平成29年度

プログラミング研修講座

Visual Basic

```
Public Class Form1

    Private Property Tn As Date

    Private Sub Form1_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load
        Timer1.Interval = 500
        Timer1.Enabled = True
    End Sub

    Private Sub Timer1_Tick(sender As System.Object, e As System.EventArgs) Handles Timer1.Tick
        Tn = Now
        Label1.Text = Tn.Hour.ToString + "時" + Tn.Minute.ToString + "分" + Tn.Second.ToString + "秒"
    End Sub

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

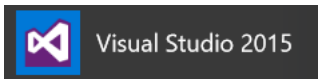
岩手県立総合教育センター

第3章　Visual Studio Community 2015 の基本操作

1　起動と画面構成

（1）Visual Studio 2015　の起動

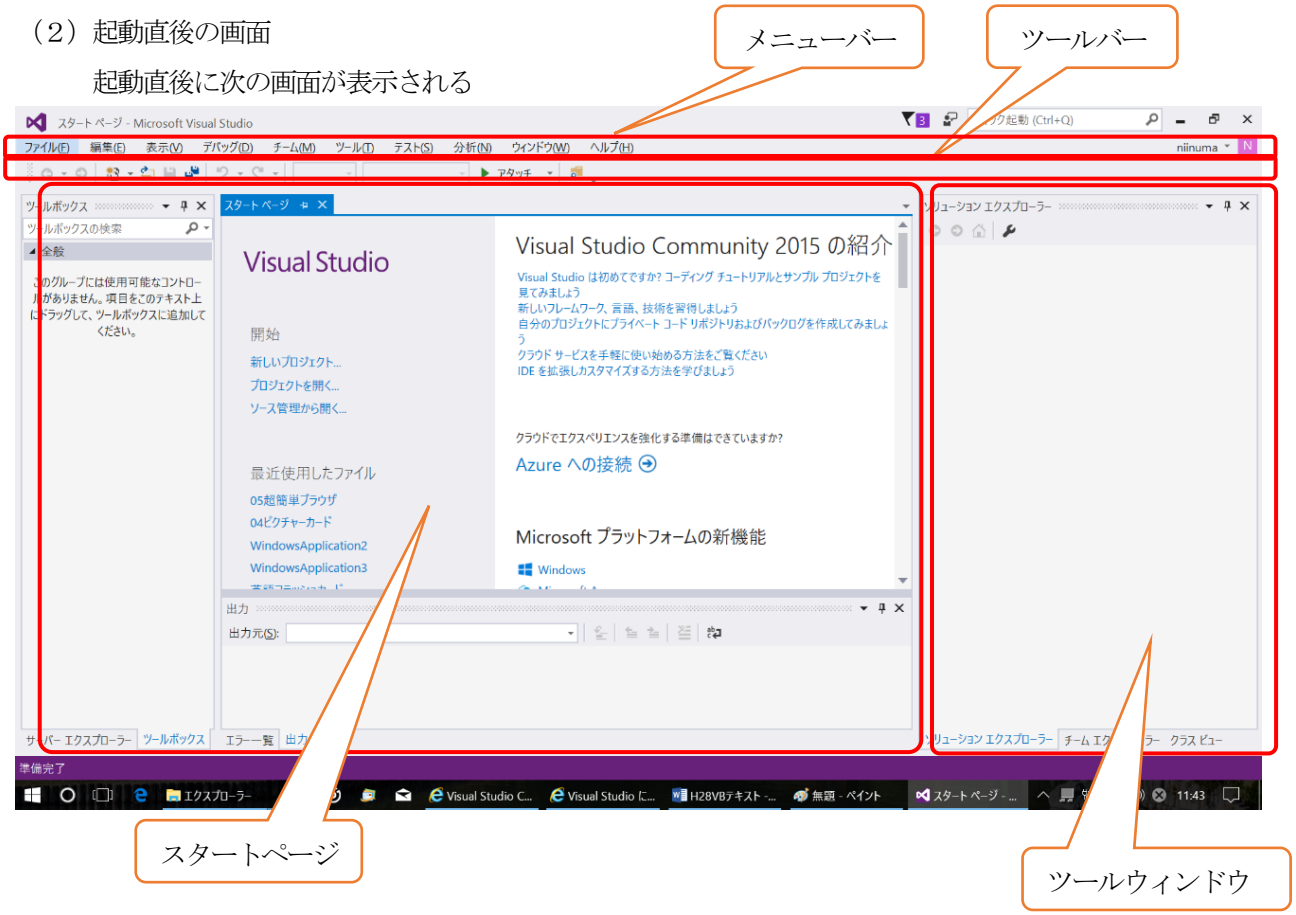
［スタート］ボタン ⇒ ［▶すべてのプログラム］ ⇒ ［Visual Studio 2015］をクリックし，



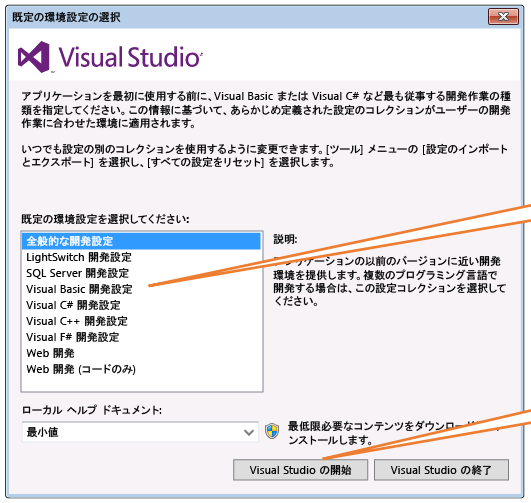
を選択します。

（2）起動直後の画面

起動直後に次の画面が表示される



※はじめて使用する場合は「既定の環境設定の選択」画面が表示される



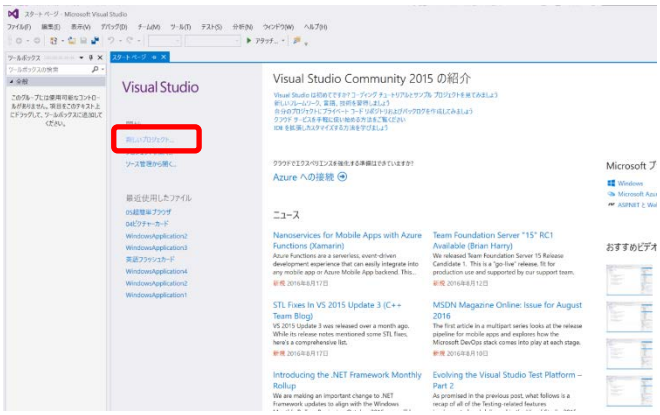
［Visual Basic 開発設定］を選択

［Visual Studio の開始］をクリック

2　プログラムファイルの作成

Visual Studio Community 2015 では，作成するプログラムファイルを**プロジェクト**と呼ばれる単位で管理します。プログラムを作成するためには，まずプロジェクトを作成する必要があります。プロジェクトを作成することで，プログラムコードを保存するためのファイルやコンパイルに関する情報を保存するファイルなどが自動的に生成されます。複数のプロジェクトをまとめたものを**ソリューション**といいます。

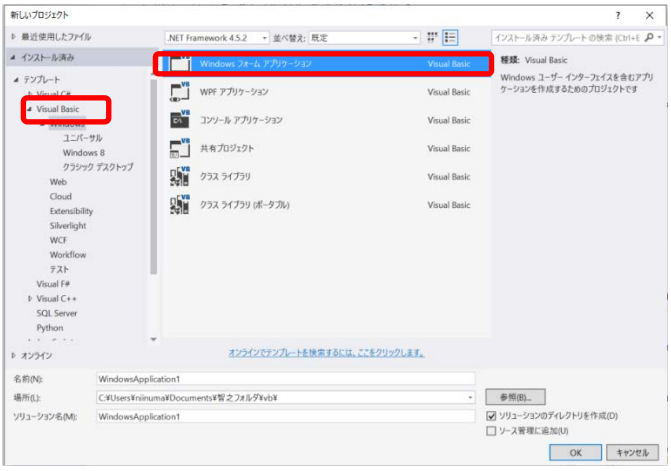
（1）新しいプロジェクトの作成



①　スタートページの［新しいプロジェクト］をクリックする

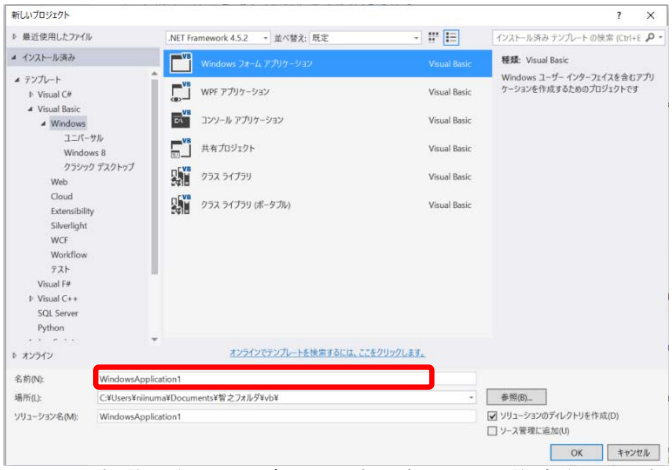
＜別法＞

メニューバーの［ファイル］をクリックし［新しいプロジェクト］をクリックする



②　「新しいプロジェクト」ダイアログボックスが表示される

インストールされたテンプレートの［Visual Basic］を選択し，［Windows フォーム　アプリケーション］を選択する



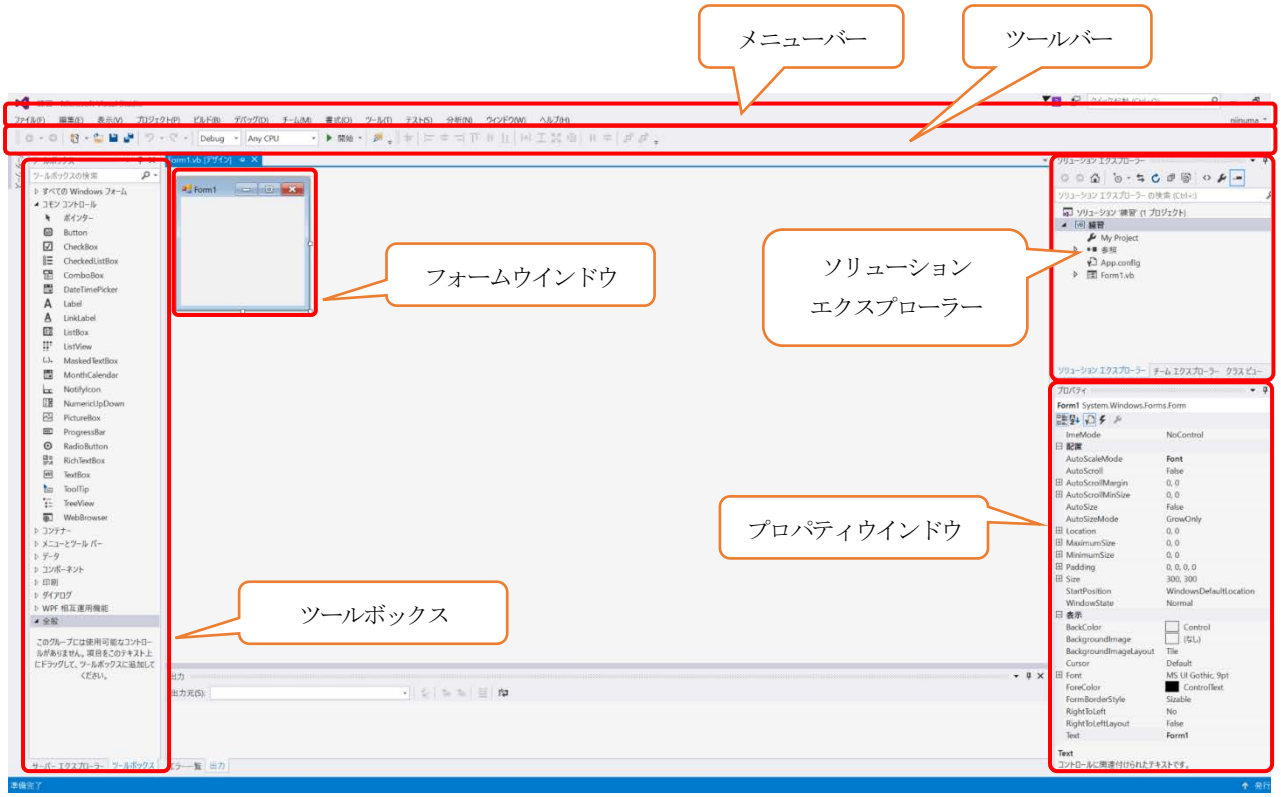
