

# Windowsプログラミングを体験しよう

～ Visual Basic Editorによるプログラミング ～

## <概要>

私たちがコンピュータ上で利用している，ブラウザやワープロ，ゲームなどのアプリケーションは，様々なコンピュータ用言語によってプログラムすることで作られている。ここでは，汎用的なプログラム開発言語であるVB (VisualBasic)を用いた簡単なプログラミングを体験し，コンピュータの基本的な処理の仕組み及び簡単なアルゴリズム（手順）を学ぶ。

<キーワード>プログラミング，Visual Basic，アルゴリズム

## 1. プログラムを作るための開発環境

従来のMS-DOS上のアプリケーションは，BASIC言語やC言語といったプログラミング言語により開発されていたため，プログラミングの基礎学習にはBASIC言語が使われていた。

BASIC言語とはBeginner's All-purpose Symbolic Instruction Codeの頭文字をとったもので，1964年にアメリカのダートマス大学で初心者向きの会話型言語として開発され，1975年にビル・ゲイツによってパソコン用に作り直されたことにより，パソコン上のプログラミング言語として普及した。

現在，Windows上で動作するアプリケーションを作成するための汎用的なプログラミング言語には，Visual Basic (VB)があり，オブジェクト指向という新しい考え方を取り入れ，イベントドリブンというプログラム方式で処理が行われている。

ここでは，VBのプログラミング環境がないことを考慮し，ワープロや表計算等のアプリケーションの繰り返し処理やより複雑な処理を記述するために内蔵されたVBA (Visual Basic for Applications)を用いてプログラミングを体験する。

VBAの言語仕様はVBに比べいろいろな制約があるが，VBでのプログラミングとほとんど同じ感覚で学習できる。

## 2. VisualBasicEditor の操作

VBAによりプログラミングするためにV

isual Basic Editor (VBE)を起動し，基本操作を習得する。

### (1) 準備

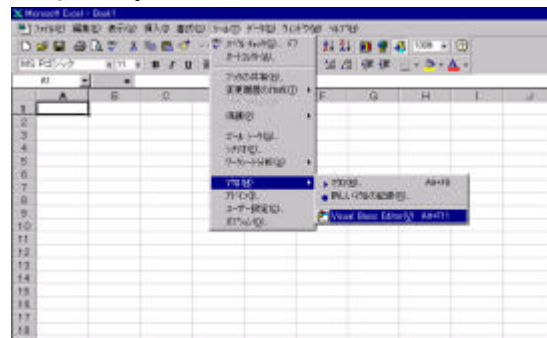
#### 【操作】

Excelを起動する。

Excelメニューの

[ツール]-[マクロ]-[Visual Basic Editor]

