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 ～

岐阜県教育センター
林　　千　尋

＜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，ＶｉｓｕａｌＢａｓｉｃ（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

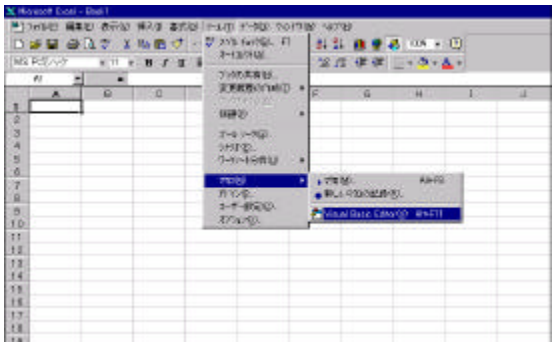
ＶＢＡによりプログラミングするためにＶｉｓｕａｌ Ｂａｓｉｃ Ｅｄｉｔｏｒ（ＶＢＥ）を起動し，基本操作を習得する。

（１）準備

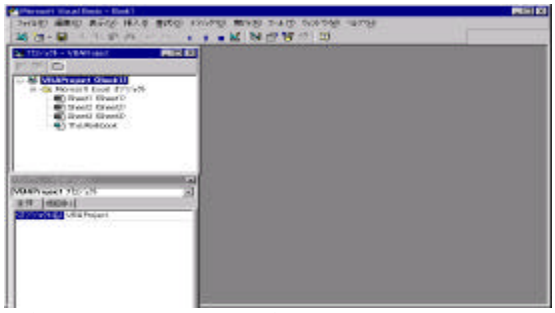
【操作】

①Ｅｘｃｅｌを起動する。

②Ｅｘｃｅｌメニューの
[ツール]-[マクロ]-[Visual Basic Editor]
を選択する。



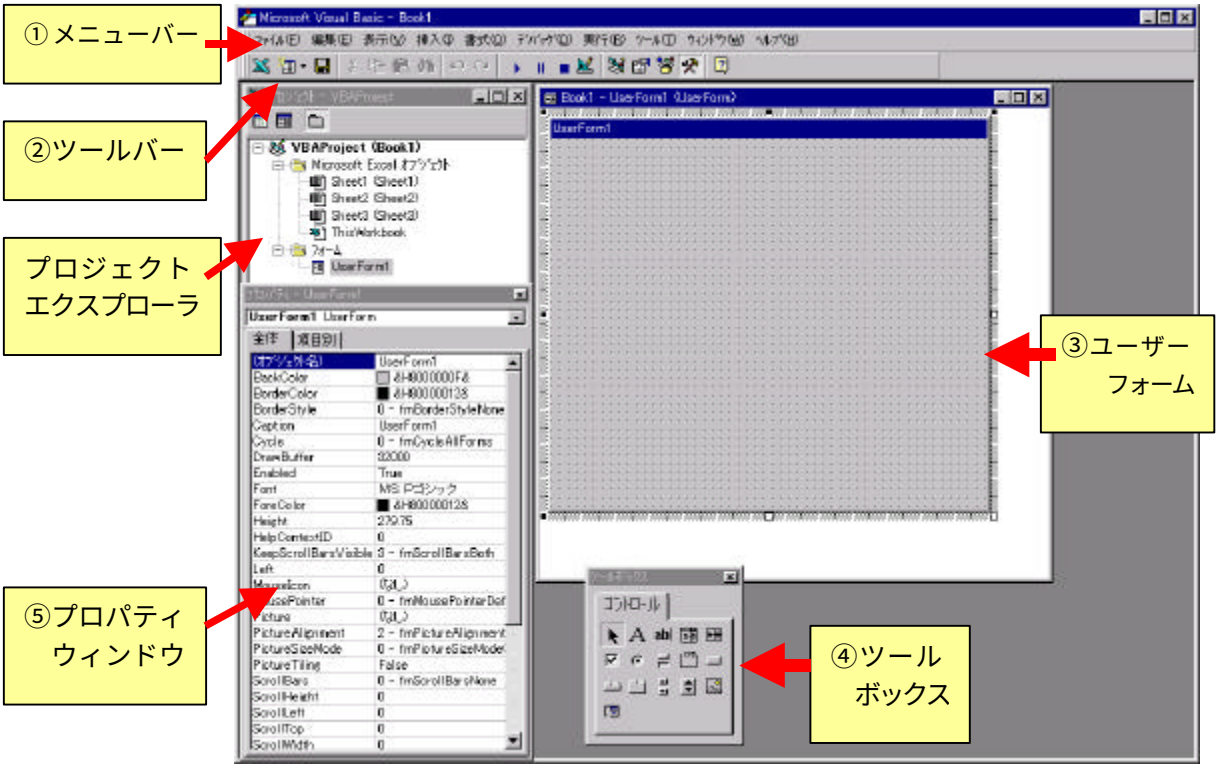
③ユーザーフォームの挿入
ＶＢＥメニューの
[挿入]-[ユーザーフォーム]を選択する。



