

スマートな画面クリアーを実装する

「登録ボタンが押された」、或いは「一連の処理が完了した」ので、ユーザが入力した画面の項目をクリアーして初期値に戻したい場合のスマートな処理方法を説明します。

項目をクリアーするには

- 1) 各項目ごとに値をセットしてクリアーする
- 2) 各項目をクリアーするロジックを実装する
- 3) 自分自身をクリアーするロジックを実装したクラスを作成し、再利用する等が考えられます。

1の方法は、フォーム上の項目数分値をセットするコードを書く必要があり、項目数が多くなると大変です、又項目名が長い場合には、タイプミスを引き起こしやすくなります。

2の方法は、以下の様なコードを実装すれば実現できそうですが、Form に PageFrom や Container クラスがあると上手く動作しません、再帰処理可能にすれば良いでしょうが面倒です。又、ある項目は空白やゼロ以外の値を設定した場合などには不向きです。

```
local nControlCount
local i

nControlCount = ThisForm.ControlCount
WITH ThisForm
For i = 1 To nControlCount
    DO CASE
        CASE UPPER(.Controls(i).Class) = 'TEXTBOX'
            IF VarType(.Controls(i).Value) = 'N'
                .Controls(i).VALUE = 0
            ELSE
                .Controls(i).VALUE = ""
            ENDIF
        CASE UPPER(.Controls(i).Class) = 'EDITBOX'
            .Controls(i).VALUE = ""
        CASE UPPER(.Controls(i).Class) = 'XXXXXXX'
            :
    ENDCASE
ENDFOR
ENDWITH
```

3の方法は、VFP の OOP を活用して最もエレガントに項目のクリアー処理を実装できます。この方法を実装するポイントは、フォームに配置される全てのビジュアル・クラスの情報をフォー

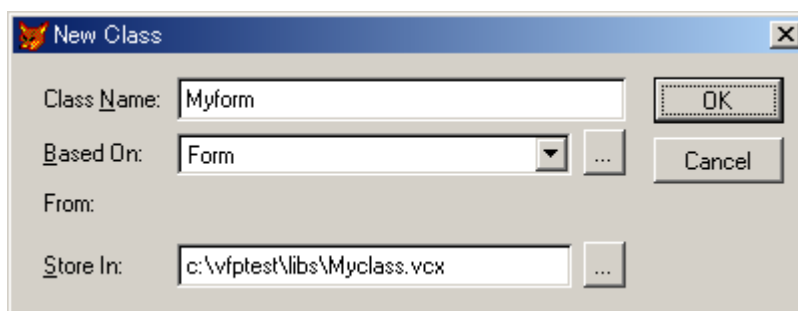
ム・レベルで管理することです。

1. クラス・ライブラリとフォーム・クラスの作成

1 - 1 . フォーム・クラスの作成

画面クリアーを実装するために、新規にクラスを作成します。

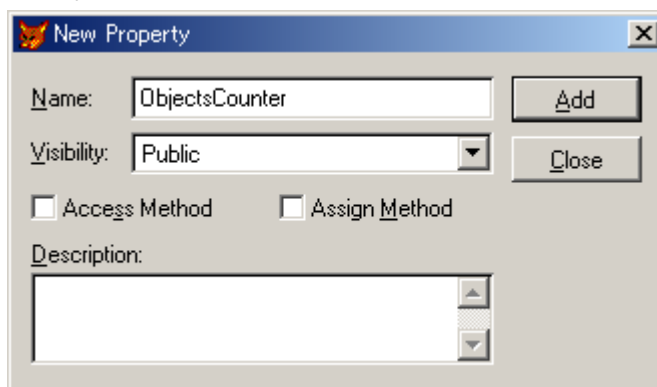
Project Manager の **Class Libraries** をクリックしハイライトした後に、**New...** ボタンをクリックします。



New Class ダイアログボックスの **Class Name** にクラス名 (Myform) を入力します、**Based On** に継承元の Form を選択します、**Store In** にクラス・ライブラリの格納先を指定します。

