = 3.14
    Const coefficient As Integer = 3

    radius = CSng(RadiusTextBox.Text)
    height = CSng(HeightTextBox.Text)

    volume = radius * radius * pai * height / coefficient

    VolumeLabel.Text = volume.ToString("#.#")
End Sub
```

2. アプリケーションを実行して、動作を確認します。

**Q デバッグ実行できますか？**

**参考: 数値書式指定文字列**

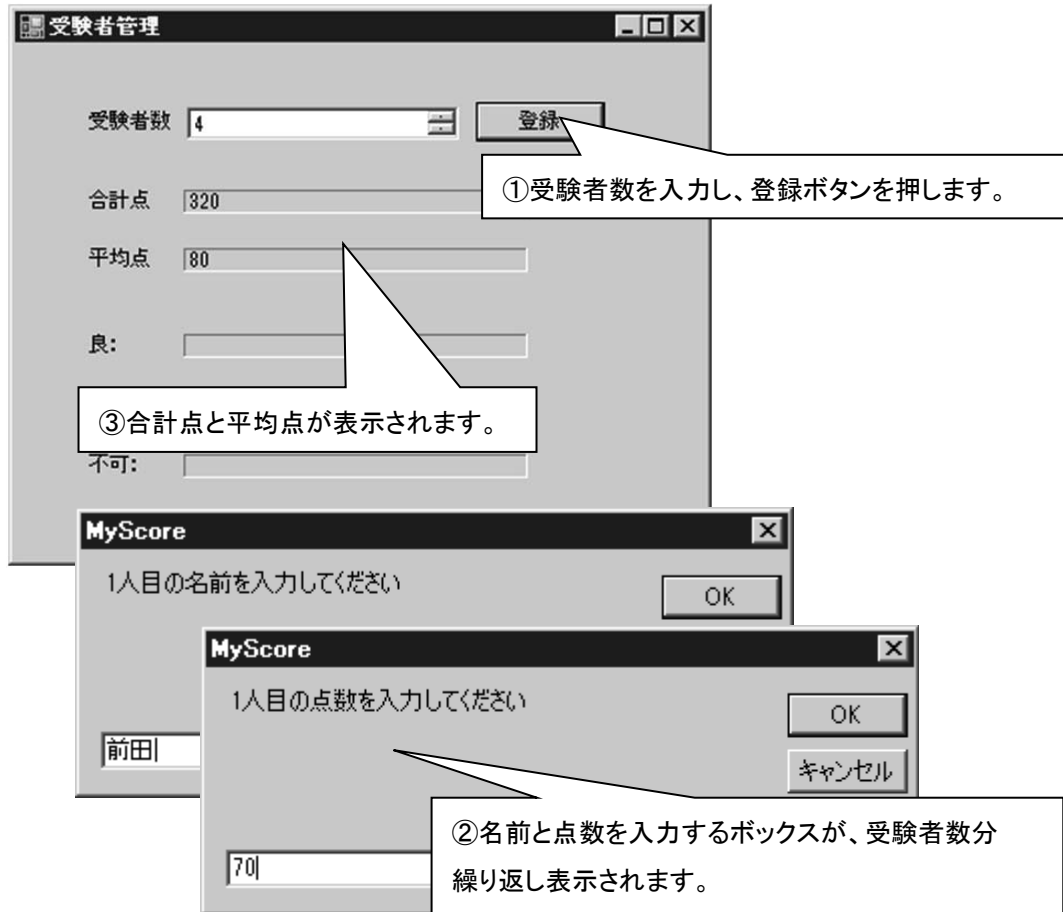
ToString メソッドは()内に、String 型へ型変換する際の書式の指定が可能です。この書式指定の記述のことを数値書式指定文字列と呼び、標準数値書式指定文字列とカスタム数値書式指定文字列の2種類が存在します。例えば#はカスタム数値書式指定文字列の1種で、桁の指定をおこないません。その他のパターンはヘルプを参照してください。

3. アプリケーションの実行を終了します。



## 練習4. 反復構造および配列の利用

For...Next ステートメントおよび配列を利用して平均点を計算するアプリケーションを作成します。



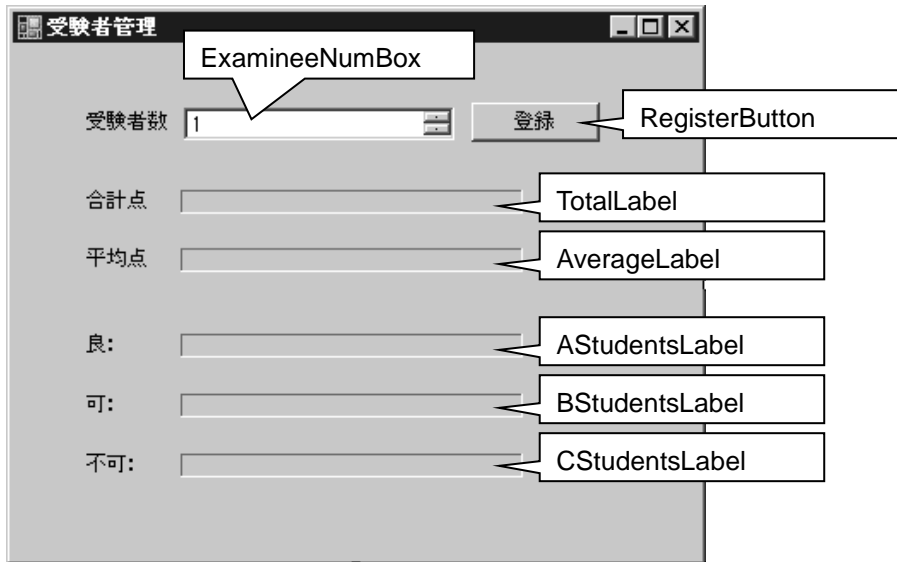
### ステップ1. ソリューションを開く

1. Visual Studio で以下のソリューションファイルを開きます。

演習用データ¥Starter¥第2章¥練習4¥MyScore¥MyScore.sln

## ステップ2. ユーザーインターフェースの確認

1. MainForm に配置されているコントロールを確認します。



### 参考: NumericUpDown コントロール

NumericUpDown コントロールを利用すると、アプリケーション実行時に、直接数値を入力することも、右側の上下ボタンをクリックして数値を増減させることもできます。

## ステップ3. プログラムコードの記述

1. RegisterButton ボタンの RegisterButton\_Click イベントハンドラーを表示し、以下のコードを、ヒントを参考に完成させます。

### ● コード

```

Dim numberStudent As Integer = ExamineeNumBox.Value
①
②
Dim totalScore As Integer = 0
③
name(index) = InputBox(index + 1 & "人目の名前を入力してください")
score(index) = InputBox(index + 1 & "人目の点数を入力してください")
④
Next
TotalLabel.Text = totalScore

```