（２）ＶＢＥの画面構成

①メニューバー
各種の操作コマンドを実行するメニュー

②ツールバー
よく使うコマンドをアイコン化したもの



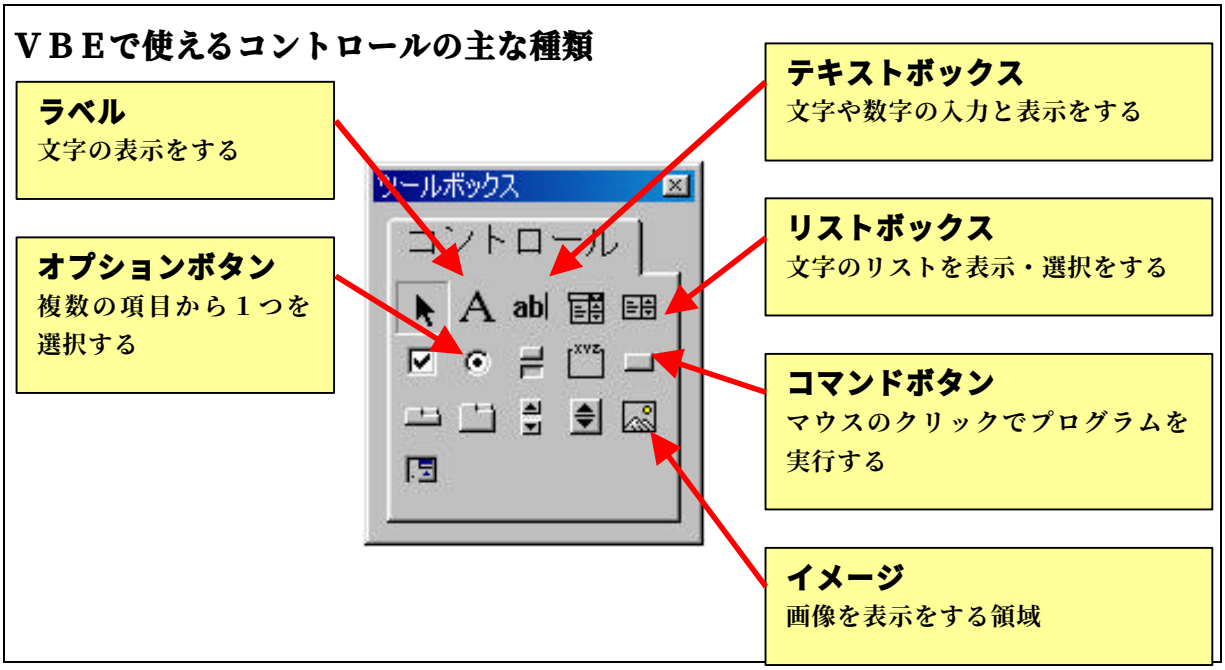
③ユーザーフォーム
プログラム作成の基本となる画面。
ここにプログラムの実行結果をイメージしながらコントロール配置し，プログラムを入力する。