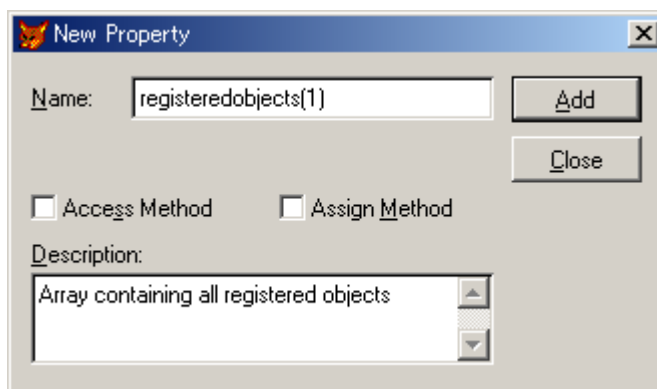
OK ボタンをクリックします。**Class Designer** が表示されます。

フォーム上にあるクラスの数とプロパティを登録します。**Class - New Property...** を選択します。**New Property** ダイアログボックスの **Name** に **ObjectsCounter** と入力し **Description** にこのプロパティの説明を入力した後に、**Add** ボタンをクリックします。



続いて、フォーム上のオブジェクトを記憶するためのプロパティを登録します。

RegisteredObjects(1) と入力します、**Description** にこのプロパティの説明を入力します。**Add** ボタンをクリックします。

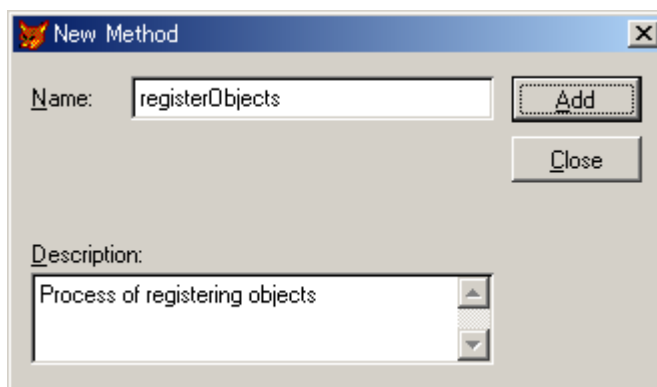


Add ボタンをクリックしてプロパティを登録した後、**Close** ボタンをクリックしてダイアログボックスを閉じます。

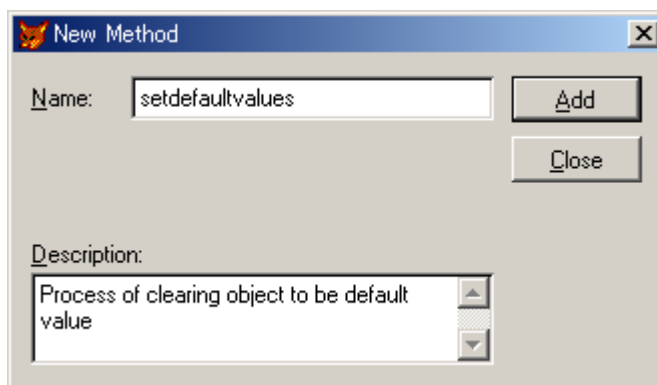
Properties ウィンドウで **ObjectsCounter** プロパティの値を **0** に設定します。

1 - 2 . フォーム・メソッドの登録

Class - New Method... を選び、フォーム上のオブジェクトを記憶するための **RegisterObjects** メソッドを登録します。



Add ボタンをクリックします。続けて、フォーム上の各オブジェクトを初期値に戻すために各オブジェクトの **SetDefaultValue** メソッドを呼び出す **SetDefaultValues** メソッドを登録します。



Add ボタンをクリックした後、**Close** ボタンをクリックしてダイアログボックスを閉じます。

1 - 3 . メッドにコードを記述する

Myform に登録した **RegisterObjects** と **SetDefaultValues** メソッドのコードを入力します。**RegisterObjects** メソッドが呼ばれると呼び出し元のオブジェクトの情報を **RegisteredObjects(1)** プロパティに格納します。**SetDefaultValues** メソッドはフォーム上のオブジェクトの値を初期値に戻すために各オブジェクトの **SetDefaultValue** メソッドを呼び出します。