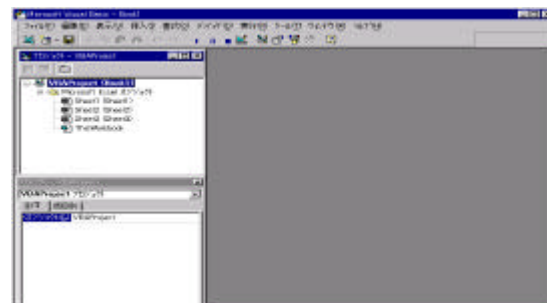
を選択する。



ユーザーフォームの挿入

VBEメニューの

[挿入]-[ユーザーフォーム]を選択する。



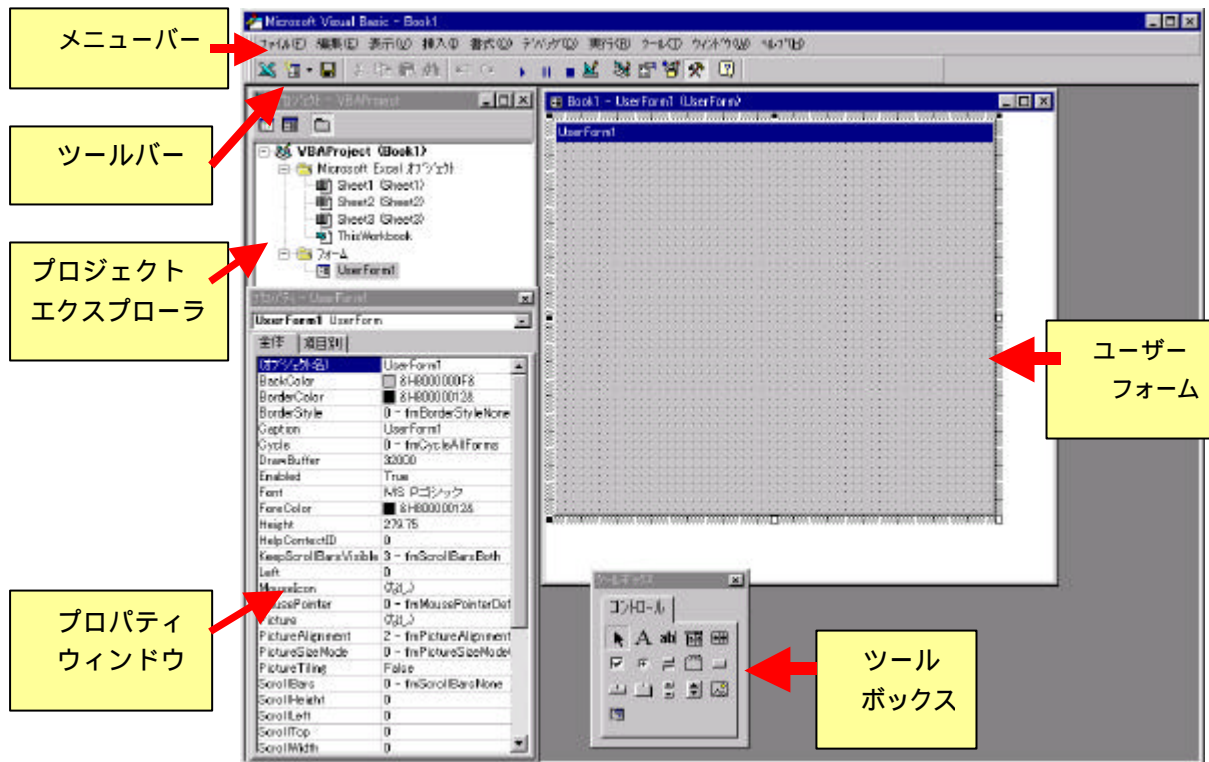
### (2) VBEの画面構成

メニューバー

各種の操作コマンドを実行するメニュー

ツールバー

よく使うコマンドをアイコン化したもの



**ユーザーフォーム**  
 プログラム作成の基本となる画面。  
 ここにプログラムの実行結果をイメージしながらコントロール配置し、プログラムを入力する。

**ツールボックス**  
 ユーザーフォーム上に配置する様々な働きをするコントロールの一覧。

**プロパティウィンドウ**  
 各コントロールの色や名前、機能、性質などの属性（プロパティ）を表示する。必要に応じて変更することができる。

(3) コントロールの種類と働き  
 ユーザーフォームに配置できるコントロールのうち、主に使用するものを次に示す。

### VBEで使えるコントロールの主な種類

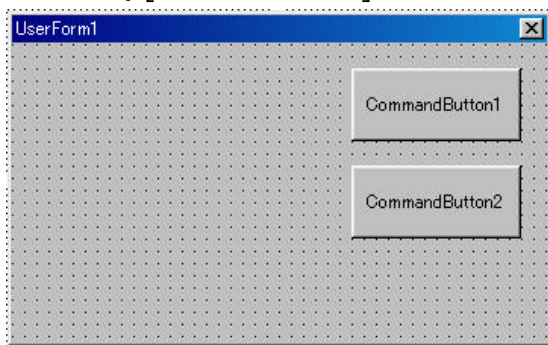
The diagram illustrates the main types of controls available in the VBE toolbox. Each control is shown with a callout box explaining its function:

- ラベル (Label):** 文字の表示をする (Display text).
- テキストボックス (Text Box):** 文字や数字の入力と表示をする (Input and display of text and numbers).
- オプションボタン (Option Button):** 複数の項目から1つを選択する (Select one item from multiple items).
- リストボックス (List Box):** 文字のリストを表示・選択をする (Display and select a list of text).
- コマンドボタン (Command Button):** マウスのクリックでプログラムを実行する (Execute the program with a mouse click).
- イメージ (Image):** 画像を表示をする領域 (Area to display an image).

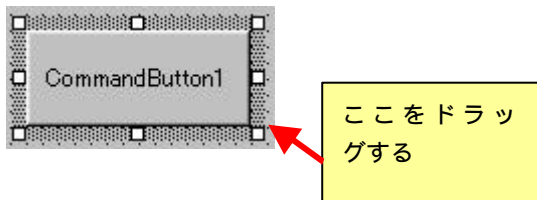
### 3. プログラミングの基本操作

**【練習課題 1】** コマンドボタンをクリックして背景の色を変える  
～ R G B 関数で画面に表示される色の構成方法を理解しよう ～

- (1) ユーザーフォームにコントロール配置  
ツールボックスからコマンドボタンをマウスでクリックし選択する。  
ユーザーフォーム上の適当な位置をマウスでドラッグしてコントロール(コマンドボタン)を配置する[ CommandButton1 ]  
と同様にもう一つコマンドボタンを配置する。[ CommandButton2 ]