④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。

⑤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。



３．プログラミングの基本操作

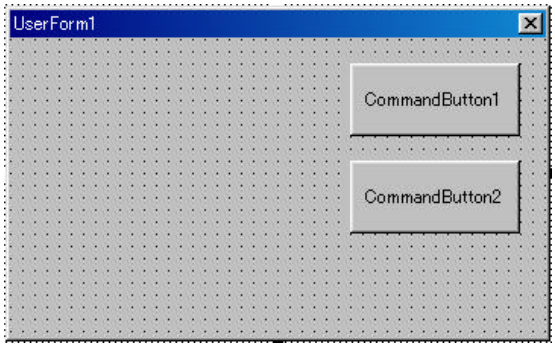
【練習課題１】コマンドボタンをクリックして背景の色を変える
～　ＲＧＢ関数で画面に表示される色の構成方法を理解しよう　～

（１）ユーザーフォームにコントロール配置

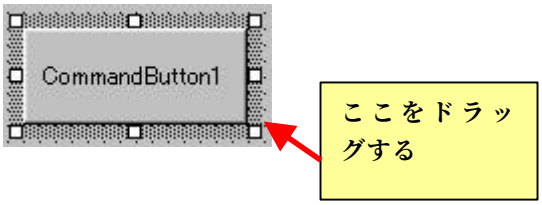
①ツールボックスからコマンドボタンをマウスでクリックし選択する。

②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］

③②と同様にもう一つコマンドボタンを配置する。［CommandButton2］

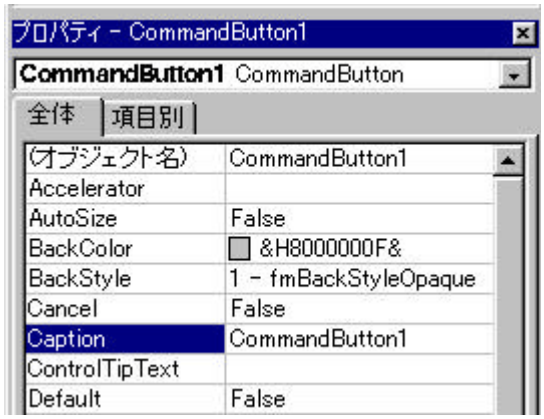


④マウスでコマンドボタンをクリックして選択状態にすると，大きさが変えられる。

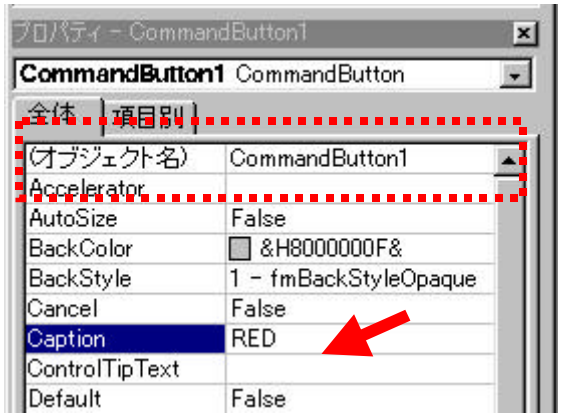


（２）プロパティ（属性）の変更

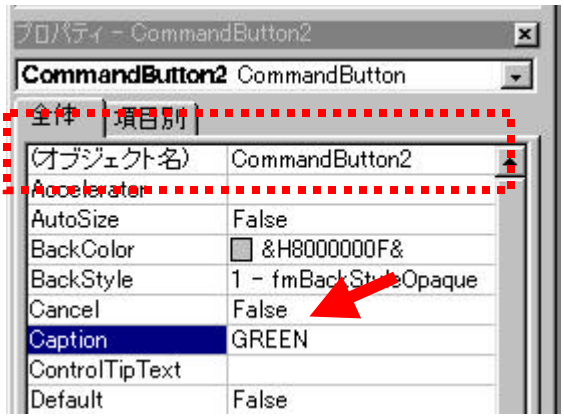
ユーザーフォームや配置したコマンドボタンをマウスで選択すると，プロパティウィンドウにそれぞれの属性が表示される。この内容を変更することにより名前や色，表示文字，性質などを変えることができる。