Properties ウィンドウの **RegisterObjects** メソッドをダブルクリックしてコード・ウィンドウを開き、下記のコードを入力します：

```
LPARAMETERS oRegObject
```

```
* 登録されたオブジェクト数をカウントアップする
```

```
ThisForm.ObjectsCounter = ThisForm.ObjectsCounter + 1
```

```
* 登録されたオブジェクト数に合わせて、配列を再配列する
```

```
Dimension ThisForm.RegisteredObjects(ThisForm.ObjectsCounter)
```

```
* オブジェクトへの参照を配列に格納する
```

```
ThisForm.RegisteredObjects[ThisForm.ObjectsCounter] = oRegObject
```

コード・ウィンドウを閉じます。次に、**SetDefaultValues** メソッドをダブルクリックしてコード・ウィンドウを開き下記のコードを入力します：

```
LOCAL lErrFlg;
```

```
lOldError
```

```
** このコードは全てのオブジェクトをデフォルトの値に戻すために、
```

```
** 各オブジェクトの SetDefaultValue メソッドを呼び出す
```

```
ThisForm.LockScreen = .T.
```

```
lErrFlg = .F.
```

```
lOldError = ON('error')
```

```
ON ERROR lErrFlg = .T.
```

```
FOR i = 1 TO This.ObjectsCounter
```

```
    This.RegisteredObjects[i].SetDefaultValue
```

```
ENDFOR
```

```
ThisForm.LockScreen = .F.
```

```
ON ERROR &lOldError
```

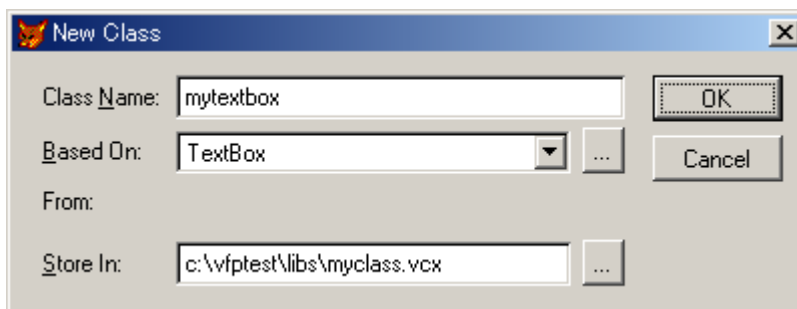
コード・ウィンドウを閉じます。

2 . ビジュアル・クラスの作成

2 - 1 . Textbox クラスの作成

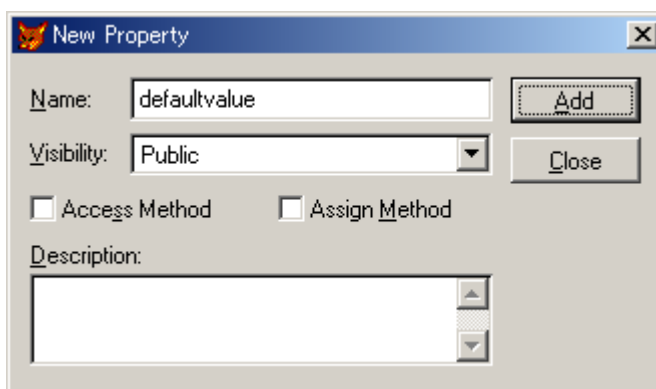
Project Manager の **myclass** をクリックしハイライトした後、**New** ボタンをクリックします。

Class Name に **mytextbox** と入力し、**Based On** に **TextBox** を選択します。**Store In** は、先ほど **Myform** クラスを作成した **myclass.vcx** がフルパスで表示されているので変更しません。