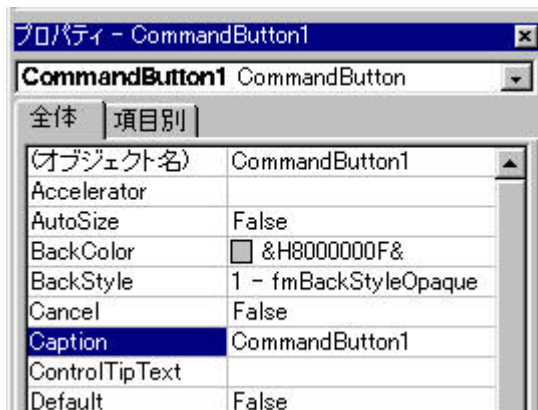


マウスでコマンドボタンをクリックして選択状態にすると、大きさが変えられる。

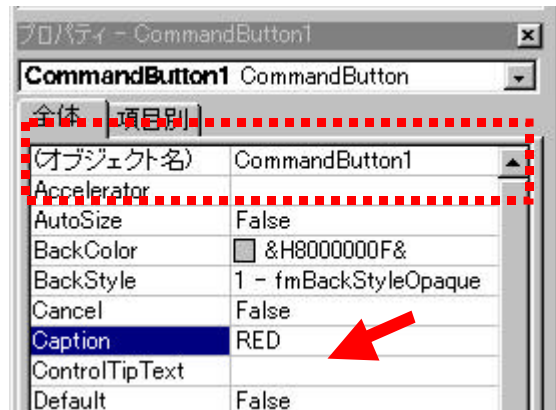


- (2) プロパティ(属性)の変更

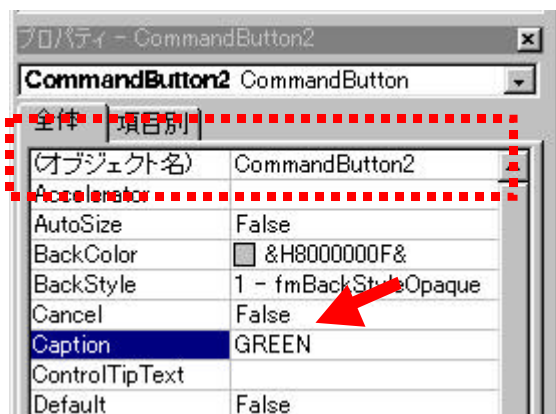
ユーザーフォームや配置したコマンドボタンをマウスで選択すると、プロパティウィンドウにそれぞれの属性が表示される。この内容を変更することにより名前や色、表示文字、性質などを変えることができる。



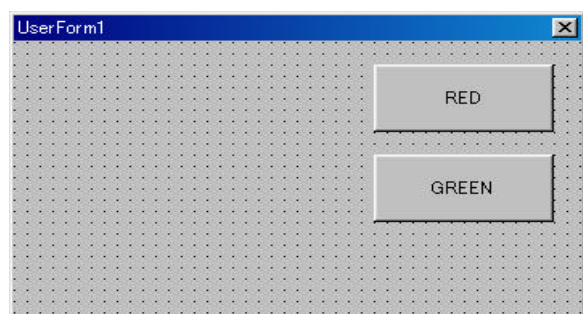
コマンドボタンのプロパティのうち [Caption]欄の[CommandButton1]を[RED]に変更する。コマンドボタンに表示される文字が変わることを確認する。



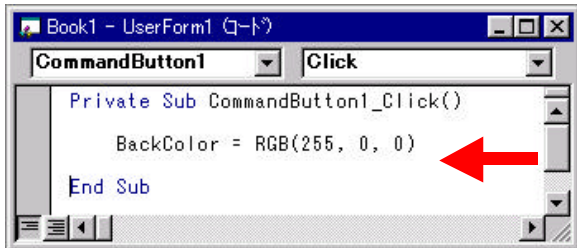
同様にもう一つのコマンドボタンの [Caption]プロパティの[CommandButton2]を[GREEN]に変更する。コマンドボタンに表示される文字が変わることを確認する。



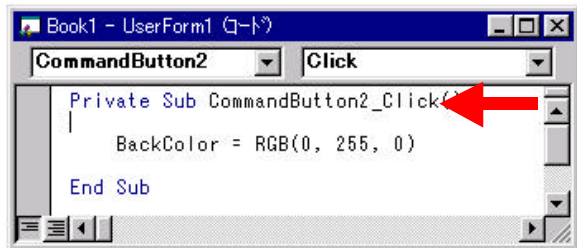
ユーザーフォームが次のようになる。このときコマンドボタンのオブジェクト名は変わっていないことを理解する。




(3) プログラムコード(命令)の記述  
 コマンドボタン[CommandButton1]をダブルクリックすると次の画面が表示される。下図のように矢印の位置に  
`BackColor = RGB(255, 0, 0)`  
 と入力する。



同様にコマンドボタン  
 [CommandButton2]  
 をダブルクリックし、次の画面を表示させる。下図のように矢印の位置に  
`BackColor = RGB( 0, 255, 0)`  
 と入力する。

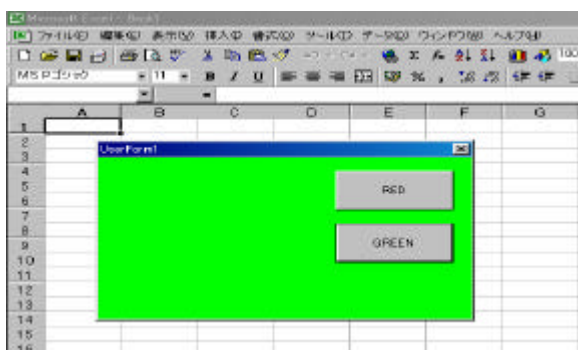


これで必要なプログラム入力が終わったので、コードの入力画面を閉じる

(4) プログラムの実行  
 メニューバー[実行]-[ユーザーフォームの実行]  
 またはツールバーの  をクリックする。



実行画面が表示される



コマンドボタンを交互にマウスでクリックするたびにユーザーフォームの背景色が赤と緑に変化することを確認する。

実行の停止はメニューバーの  
 [実行]-[リセット]または実行画面を閉じる。

(5) プログラムの解説  
 プログラムの処理単位(プロシージャ)  
`Private Sub CommandButton1_Click()`  
 オブジェクト名 イベント名  
 (コントロール名)