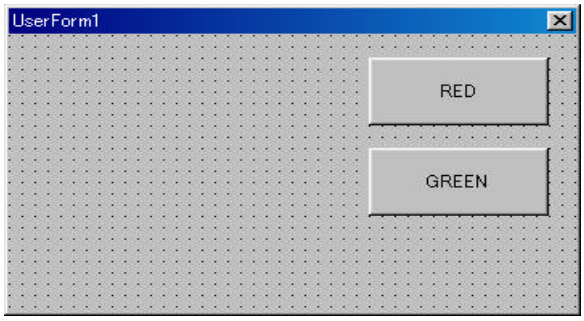
①コマンドボタンのプロパティのうち[Caption]欄の[CommandButton1]を[RED]に変更する。コマンドボタンに表示される文字が変わることを確認する。



②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。



③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。

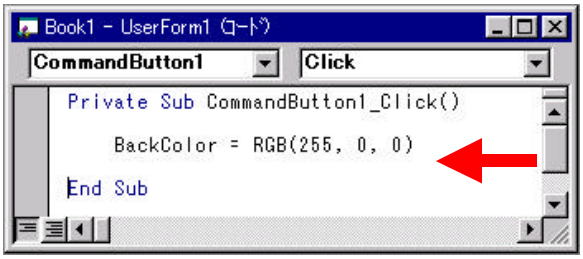


（３）プログラムコード（命令）の記述

①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)