③　名前の入力欄にプロジェクト名を入力し［OK］ボタンをクリックする

【設定】練習

以上の操作を行うと，新しいプロジェクトが作成され，プログラムの作成画面が表示されます。

３　IDE（統合開発環境）画面の構成

プログラムの開発は，この画面で「Windows アプリケーションプロジェクト用のプロジェクトの作成」，「コントロールの配置」，「コントロールのプロパティの設定」，「プログラムコードの編集・実行・デバッグ」を行います。

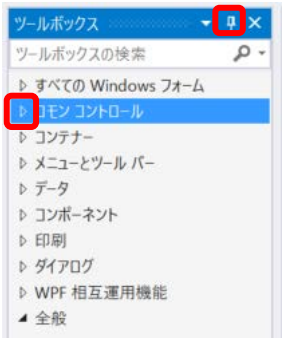


§　用語の確認

用　　語	説　　明
プロジェクト	プログラムに必要なファイルを管理する単位
ソリューション	大規模なプログラムを作成する場合に，複数のプロジェクトを管理する単位
ツールボックス	あらかじめ用意されている部品（コントロール）をここからドラッグして使います
フォームウインドウ	作成するアプリケーションの基本画面です
ソリューションエクスプローラー	プログラム作成に必要なファイルの情報をエクスプローラー風に表示・管理します
プロパティウインドウ	画面に貼り付けた部品（コントロール）の値を設定・変更することができます

4　ツールボックス内のコントロールとその機能

●Windows フォームデザイナーを表示しているときのツールボックス

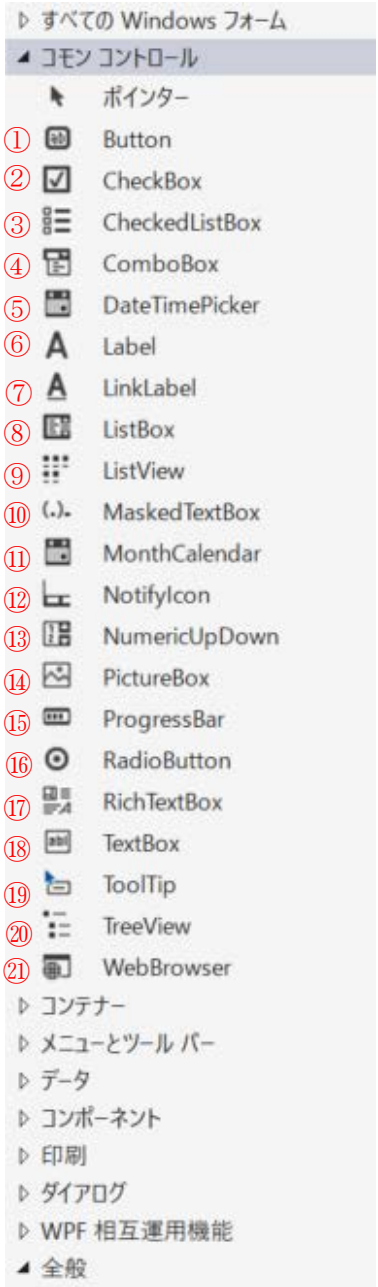


各タブの ▷ をクリックするとグループ分けされたコントロールが展開されます。◢をクリックすると折りたたむことができます。

ピンがヨコになっていると時は使用後ツールボックスが最小化されます

クリックしタテにするとツールボックスは固定されます

▼「**コモン コントロール**」：　主にフォーム上に配置するコントロールがグループ化されている



① クリックしイベントを動作させるボタンを配置する
② チェックボックスを配置する
③ 各項目のヨコにチェックボックスの付いた一覧を配置する
④ プルダウンコンボボックスを配置する
⑤ 日付や時刻を表示する
⑥ ユーザが編集できないテキストを表示する
⑦ Web スタイルのリンクを追加する
⑧ 定義済み一覧から項目を選択できる
⑨ 項目の一覧をアイコンで表示する
⑩ 入力したデータが正しいかどうか判定する
⑪ 日付情報を表示および配置する
⑫ バックグラウンドで動作するプロセスのためにアイコンを表示する
⑬ 参照して選択できる数値の一覧を表示する
⑭ 画像ファイルを表示する
⑮ 処理の進行状況を表示する
⑯ ラジオボタンを配置する
⑰ テキストの入力，表示，操作できる
⑱ 複数行のテキストを入力，編集できる
⑲ カスタムツール，メニューを作成する
⑳ 展開や折りたたみが可能なノードの階層を表示する
㉑ Web ブラウザ機能をアプリケーションに追加する

5　「プロパティ」ウインドウの機能

フォームやコントロールの状態を表す値をプロパティといいます。プロパティの値を設定するためには、対象のコントロールを選択状態にして、プロパティウインドウで行います。プログラム上から設定をすることができます。

ここでは、「Form1」を選択した場合のプロパティウィンドウの内容について解説します。

「ウインドウスタイル」の項目

① タイトルバーの最小化，最大化，閉じるボタンを表示・非表示にする

② 最大化ボタンの表示・非表示にする

③ 最小化ボタンの表示・非表示にする

④ 不透明度の割合　※０％＝透明

⑤ タスクバーへの表示・非表示