`End Sub` ← プロシージャの終わり  
 ユーザーフォームの背景色[BackColor]  
 プロパティをRGB関数で指定した色にする。

`BackColor = RGB(255, 0, 0)` 赤色

RGB関数と表示色の成り立ち  
 表示色は赤(Red), 緑(Green), 青(Blue)  
 の成分がどれくらいの割合で加色されているかで表す。

RGB ( 赤成分 , 緑成分 , 青成分 )  
 (0-255) (0-255) (0-255)

<例>

- RGB(255, 0, 0) ... 赤色
- RGB( 0,255, 0) ... 緑色
- RGB( 0, 0,255) ... 青色
- RGB(255, 0,255) ... 赤色 + 青色 = 紫色
- RGB(255,255, 0) ... 赤色 + 緑色 = 黄色
- RGB( 0,255,255) ... 緑色 + 青色 = 水色
- RGB(255,255,255) ... 赤 + 緑 + 青 = 白色
- RGB( 0, 0,128) ... どんな色になるか?

【練習問題 1 - 1】 RGB関数の各値を変えて、プログラムを実行してみなさい。

【練習問題 1 - 2】 光の三原色(赤、緑、青)の各成分を0~255の値で指定して色の合成をすると何通りの色が表示できるか考えなさい。

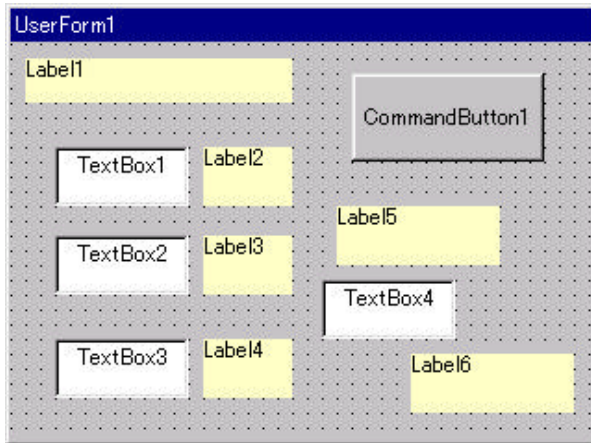


## 【練習課題 2】自分の生まれた日の曜日を求める

～ グレゴリオ歴の規則からうるう年を考慮し、曜日を求める手順 ～

### (1) ユーザーフォームにコントロール配置

次のようにラベル(Label1 ~ Label6)、テキストボックス(TextBox1 ~ TextBox4)、コマンドボタン(CommandButton1)の各コントロールをフォームに配置する。

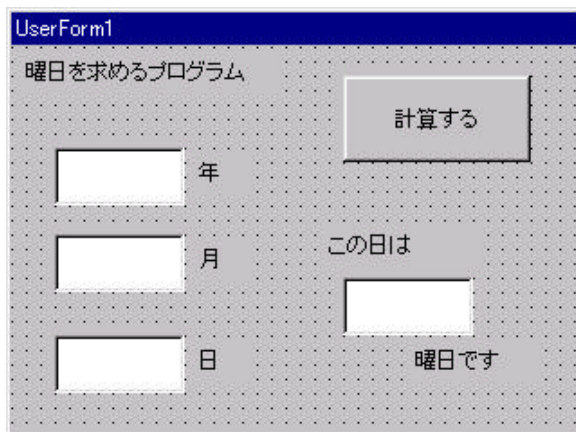


(注：一部説明のためにプロパティが変更された画面です)

### (2) プロパティの設定

ラベル(Label1 ~ Label6)とコマンドボタン(CommandButton1)をそれぞれクリックし、プロパティウィンドウの Caption プロパティの値を次のように変更する。

コントロール	プロパティ	設定内容
Label1	Caption	曜日を求める
Label2	Caption	年
Label3	Caption	月
Label4	Caption	日
Label5	Caption	この日は
Label6	Caption	曜日です
CommandButton1	Caption	計算する




次の画面のようになることを確認する。

### (3) プログラムコード(命令)の記述

コマンドボタンをダブルクリックし、コード記述画面を出して、次のようにコードを記述する。

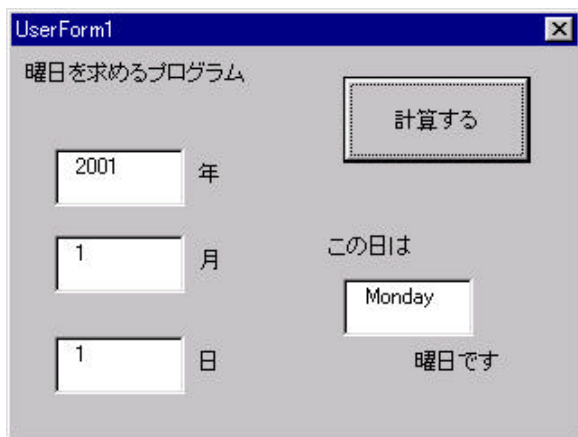
```
Private Sub CommandButton1_Click()
    Dim mt(12) As Integer
    mt(1)=31: mt(2)=28: mt(3)=31: mt(4)=30
    mt(5)=31: mt(6)=30: mt(7)=31: mt(8)=31
    mt(9)=30: mt(10)=31: mt(11)=30: mt(12)=31
    y = Val(TextBox1.Text)
    m = Val(TextBox2.Text)
    d = Val(TextBox3.Text)
    If (y Mod 4)=0 And (y Mod 100)<>0 Or
        (y Mod 400)=0 Then
        mt(2) = 29
    Else
        mt(2) = 28
    End If
    zure = y+(y-1)¥4 - (y-1)¥100 + (y-1)¥400
    days = 0
    For k = 1 To m - 1
        days = days + mt(k)
    Next k
    week = (zure + days + d - 1) Mod 7
    If week = 0 Then
        TextBox4.Text = "Sunday"
    ElseIf week = 1 Then
        TextBox4.Text = "Monday"
    ElseIf week = 2 Then
        TextBox4.Text = "Tuesday"
    ElseIf week = 3 Then
        TextBox4.Text = "Wednesday"
    ElseIf week = 4 Then
        TextBox4.Text = "Thursday"
    ElseIf week = 5 Then
        TextBox4.Text = "Friday"
    ElseIf week = 6 Then
        TextBox4.Text = "Saturday"
    Else
        TextBox4.Text = "Error"
    End If
End Sub
```