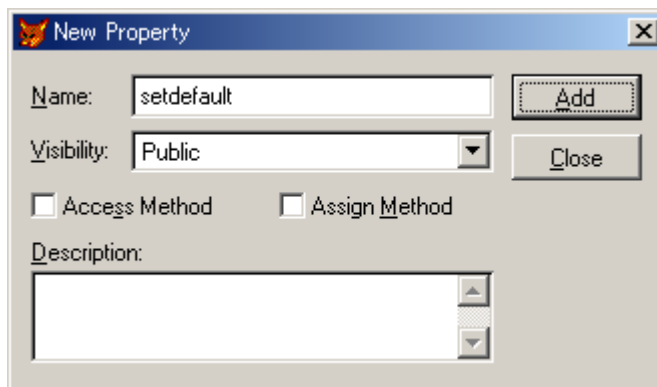
OK ボタンをクリックし、ダイアログボックスを閉じます。**Class Designer** が表示されます。

Textbox が初期化されたときに表示する初期値を保持するために **DefaultValue** プロパティ (**Value** プロパティの値を保持します) を追加します。**Class - New Property...** を選択します。**New Property** ダイアログボックスの **Name** に **defaultvalue** と入力し、**Add** ボタンをクリックします。



続けて、**SetDefault** プロパティを追加します。このプロパティは、**SetDefault** メソッドが呼び出された時の初期化実行を迂回させるために使います。

Name に **SetDefault** と入力し、**Add** ボタンをクリックした後、**Close** ボタンをクリックしてダイアログボックスを閉じます。



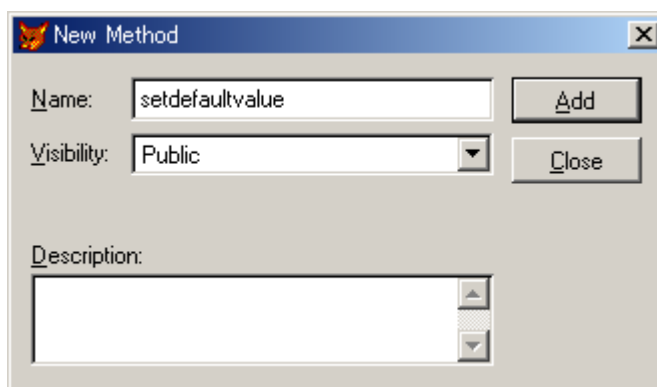
Properties ウィンドウで **SetDefault** プロパティの値を **.T.** に設定します。

Init Event をダブルクリックしコード・ウィンドウを表示させ、以下のコードを入力します。

```
ThisForm.RegisterObjects(This)
```

```
This.DefaultValue = This.Value
```

Class – New Method を選択し、初期値をセットするためのメソッドを追加します。



Add ボタンをクリックした後、**Close** ボタンをクリックしてダイアログボックスを閉じます。**Properties** ウィンドウの **SetDefaultValue** をダブルクリックしコード・ウィンドウを開き下記のコードを入力します。

```
IF This.SetDefault = .T.
```

```
    This.Value = This.DefaultValue
```

```
ENDIF
```