「デザイン」の項目

⑥ コントロールの内容が分かるような名前をここで付ける

「配置」の項目

⑦ フォームの外形にサイズをここで設定する

例）300，300　＝　ヨコ，タテ

「表示」の項目

⑧ フォームの背景色を設定する

ボタンをクリックすると色選択のダイアログボックスが表示される

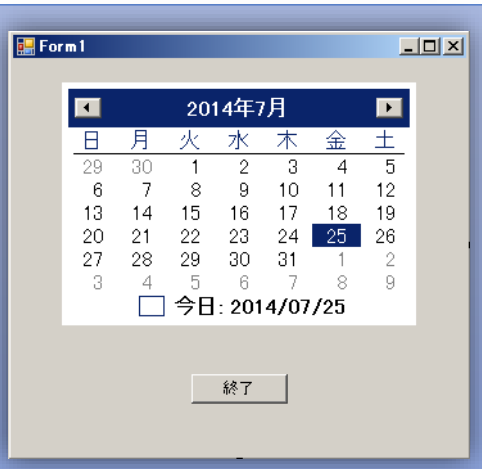
⑨ フォントを指定する

⑩ タイトルバーに指定した文字を表示する

機能を選択する

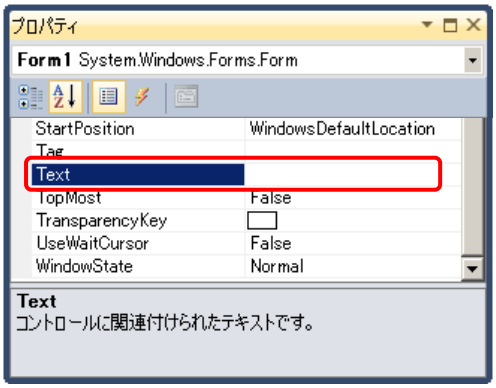
6　演習１　プログラムファイルの作成（簡単カレンダー）

「簡単カレンダー」を作成しながら，プログラミングの基本的な手順と操作について説明します。



ここでは，フォーム
［MonthCalendar］
貼り付けを行います。
これだけでは，プロ
せん。
そこで，［終了］ボ
プリケーションが終了
します。

STEP１　フォームの設定



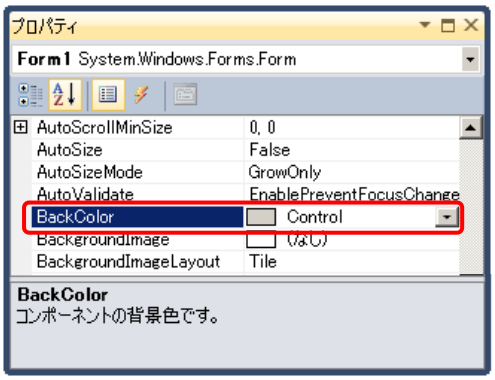
① プロパティウインドウの Text をクリックする

② 表示されている「Form1」を削除し，フォーム名を入力する

ここをクリックし，
「簡単カレンダー」と入力

【設定】　簡単カレンダー

STEP２　フォームの背景書を変更



③ BackColor をクリックする

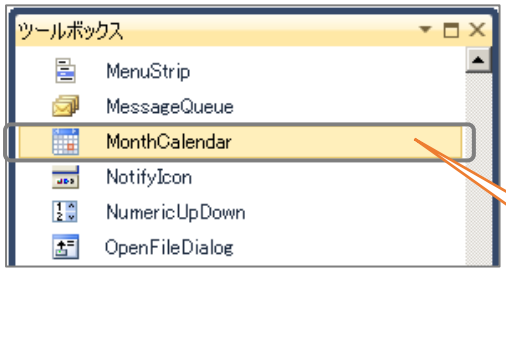
④ ▼をクリックし，［カスタム］タブをクリックする

⑤ 好きな色を選択する

▼をクリックします

［カスタム］タブをクリックし，
目的の色を選択します

STEP3　MonthCalender の配置

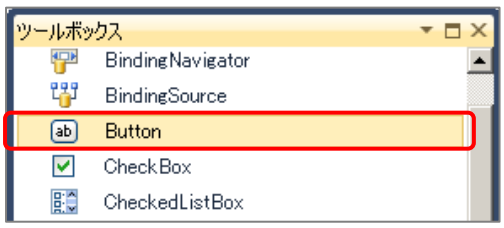


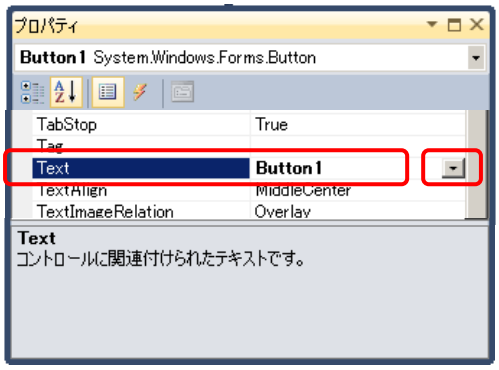
① ツールボックスのコモンコントロールから
［MonthCalendar］をクリックする

② フォーム上に配置したい場所でクリックする
＜別法＞ ツールボックス内の［MonthCalendar］
をダブルクリックしても配置できる

③ フォーム内でドラッグし位置を調整する

STEP4　Button の配置





① ツールボックスから，［Button］をクリックする

② フォーム上で配置したい場所でクリックする

③ 位置調整を行う場合は，フォーム内でドラッグし
カレンダーの下に移動する

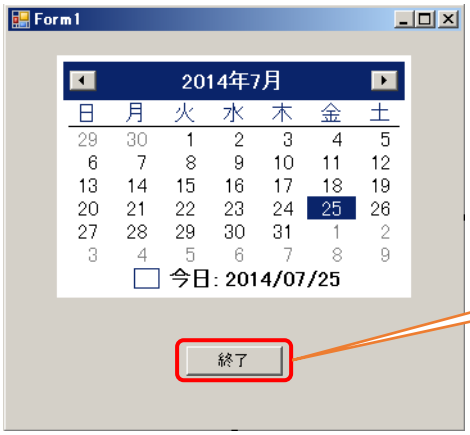
④ フォーム上に配置した Buton を選択している状態で，
「プロパティ」ウインドウ内の Text をクリック
する

⑤ 「Button1」を削除しボタンに名前をつける
【設定】 終了

⑥ Enter キーを押すと Button の名前が変わる

※ボタンの色を変更したい場合は，BackColor で変更する

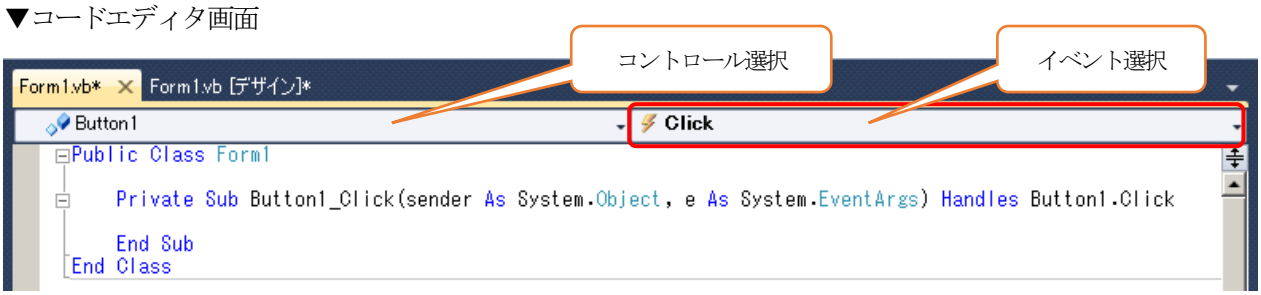
STEP5　プログラムコードの入力



①　作成した Button［終了］をダブルクリックします。

②　コードエディタが起動し，**イベントプロシージャ**（サブルーチン）が自動的に生成されます。

▼コードエディタ画面



③　「Private Sub」と「End Sub」の間に，半角英数で［End］と入力し，［Enter］キーを押します。