#### (4) プログラムの実行

メニューバー[実行]-[ユーザーフォームの実行]  
またはツールバーの  をクリックする。



実行画面の年、月、日のテキストボックス  
に自分の誕生日などの数字を入力する。



「計算する」と書かれたコマンドボタンを  
クリックすると、その日の曜日が表示さ  
れる。

#### (5) プログラムの解説

年月日の値から、どのように曜日が求めら  
れるか、そのアルゴリズム(手順)を考えて  
みる。地球の公転周期は 365.2422日  
である。1年を365日にするはずれが生じ  
るため、4年に一度閏年が設けられる。

このグレゴリオ暦の規則から

西暦の年数が4で割れ、かつ100で割  
り切れない年を閏年とする。

西暦が400で割切れる年も閏年とする。

都合上西暦1年1月1日を月曜日とする。

1年365日を1週間の7日で割ると1日  
余ることから、Y年たつとY日分曜日がずれ  
ることになる。また、閏年の年数分だけさら  
にずれる。

このずれる日数は次の式で求まる。

$zure = y + (y-1) \times 4 - (y-1) \div 100 + (y-1) \div 400$   
曜日を求めたい年の1月1日からの日数は、  
days = 0

For k = 1 To m - 1

days = days + mt(k)

Next k

とすることで、days 日と求められる。

西暦1年1月1日(月曜日)からのずれた日  
数を7日で割り、余りの数字 week から曜日を  
求める。

$week = (zure + days + d - 1) \text{ Mod } 7$

week=0 を日曜日、week=1 を月曜日と考える。

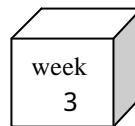
次の条件判断文で答えをテキストボック  
スに表示する。

If week = 0 Then

TextBox4.Text = "Sunday"

#### 変数

このプログラムで使われている y,m,d  
zure , days , week という文字は変数と呼ば  
れ、値を入れる容器のようなものと考え  
る。

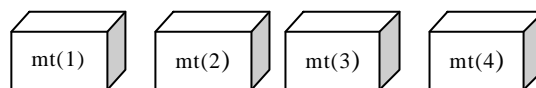


変数につける名前を変数名といい、変数  
名には英数文字を用い、必ずアルファベッ  
トから始まる名前とする。

#### 配列変数(一次元配列)

このプログラムで使われている mt(1) ~  
mt(12) という文字は配列変数と呼ばれる。  
配列名(添字)という形で記述される。

配列は Dim 文であらかじめ宣言して  
おく必要がある。Dim mt(12) As Integer



#### 条件判断(IF文)

条件を判断して、それぞれの条件に応じ  
たプログラムを実行するには、次のような  
形で記述する。

If 条件式1 Then

条件式1を満たしたときに実行

Elseif 条件式 Then

条件式2を満たしたとき実行

Else

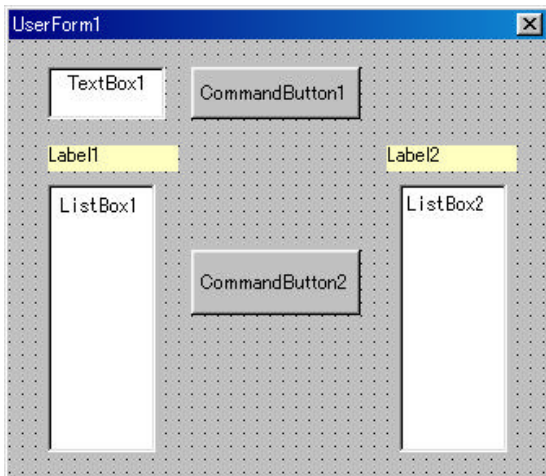
以上の条件式を満たさないときに実行

End If

**【練習課題3】数値を大きい順（降順）に並び替える（ソート）**

～ 直接選択法による並び替えのアルゴリズムを理解しよう ～

(1) ユーザーフォームにコントロール配置  
 次のようにラベル(Label1,Label2), テキストボックス ( TextBox1 ), リストボックス ( ListBox1,ListBox2 ), コマンドボタン ( CommandButton1, CommandButton2 ) の各コントロールをフォームに配置する。



