

快速脱! 休日出勤

.NETアプリケーション障害解析の極意

デバッグ技法

第2回

ブレークポイントを利用した デバッグ - デバッグ工程 -

株式会社NTTデータ
飯山 教史
IIYAMA, Takashi

実行時にコードの 状態を調査する

今回はエラー処理をどのように書けばよいのかを指針レベルで説明した。

- ①仕様の把握
- ②正常系の記述
- ③エラー処理

と開発を進めて、いざ実行というときに、作ったプログラムが予想外の動きをしたら、皆さんはどうするだろうか？当然、その原因を突き止めるべく調査

を開始するだろう。しかし、コンパイルエラーならともかく入力値に対して仕様どおりの動きをしない場合は、実行しながらコードの状態を調査する必要がある。

この作業を助けてくれるのが「ブレークポイント」である。ブレークポイントを使えばコードを好きなタイミングでストップできる。そのとき、Visual Studio 2005が提供する機能を利用すればプログラムの状態を簡単に確認できるので、実装したロジックのミスが原因であれば「何らかの値が想定外の値になっている」程度なのですぐに修正できる。

今回はこのブレークポイントを中心に、プログラムの状態を調べる方法を説明する。

単に「設定する」と書いたが、どこに設定するかは注意が必要である。すでに間違ったパスに入った後にブレークポイントを設定しても意味がない。論より証拠なので、実際にコードで確認してみよう。

リスト1にサンプルとして苗字を指定するとフルネームを表示するプログラムを示す(図1)。このコードを実行してみると、苗字入力欄に空文字を入れても「不正な入力です」と表示されてしまう。仕様としては、13行目にある「苗字を入れてください」を表示したいので、これはバグである。この原因を調べるため、ブレークポイントを設定してみよう。

27行目に「不正な入力です」を表示するコードがあるので、ここより前にバグがあるはずである。12行目にメッセージの表示判定を行なうIf文があるので、この行にブレークポイントを設定してどのように判定しているかを確認してみる。

ブレークポイントは、Visual Studio 2005のコードエディタでブレークポイ

レベル >>> Level

1 2 3 4 5

言語 >>> Language

Visual Basic

ツール >>> Tool

Visual Studio 2005 Team Edition
for Software Testers

ブレークポイントの設定

ブレークポイントはプログラムの開発時にプログラム状態を確認するためにプログラムコード中に設定する。簡

リスト1：苗字を指定するとフルネームを表示するサンプルプログラム

```

1: Public Class Form1
2:
3:     ' 配列を宣言
4:     Dim CanCanModel As String() = _
5:         New String(2) ("山田 優", "蛭原 友里", "押切 もえ")
6:
7:     Private Sub btnDisplayName_Click( _
8:         ByVal sender As System.Object, _
9:         ByVal e As System.EventArgs) Handles btnDisplayName.Click
10:        ' 苗字として何らかの値が入力されているかをチェック。
11:        ' ↓ ここでブレイクポイントを設定する。
12:        If ("".Equals(txtFamilyName)) Then
13:            MessageBox.Show("苗字を入れてください。")
14:        End If
15:        ' 苗字からフルネームを取得する。
16:        txtFullName.Text = SelectName(txtFamilyName.Text)
17:    End Sub
18:
19:    Private Function SelectName( _
20:        ByVal SubName As String) As String
21:        Dim Index As Integer = -1
22:        Dim i As Integer
23:        Dim location As Integer = -1
24:
25:        ' 入力チェック (空文字はNG。また、2文字以上の苗字はない)
26:        If ("".Equals(SubName) Or 3 > SubName.Length) Then
27:            Return "不正な入力です。"
28:        End If
29:
30:        ' 名前の検索
31:        For i = 0 To CanCanModel.GetLength(0) - 1
32:            Index = CanCanModel(i).IndexOf(SubName)
33:            If (0 = Index) Then
34:                location = i + 1
35:                GoTo ReturnName
36:            End If
37:        Next
38:
39:        ' -1の場合は入力された文字が配列にないことを意味している
40:        If (-1 = Index) Then
41:            Return "その名前はありません。"
42:        End If
43:
44:    ReturnName:
45:        Return CanCanModel(location)
46:    End Function
47:
48:    Public Sub New()
49:        InitializeComponent()
50:
51:        ' InitializeComponent()呼び出しの後で初期化を追加します。
52:        MessageBox.Show( _
53:            "山田、蛭原、押切のいずれかの名前を入力してください。")
54:    End Sub
55:
56: End Class

```

図1：サンプルプログラムのUI

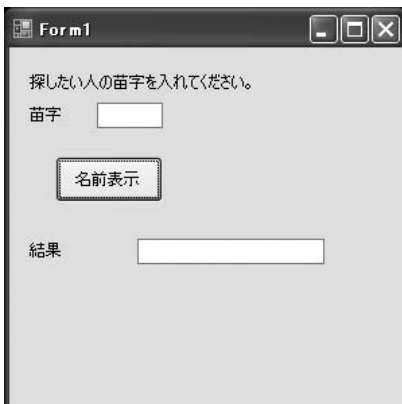
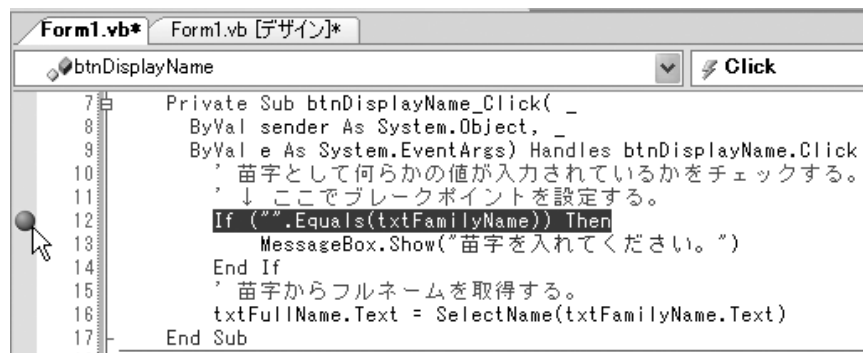


図2：ブレイクポイントを設定した状態



ントを設定したい行へカーソルを移動し、以下のいずれかの操作を行なうことで設定/削除できる。

- ・メニューから [デバッグ] - [ブレイクポイントの設定/解除] を選択

- ・ [F9] ボタンを押す
- ・ コードエディタの一番左端の灰色部分をクリック (図2)

デフォルトであれば、ブレイクポイントを設定した行が茶色く反転される。

●ブレイクポイントを利用したバグ調査

12行目にブレイクポイントを設定したら、早速 [F5] ボタンを押して (またはメニューから [デバッグ] - [デバッグ開始] を選択)、プログラムを実行する。すると、プログラムはブレイク