```
Private Sub Button1_Click(Sender As System.Object, e As System.EventArgs) Handles Button1.Click
End
End Sub
```

【Point】　Button1 をクリックしたときに実行するプログラムコード記述する。
※半角英数で入力　　※補助ウインドウが表示され選択できます。

★イベントプロシージャの構成

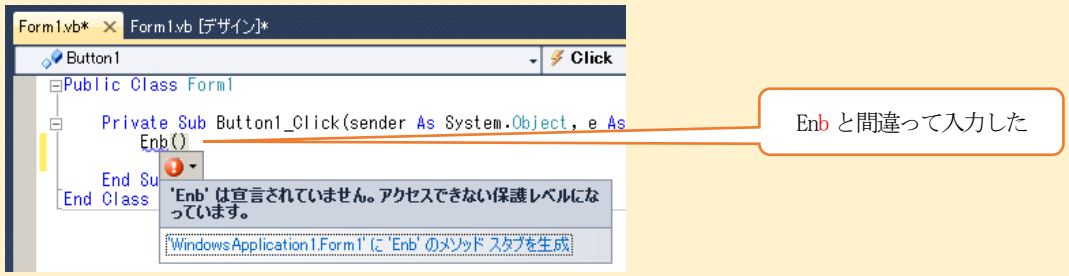
Private Sub **Button1_Click**(Sender As System.Object,e As System.EventArgs) **Handles Button1.Click**

＝ **オブジェクト名_イベント名**（　　**引　数**　　）**オブジェクト名．イベント**

● プログラムコードの入力ミスを修正する

プログラムコードを間違えて入力した場合は，間違っている箇所に，波線が表示されます。

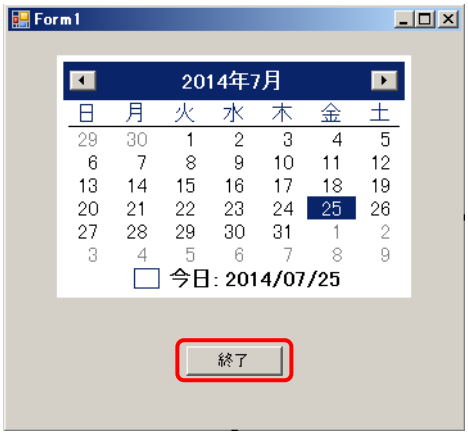
このとき，波線のでクリックすると，間違っている内容を確認することができます。



STEP6　プログラムの実行とデバッグ

入力したプログラムコードが正しいかを確認するために，**デバッグ**を行います。デバッグとは，作成したプログラムを実行し，誤り（**バグ**）を見付ける作業のことです。

①ツールバーにある［デバッグ開始］ボタンをクリックする

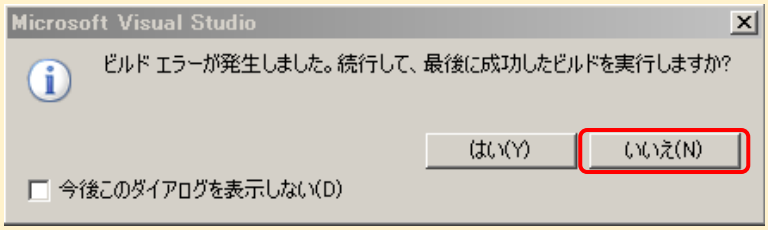


②作成したプログラムが実行され「MonthCalendar」が表示される

③［終了］ボタンをクリックし，表示された「MonthCalendar」のウインドウが消えると作成したプログラムが正しく実行されたことになる

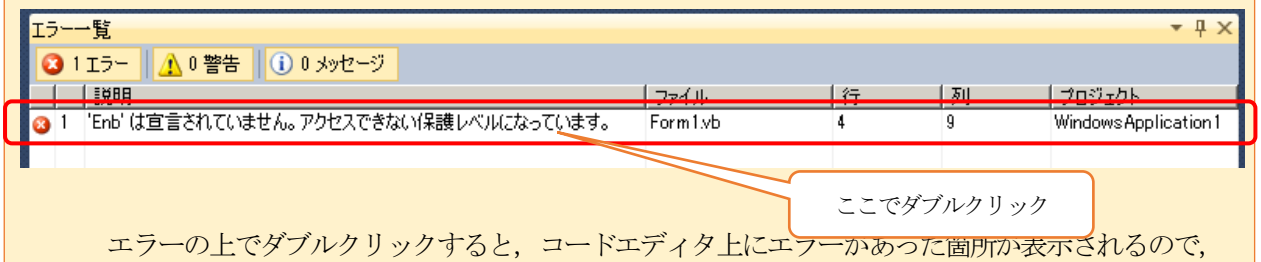
● デバッグにより，正しく実行されなかった場合

入力時にミスに気付かず，デバッグすると次のようなエラーメッセージが表示されます。

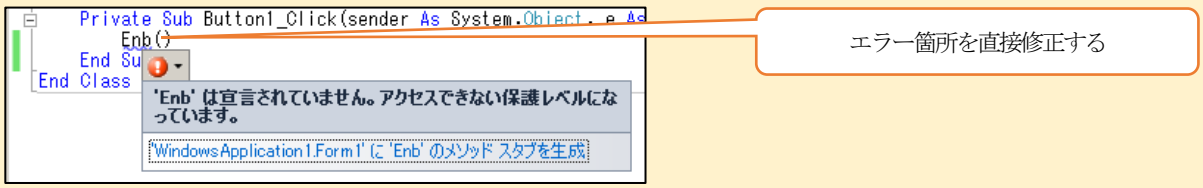


このような場合は，［いいえ］ボタンをクリックして，プログラムの実行を停止します。

次のようなエラー一覧ウインドウが表され，エラー件数，エラー説明，エラーがあった行が表示されます。



エラーの上でダブルクリックすると，コードエディタ上にエラーがあった箇所が表示されるので，修正し再度デバッグを開始してください。この作業をエラーが無くなるまで繰り返します。



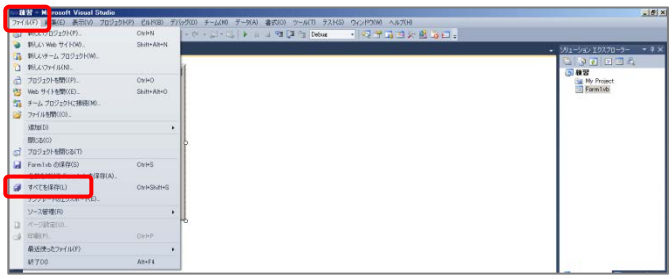
デバッグが完了した段階でのプログラムは，VisualBasic 上でしか実行することができません。