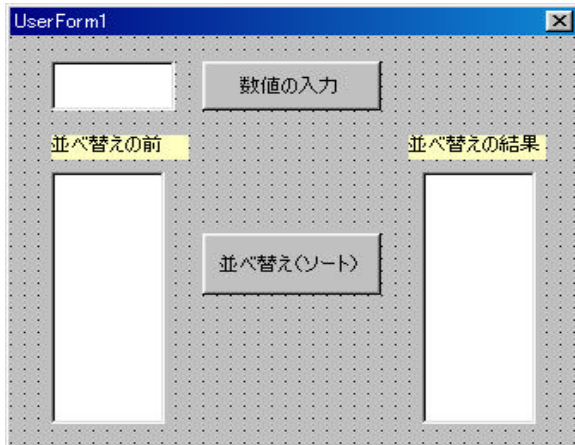
(注：一部説明のためにプロパティが変更されている)

(2) プロパティの設定

ラベルとコマンドボタンをそれぞれクリックし、プロパティウィンドウの Caption プロパティの値を次のように変更する。

コントロール	プロパティ	設定内容
Label1	Caption	並べ替えの前
Label2	Caption	並べ替え結果
CommandButton1	Caption	数値の入力
CommandButton2	Caption	並べ替え(ソート)

次の画面のようになることを確認する。



(注：一部説明のためにプロパティが変更されている)

(3) プログラムコード (命令) の記述  
 コマンドボタンをダブルクリックし、コード記述画面を出して、次のようにコードを記述する。

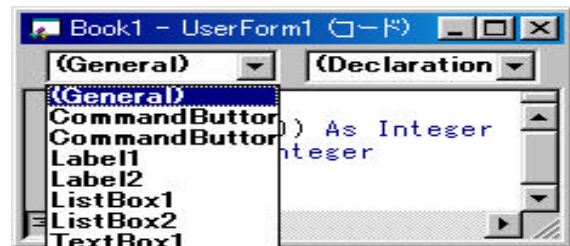
```
Dim kazu(50) As Integer
Dim n As Integer
```

```
Private Sub CommandButton1_Click()
    n = n + 1
    kazu(n) = Val(TextBox1.Text)
    ListBox1.AddItem kazu(n)
    TextBox1.Text = ""
End Sub
```


```
Private Sub CommandButton2_Click()
    For i = 1 To n - 1
        For j = i + 1 To n
            If kazu(j) > kazu(i) Then
                Max = kazu(j)
                kazu(j) = kazu(i)
                kazu(i) = Max
            End If
        Next j
    Next i

    For k = 1 To n
        ListBox2.AddItem kazu(k), k - 1
    Next k
End Sub
```

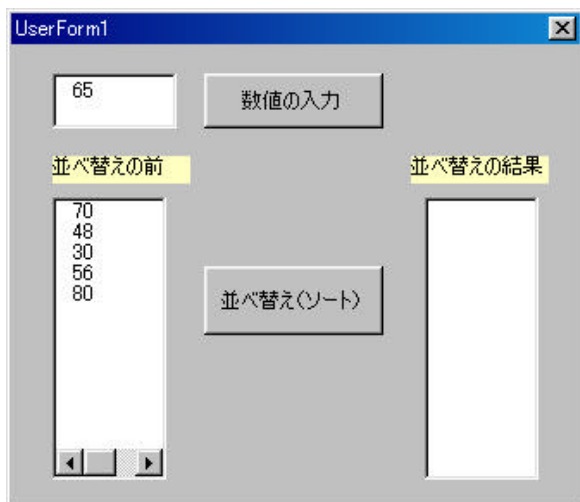
上記のプログラムコードの上二行は、図のようにコマンド名の部分を ( General ) に変更し、コードを記述する。