コード・ウィンドウを閉じます。

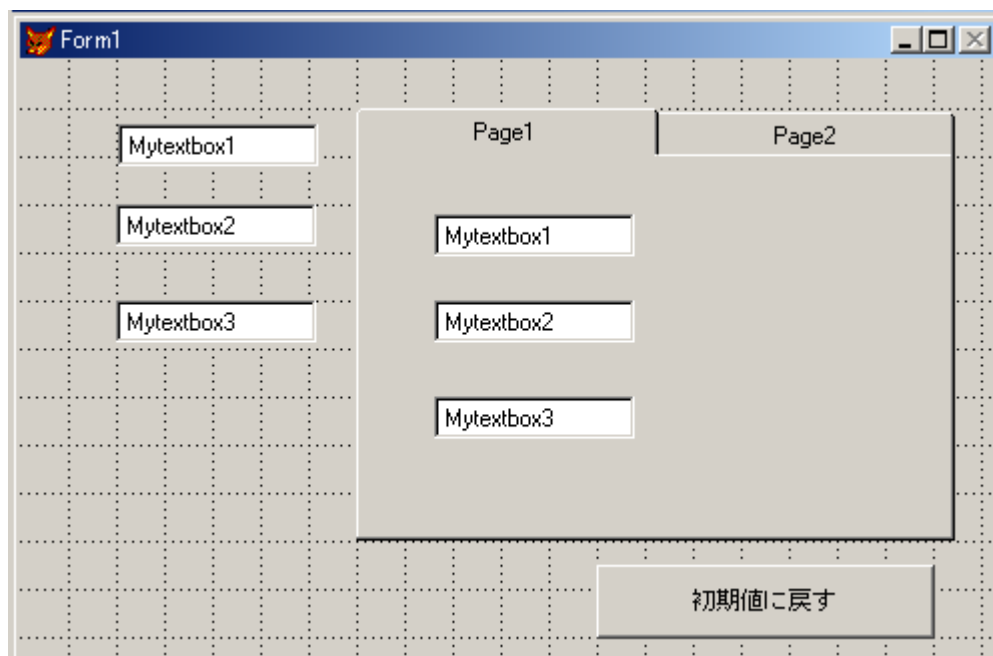
他のビジュアル・クラスも同様の方法で作成します。

3 . 作成したクラスの使用方法

Project Manager の **Forms** をクリックしハイライトした後、**New** ボタンをクリックします。

Form Designer が表示されます。これに **Myform** を **ドラッグ&ドロップ** します、**Form set** オブジェクトを作成するか問い合わせるダイアログボックスが開くので、**Yes** を選択します。**Property** ウィンドウで **Form1** を選択した後、**Form – Remove From** を選択し **Form1** を **Form Designer** から削除します。次に、**Form – Remove Form Set** を選択し **Form Set** を **Form Designer** から削除します。これで、**Form Designer** には **MyFormxx** だけが残ります。

下図のサンプル画面のように適当に **MyTextBox** を配置します。この例では、**_PageFrame** を継承して **MyPageFrame** も作成してあります。



Property ウィンドウで各クラスの初期値を設定します。

Form 上のビジュアル・クラスの **Value** プロパティに初期値を設定します。一連の処理が終わり、**Form** 上の各項目をクリアーする際、項目のクリアーを迂回したいクラスは **SetDefault** プロパティを **.F.** に設定します。

初期値に戻す ボタンの **Click** イベントに、以下のコードを入力します。

```
ThisForm.SetDefaultValues
```

Form を保存して実行します。各項目に適当な値を入力し**初期値に戻す** ボタンをクリックしてみてください。見事に各項目がクリアーできます。

動作説明

クラスの作成方法も説明しているので面倒な感じがするかもしれませんが、動作はとても単純です。

オブジェクト情報の保持

- 1 . Form がロードされ Form 上のオブジェクトの Init メソッドが呼ばれます。
- 2 . Init メソッドは自分自身の情報を Form に渡すために、Form の RegisterObjects メソッドを呼びだし、自分自身の情報を渡します。
- 3 . RegisterObjects メソッドが呼び出されると、ObjectsCounter をプラス 1 し、配列のサイズを再配列し、呼び出し元のオブジェクトの情報を RegisterdObject 配列に保持します。

画面のクリアー

- 1 . 画面のクリアーを実行するために、Form の SetDefaultValues メソッドを呼びます。
- 2 . SetDefaultValues メソッドは、RegisterdObject 配列に保持してある Form 上の各オブジェクトの SetDefaultValue メソッドを呼び出します。
- 3 . 各オブジェクトの SetDefaultValue メソッドは、SetDefault プロパティが .T.であれば、DefaultValue に保持してある値を自分自身に設定します。

この様な、非常に単純明快な動作を行っています。

まとめ

項目のクリアー処理を実装したクラスを作成しておけば、Form 上の各項目を一括クリアーする時に、**ThisForm.SetDefaultValues** : たった一行のコードで、各項目のクリアー処理を行う事ができます。

TextBox の Value のみを初期化するように作ってありますが、Init メソッドで BackColor ・ ForeColor 等のプロパティも保持して SetDefaultValue メソッドで元に戻せば、色までも初期化できるようになります。

備考：サンプルの実行方法

適当なプロジェクトを作成します。Form1 と MyClass をプロジェクトに追加し、実行します。