VisualBasic がインストールされていないコンピュータで，単独で動作させるためには，作成したプログラムを**コンパイル**して実行可能な形式のファイル（exe ファイルまたは dll）に変換し，各プログラムファイルとの関連づけ（**リンク**）をする必要があります。この作業を**ビルド**といいます。

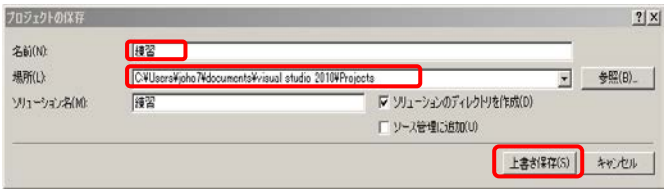
ビルドを行うことで，初めてアプリケーションソフトが完成します。

STEP7　プロジェクトの保存

プロジェクトを作成しただけでは，プロジェクト用ファイルの保存は行われないので，作成したプロジェクトは保存しておきます。



① ［ファイル］メニューをクリックし，［すべてを保存］を選択する

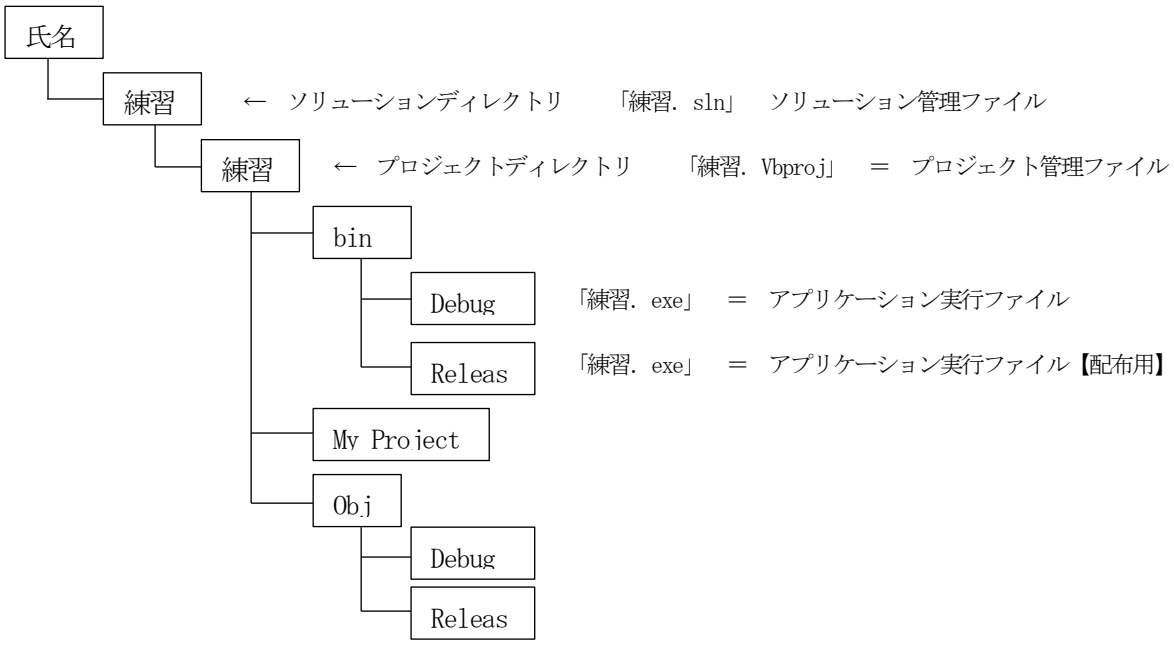


② 名前の入力欄に，プロジェクト名を入力する
※プロジェクト作成時に入力している場合は
そのプロジェクト名が表示される

③ 保存場所は，［参照］ボタンをクリックし任意の保存先を指定することができる
【設定】　D:¥氏名

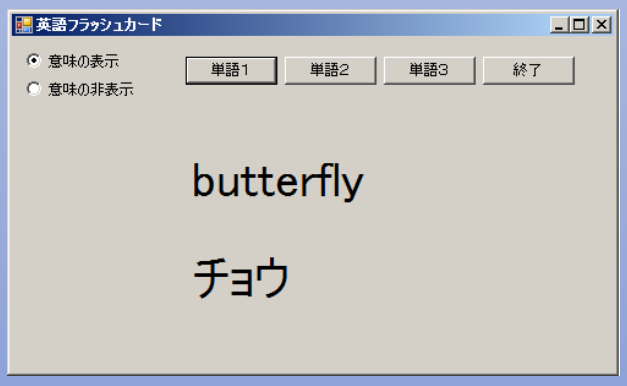
④ 入力内容を確認し，［上書き保存］ボタンをクリックする

【　保存先のフォルダ構成　】



（1）　演習３　英語フラッシュカード

文字と画像の表示（同じウインドウで表示）するアプリケーションを作成します。



ここでは，ラジオボタンを配置し文字の表示・非表示をさせます。

■　コントロールを配置とプロパティの設定

（1）　スタートページで［新しいプロジェクト］をクリックし，「新しいプロジェクト」のダイアログウインドウを表示させる

（2）　［Windows フォームアプリケーション］を選択しプロジェクト名を入力する

【設定】　英語フラッシュカード

（3）　フォームウインドウの［Form1.vb］をクリックする

（4）　「プロパティ」ウインドウで、「Form1」のサイズと名前を設定する

【設定】　　Text　英語フラッシュカード　　Size　500, 300

（5）　ツールボックスからラジオボタン［RadioButton］を２つ配置し，それぞれの名前を設定する

【設定】

RadioButton1	・・・　Text	意味の表示
RadioButton2	・・・　Text	意味の非表示

（6）　［Button］を選択して横に４つボタンを配置し，それぞれの名前を設定する

【設定】

Button1	・・・　Text	単語１
Button2	・・・　Text	単語２
Button3	・・・　Text	単語３
Button4	・・・　Text	終了

（7）　［Label］を選択して２つ配置し，それぞれのフォントサイズを設定する

Label1	・・・　Font	30
Label2	・・・　Font	30

■ プログラムの記述

（1）フォームをダブルクリックして次のコードを入力する