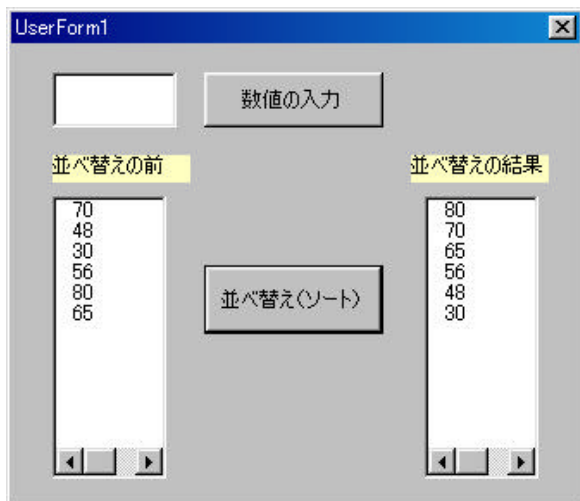
#### (4) プログラムの実行

メニューバー[実行]-[ユーザーフォームの実行]  
またはツールバーの  をクリックする。

テキストボックスに適当な数値を入力し、  
[数値の入力]コマンドボタンをクリック  
する。これを数回繰り返して、次の画面の  
ようにする。



[並べ替え(ソート)]コマンドボタンをク  
リックすると、次の画面のように並べ替  
えられた結果が表示される。



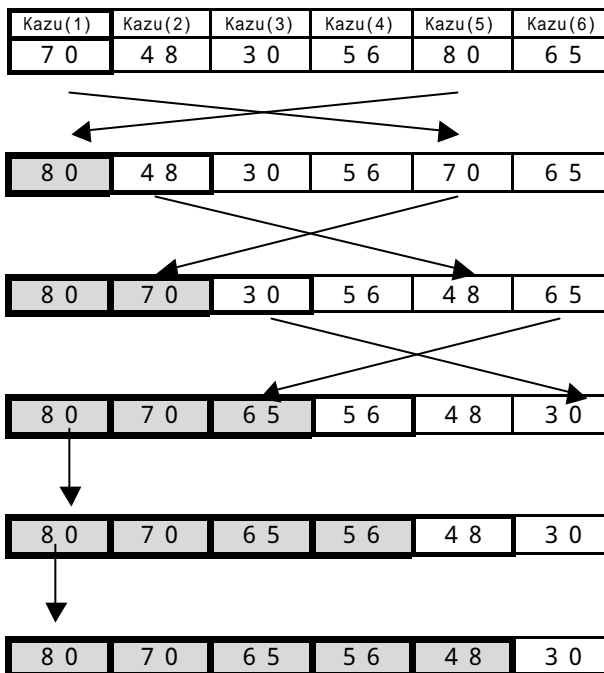
#### (5) プログラムの解説

並べ替えをするには、いろいろな方法があ  
るが、ここでは直接選択法と呼ばれるアルゴ  
リズムについて考える。

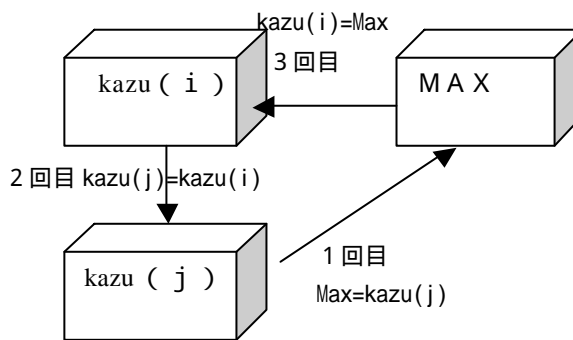
配列変数 kazu(1)に代入された値を基準に、  
右側にある数値の中から最大のものを見つけ、  
データを入れ替える。これで、最大値が確定  
する。

次に配列変数 kazu(2)に代入されている  
値を基準に、右にあるデータの中から最  
も大きい値を見つけ、データを入れ替え  
る。これで、2番目に大きい値が確定さ  
れる。

以上の操作を繰り返すことによりデータ  
がすべて降順に並び変わる。



#### データを入れ替える手順



#### 繰り返し文 (FOR文)

FOR文は決められた回数を繰り返し  
て実行する場合に使われる。

例 For k = 1 To n

    ListBox2.AddItem kazu(k), k - 1

Next k

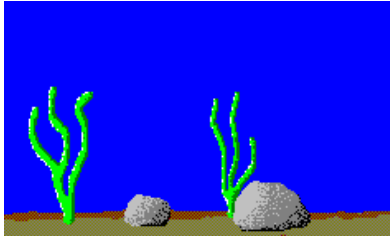
変数 k に初期値 1 を入れ、Next までの  
処理を実行した後、変数 k の値を 1 増や  
し、k = n になるまで繰り返す。



【練習課題 4】自分で描いた水槽に魚を泳がせ，マウスの動きに追従させる。  
 ~ 画像（イメージデータ）を表示する座標の考え方を理解する ~

(1) 事前準備

ペイントツールを使って，次のような水槽と右向きの魚，左向きの魚の画像を描く。



水槽の絵  
water.bmp



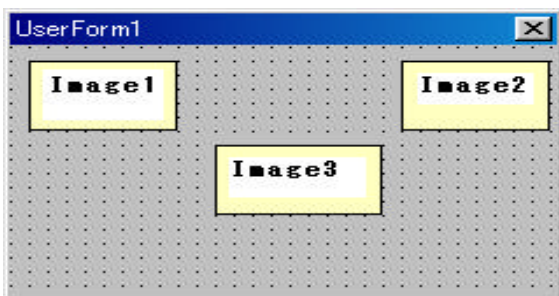
Fish\_r.GIF



Fish\_l.GIF

魚の絵は，透過G I F 形式にすると効果的

(2) ユーザーフォームにコントロール配置  
 イメージコントロール(Image1 ~ Image3)を次の画面のように配置する。



(注：一部説明のためにプロパティが変更されている)

(3) プロパティの設定

ユーザーフォームに水槽の絵を表示する。  
 Picture プロパティを選択する。



作成した水槽のファイル名を選択する。



Image1 コントロールに右向の魚を表示  
 ユーザーフォームと同様に Image1 の picture プロパティに作成した魚のファイルを指定する。

その他，バックスタイルおよびボーダースタイルのプロパティを次のように変更する。この変更は Image1 ~ Image3 のコントロールすべてに設定する。

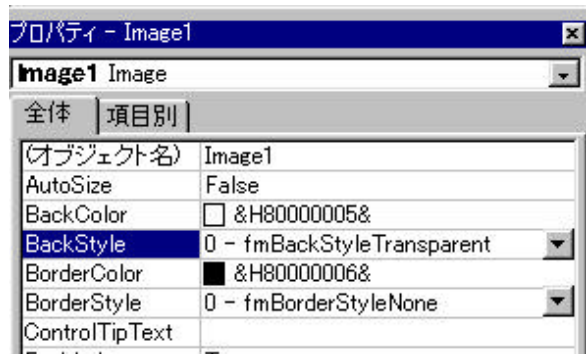
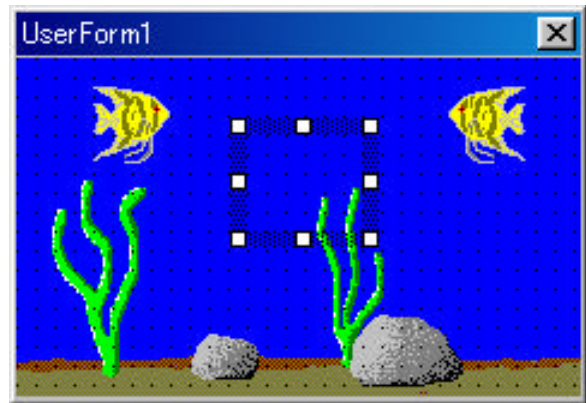