```
Label1.Text = ” ”
Label2.Text = ” ”
RadioButton1.Checked = True
```

（2）RadioButton1「意味の表示」をダブルクリックして次のコードを入力する

```
Label2.Visible = True
```

（3）RadioButton2「意味の非表示」をダブルクリックして次のコードを入力する

```
Label2.Visible = False
```

（4）Button1「単語１」をダブルクリックして次のコードを入力

```
Label1.Text = "butterfly"
Label2.Text = "チョウ"
```

（5）Button2「単語２」をダブルクリックして次のコードを入力する

```
Label1.Text = "beetle"
Label2.Text = "カブトムシ"
```

（6）Button3「単語３」をダブルクリックして次のコードを入力する

```
Label1.Text = "dragonfly"
Label2.Text = "トンボ"
```

（7）Button4「終了」をダブルクリックして次のコードを入力する

```
Me.Close()
```

■ デバッグ

コードを入力し終えたら、開始ボタン  を押して動作確認する

■ 保存

すべてを保存 を  クリックして、ソリューションを保存する

【作成されたプログラム】

```
Public Class Form1

    Private Sub Form1_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load
        Label1.Text = ""
        Label2.Text = ""
        RadioButton1.Checked = True
    End Sub

    Private Sub RadioButton1_CheckedChanged(sender As System.Object, e As System.EventArgs) Handles
RadioButton1.CheckedChanged
        Label2.Visible = True
    End Sub

    Private Sub RadioButton2_CheckedChanged(sender As System.Object, e As System.EventArgs) Handles
RadioButton2.CheckedChanged
        Label2.Visible = False
    End Sub

    Private Sub Button1_Click(sender As System.Object, e As System.EventArgs) Handles Button1.Click
        Label1.Text = "butterfly"
        Label2.Text = "チョウ"
    End Sub

    Private Sub Button2_Click(sender As System.Object, e As System.EventArgs) Handles Button2.Click
        Label1.Text = "beetle"
        Label2.Text = "カブトムシ"
    End Sub

    Private Sub Button3_Click(sender As System.Object, e As System.EventArgs) Handles Button3.Click
        Label1.Text = "dragonfly"
        Label2.Text = "トンボ"
    End Sub