Image2 コントロールに左向の魚を表示  
 Image1 コントロールと同様な手順で魚のファイルを指定する。

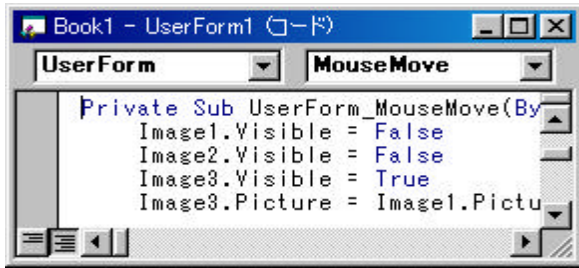
次のような画面になることを確認する。



(4) プログラムコード（命令）の記述

ユーザーフォームをダブルクリックし，コードの入力画面を表示する。

次の図のようにイベントの内容を [MouseMove] に変更し，コードを入力する。




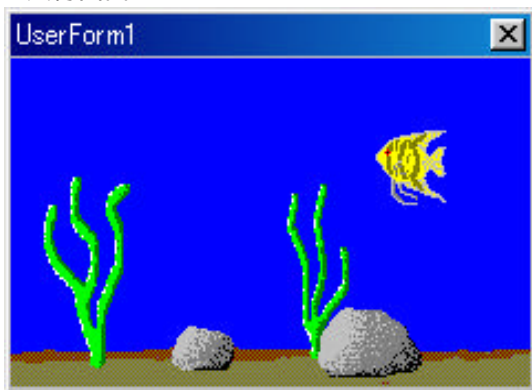
```

Private Sub UserForm_MouseMove(ByVal
Button As Integer, ByVal Shift As
Integer, ByVal X As Single, ByVal Y As
Single)
    Image1.Visible = False
    Image2.Visible = False
    Image3.Visible = True
    Image3.Picture = Image1.Picture
    If X > Image3.Left Then
        Image3.Picture = Image1.Picture
        Image3.Left = Image3.Left + 1
    ElseIf X < Image3.Left Then
        Image3.Picture = Image2.Picture
        Image3.Left = Image3.Left - 1
    Else
        Image3.Left = Image3.Left
    End If
    If Y > Image3.Top Then
        Image3.Top = Image3.Top + 1
    ElseIf Y < Image3.Top Then
        Image3.Top = Image3.Top - 1
    Else
        Image3.Top = Image3.Top
    End If
End Sub

```

(5) プログラムの実行

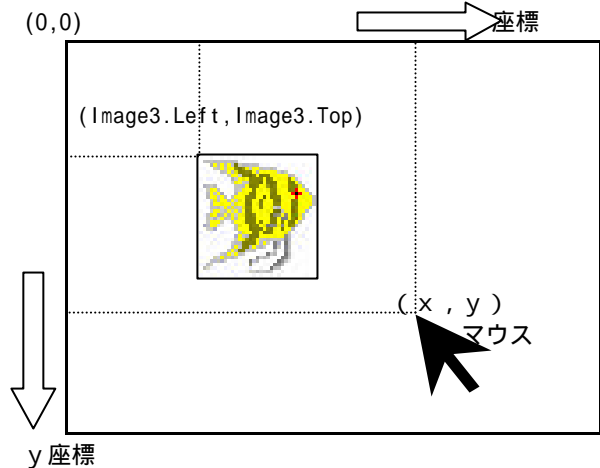
メニューバー[実行]-[ユーザーフォームの実行]  
 またはツールバーの  をクリックする。  
 実行画面



マウスカーソルを画面上で動かすと、マウスを追いかけて魚が泳ぐように動作することを確認する。

(6) プログラムの解説

画面上の座標



マウスが動くと UserForm の MouseMove イベントにより、マウスの位置の x 座標値が変数 x に、y 座標値が変数 y に与えられる。

魚の画像の左上の位置は、x 座標値を Image3.Left, y 座標値を Image3.Top で知ることができる。

マウスが魚より右側にあれば、Image3 に右向き魚の画像(Image1.Picture)を表示させ、魚の位置を右に移動させる。

```

If X > Image3.Left Then
    Image3.Picture = Image1.Picture
    Image3.Left = Image3.Left + 1

```

マウスが魚より左側にあれば、Image3 に左向き魚の画像(Image2.Picture)を表示させ、魚の位置を左に移動させる。

```

Elseif X < Image3.Left Then
    Image3.Picture = Image2.Picture
    Image3.Left = Image3.Left - 1

```

同様にマウスが魚より上にあれば上に、下にあれば下に移動させる。

```

If Y > Image3.Top Then
    Image3.Top = Image3.Top + 1
Elseif Y < Image3.Top Then
    Image3.Top = Image3.Top - 1

```

Image2.Visible = False (偽) 非表示  
 Image3.Visible = True (真) 表示