    Private Sub Button4_Click(sender As System.Object, e As System.EventArgs) Handles Button4.Click
        Me.Close()
    End Sub
End Class
```

（3） 演習5 超簡単ブラウザ

自作のブラウザを作成します。

ここでは，URL を入力して「移動」ボタンをクリックすると Web ページを表示します。

当然、ハイパーリンクで他のページを表示しますし、ページを移動した場合にはその URL も表示します。ウインドウの大きさを変えることもできます。

■ コントロールを配置とプロパティの設定

（1） スタートページで［新しいプロジェクト］をクリックし，「新しいプロジェクト」のダイアログウインドウを表示させる

（2） ［Windows フォームアプリケーション］を選択しプロジェクト名を入力する

【設定】 Ｍｙブラウザ

（3） フォームウインドウの［Form１.vb］をクリックする

（4） 「プロパティ」ウインドウで、「Form1」のサイズと名前を設定する

【設定】 Size 530, 540

（5） ツールボックスで［Label］を１つ配置する

【設定】 Label1 Text ＵＲＬ

（6） ［TextBox］を選択して１つ配置する

【設定】size 345, 19

（7） ［Button］を選択して１つ配置する

【設定】 Button1 Text 表示する

（8） 「ツールボックス」で「すべての Windows フォーム」を選択して、一番下の［WebBrowser］ WebBrowser を選択して１つ配置し，次のように設定する

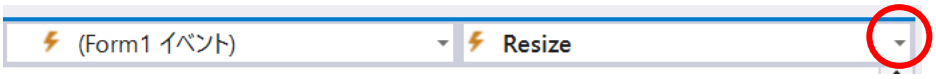
【設定】 Sise 490, 434

■ プログラムの記述

（1） フォームをダブルクリックして「Private Sub Form1_Load」の中に次のコードを記入する

```
TextBox1.Text = "http://www1.iwate-ed.jp/"
WebBrowser1.Navigate(TextBox1.Text)
```

（2） フォームのイベントから「⌄」をクリックして「Resize」を選択する



「Private Sub Form1_Resize」の中に次のコードを記入する

```
WebBrowser1.Width = Me.Width - 10
WebBrowser1.Height = Me.Height - 65
```

（3） Button1「表示する」をダブルクリックして次のコードを入力する

```
WebBrowser1.Navigate(TextBox1.Text)
```

（4） ウェブブラウザ(WebBrowser)をダブルクリックして「Private Sub WebBrowser1_Navigated」に次のコードを入力する

```
Me.Text = WebBrowser1.Document.Title.ToString() + " - MyBrowser"
Me.TextBox1.Text = WebBrowser1.Url.ToString()
```

■ デバッグ

コードを入力し終えたら、開始ボタン を押して動作確認する

■ 保存

すべてを保存 を クリックして、ソリューションを保存する

【作成されたプログラム】

```
Public Class Form1

    Private Sub Form1_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load
        TextBox1.Text = "http://www1.iwate-ed.jp/"
        WebBrowser1.Navigate(TextBox1.Text)
    End Sub

    Private Sub Form1_Resize(sender As Object, e As System.EventArgs) Handles Me.Resize
        WebBrowser1.Width = Me.Width - 10
        WebBrowser1.Height = Me.Height - 65
    End Sub

    Private Sub Button1_Click(sender As System.Object, e As System.EventArgs) Handles Button1.Click
        WebBrowser1.Navigate(TextBox1.Text)
    End Sub

    Private Sub WebBrowser1_DocumentCompleted(sender As System.Object, e As
System.Windows.Forms.WebBrowserDocumentCompletedEventArgs) Handles WebBrowser1.DocumentCompleted
        Me.Text = WebBrowser1.Document.Title.ToString() + "-mybrowse"
        Me.TextBox1.Text = WebBrowser1.Url.ToString()
    End Sub

End Class
```

2　変数と演算子

（1）　変数の基礎知識

変数はデータを入れる容器です。変数をプログラムの中で使うためには、最初に「変数の宣言」をします。変数が使える有効範囲をスコープといいます。

① 変数の宣言

Dim　：　普通の変数の宣言に用います。
プロシージャ内で宣言した場合には、そのプロシージャ内で実行している間だけ有効です（ローカル変数）。
※　プロシージャ（クラスコードのまとまり）

Public　：　アプリケーションのどこからでも使用可です。

Static　：　プロシージャ内で使用します。プロシージャが終了しても変数の値を保持します（静的変数（プログラム終了まで値を所持する変数））。

const　：　値を参照することだけが可能な変数を定義することができます。

② 変数の宣言方法

例：　**Dim 変数 As 型**

Public 変数 As 型

（2）変数名について

- 変数名は英数または_（アンダスコア）であること
- 漢字、ひらがなの変数名をつけることもできます（普通はつけませんが・・・）
- 変数名に VB2008 のコマンド名、オブジェクト名は使用できない。
- 英字の大文字と小文字は区別しない（C など他の言語は区別するのが普通です）

（3）代表的なデータ型について

型　名	内　容	サイズ	値の範囲
Boolean	論理型	2バイト	True（真）または False（偽）
Integer	整数型	4バイト	-2147483648 ～214748367 の整数
Single	単精度 浮動小数点型	4バイト	負の値 約-3.4×10(38 乗)～-1.4×10(-45 乗) 正の値 約 4.9×10(-45 乗)～1.8×10(38 乗)
Double	倍精度 浮動小数点型	8バイト	負の値 約-1.8×10(308 乗)～-4×10(-324 乗) 正の値 約 4.9×10(-324 乗)～1.8×10(308 乗)
String	文字列型	不定	最大 20 億個

（4）変数への値の代入について

変数「Hensu」に数値の「1」を代入する場合には、次のように記入します。

```
Dim Hensu As Integer
Hensu = 1
```

または、次のように記入することにより、変数の宣言と代入ができます。

```
Dim Hensu As Integer =1
```

（5）型変換

データの型を持っているため、型が異なると計算ができません。例えば、テキストボックスに入力された文

字を数値にするためには型変換を行います。

```
Dim Suuti As Single
Dim Mozi As String
        Suuti = Convert.ToSingle(Mozi)
```

逆に、数値を文字列に変換する場合には、

```
Mozi = Convert.ToString(Suuti)
```

（6）演算子

下表の上位が優先となります。優先順位を変える、または明確にしたい場合には括弧（）を用いてください。

〈算術演算子〉

^	べき乗
*	乗算
/	除算
¥	整数の除算
Mod	整数の除算の余り
+	加算
-	減算
&	文字列の連結（＋でも連結できます）

〈関係演算子〉

=	等しい	
<>	等しくない	not(A=B)
<	小さい、未満	
<=	以下（小さいか等しい）	
>	大きい	
>=	以上（大きいか等しい）	

（6）関数

Int(x)	その数を超えない最大の整数	Int(1.5)は1、Int(-1.5)は-2
Fix(x)	その数の整数部分	Int(1.5)は1、Int(-1.5)は-1
Rnd()	0～1までの乱数の発生	
Math.Round(x)	四捨五入	
Math.Abs(x)	絶対値	
Math.Sin(x)	サイン、xの単位はラジアン	

Sin45° の値をAに代入するためには次のように記入します。

```
A = Math.Sin(45/180*Math.PI)
```

Math.Cos(x)	コサイン、xの単位はラジアン
Math.Tan(x)	タンジェント、xの単位はラジアン
Math.Sqr(x)	平方根

（7） 配列

配列は、同じ性質を持った値を効率的に管理するためのデータ構造です。変数の集合体ともいえます。配

列の各要素に書き込みを行うためには、インデックス（番号、添字）で指定します。インデックスが一つのものを一次元配列、二つのものを二次元配列といいます。配列も宣言を行ってから使います。

① 整数型の一次元配列宣言の例

```
Dim Hairetu(10) As Integer
```

② 文字列の２次元配列宣言の例

```
Dim Mozi(10,2) As String
```

Visual Basic 2010 の「こんなこともできる」

1　変数に全角日本語が使えます。

```
例　Dim 大根 As Single                    ’変数として「大根」を定義
    大根 = 100                            ’大根に 100 を代入
```

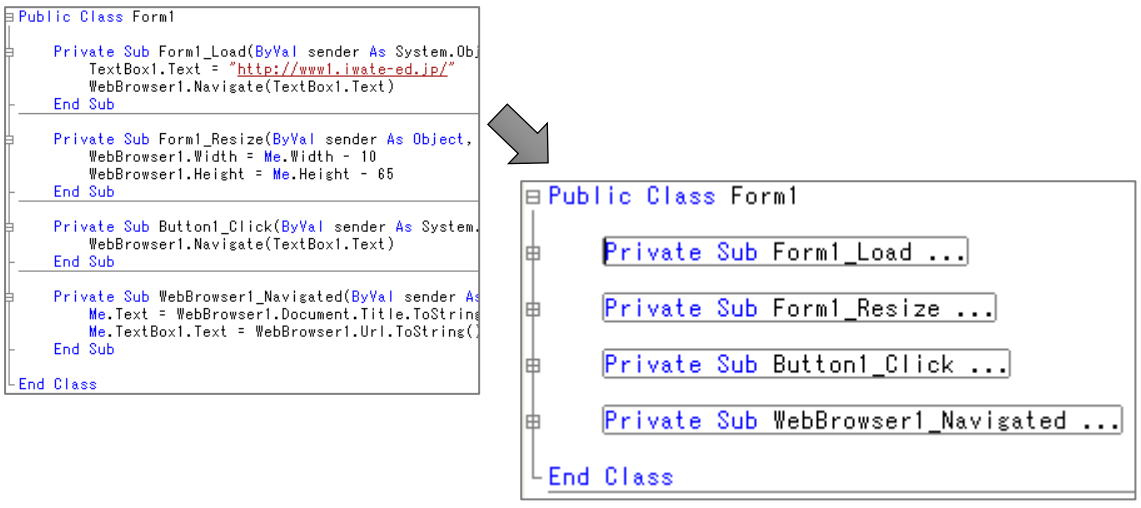
2　プロシージャ名に全角日本語が使えます。

クラス名にも使うことができます。

```
Private Function 代金(ByVal 値段 As Single) As Single
    Dim 消費税 As Single
    消費税= 0.05
    代金= 値段 * (1 + 消費税)
End Function
```

3　コードをたたんで表示させることができます。

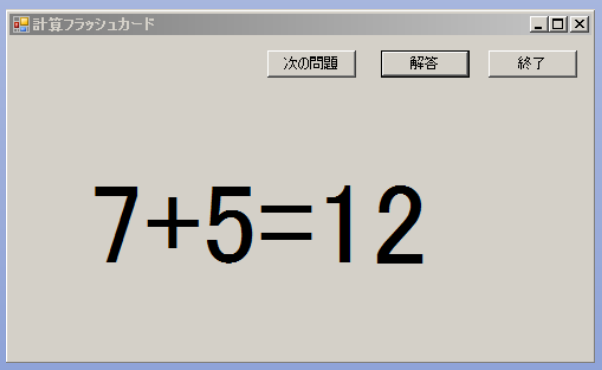
この機能を使うと、プロシージャ名だけを表示させることができます。



動作確認した部分をたたんでおいて、今書いているところだけを表示させると便利です。

（1）　演習6　計算フラッシュカード

ランダムに出てくる１桁の足し算をするプログラムを作成します。



ここでは，小学校１年生用のたし算の計算フラッシュカードを作成します。少し改変することにより「ひき算」「かけ算」「わり算」に変更することができます。

■　コントロールを配置とプロパティの設定

（1）　スタートページで［新しいプロジェクト］をクリックし，「新しいプロジェクト」のダイアログウインドウを表示させる

（2）　［Windows フォームアプリケーション］を選択しプロジェクト名を入力する

【設定】　計算フラッシュカード

（3）　「ソリューション　エクスプローラー」ウインドウの［Form1.vb］をクリックする

（4）　「プロパティ」ウインドウで、「Form1」のサイズと名前を設定する

【設定】　Text　計算フラッシュカード　　Size　700, 300

（5）　ツールボックスから［Button］を選択して３つボタンを配置し次のように設定する

【設定】

Button1　Text　次の問題

Button2　Text　解答

Button3　Text　終了

（6）　ツールボックスから［Label］を選択して配置し次のように設定する

【設定】　Font → size　72

■ プログラムの記述

```
Public Class Form1
    Public suuti1 As Single
    Public suuti2 As Single


    Private Sub Form1_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load
        Label1.Text = ""
    End Sub

    Private Sub Button1_Click(sender As System.Object, e As System.EventArgs) Handles Button1.Click
        suuti1 = Int(Rnd() * 9 + 1)
        suuti2 = Int(Rnd() * 9 + 1)
        Label1.Text = suuti1.ToString + "+" + suuti2.ToString + "="
    End Sub

    Private Sub Button2_Click(sender As System.Object, e As System.EventArgs) Handles Button2.Click
        Label1.Text = suuti1.ToString + "+" + suuti2.ToString + "=" + (suuti1 + suuti2).ToString
    End Sub

    Private Sub Button3_Click(sender As System.Object, e As System.EventArgs) Handles Button3.Click
        Me.Close()
    End Sub
End Class
```

※　数値を文字列に変換するには,

- 数値.ToString
- Convert.ToString(数値)
- Str(数値)

このように複数の方法があります。「.ToString」は.NETからの新しい表記です。

■ デバッグ

コードを入力し終えたら、開始ボタン  を押して動作確認する

■ 保存

すべてを保存 を  クリックして、ソリューションを保存する

【発展問題】

- 計算式の文字を大きくしましょう。
- 計算式の文字の色を変えてみましょう。
- ２桁の数の計算に変えてみましょう。
- 「かけ算」のカードを作ってみましょう。
- 「ひき算」「わり算」のカードを作ってみましょう。