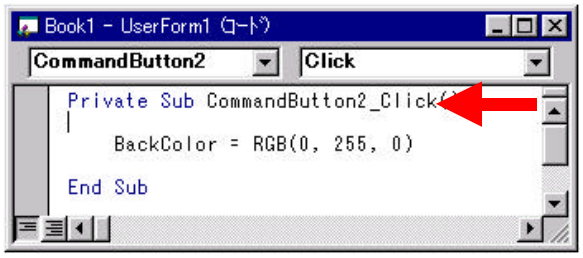
と入力する。



②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。

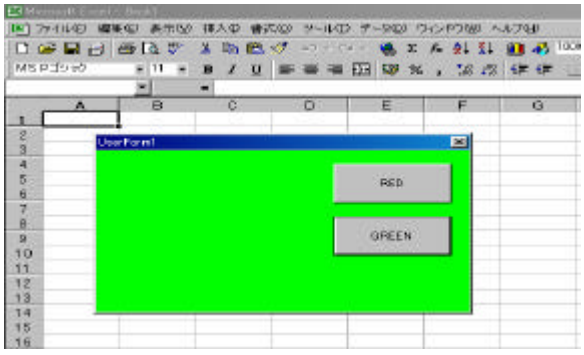


③これで必要なプログラム入力が終わったので，コードの入力画面を閉じる

（４）プログラムの実行

①メニューバー[実行]-[ﾕｰｻﾞｰﾌｫｰﾑの実行]またはツールバーの ▶ をクリックする。

②実行画面が表示される



③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。

④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

①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（コントロール名）　イベント名

End Sub ← プロシージャの終わり

②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

③ＲＧＢ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？

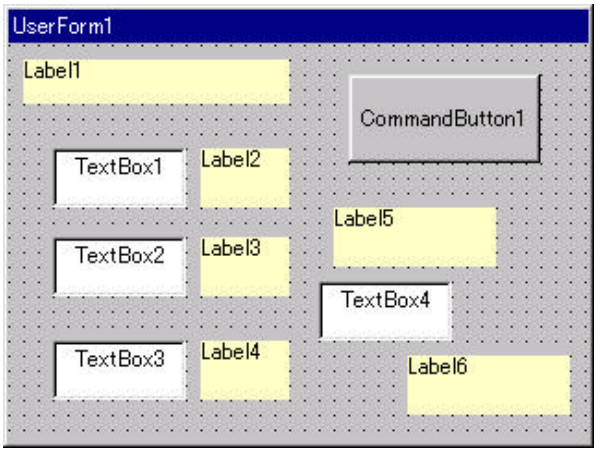
【練習問題１－１】 ＲＧＢ関数の各値を変えて，プログラムを実行してみなさい。

【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。

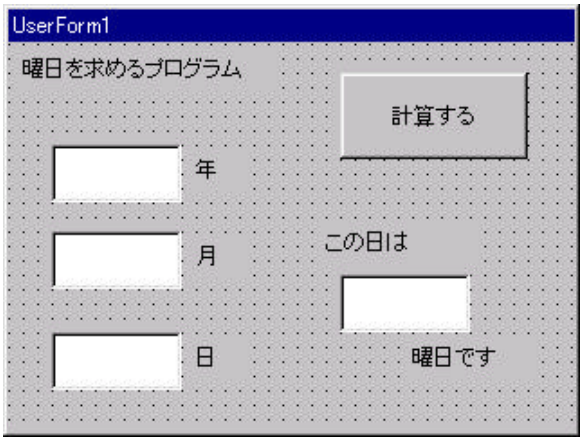


(注:一部説明のためにプロパティが変更された画面です)

（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

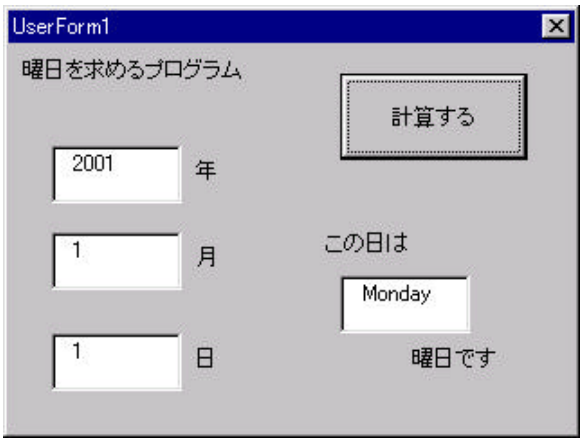
コマンドボタンをダブルクリックし，コード記述画面を出して，次のようにコードを記述する。

```
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

①メニューバー[実行]-[ﾕｰｻﾞｰﾌｫｰﾑの実行]
　またはツールバーの ▶ をクリックする。

②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。



③「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日にするとずれが生じるため，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

①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。

②西暦が４００で割切れる年も閏年とする。

③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)\4 - (y-1)\100 + (y-1)\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) Mod 7
```

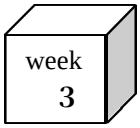
week=0 を日曜日，week=1 を月曜日と考える。

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。

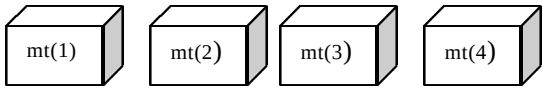


変数につける名前を変数名といい，変数名には英数文字を用い，必ずアルファベットから始まる名前とする。

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）

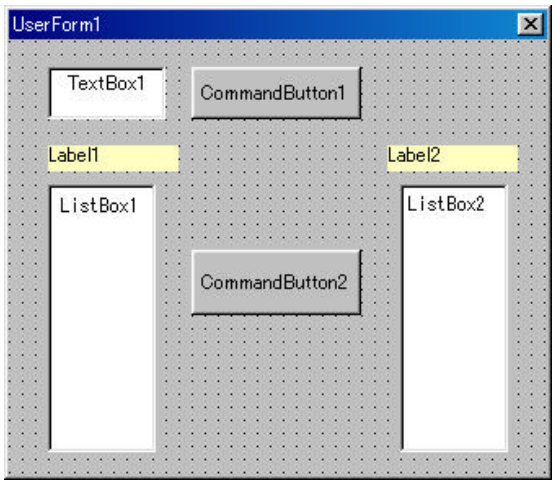
条件を判断して，それぞれの条件に応じたプログラムを実行するには，次のような形で記述する。

```
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。



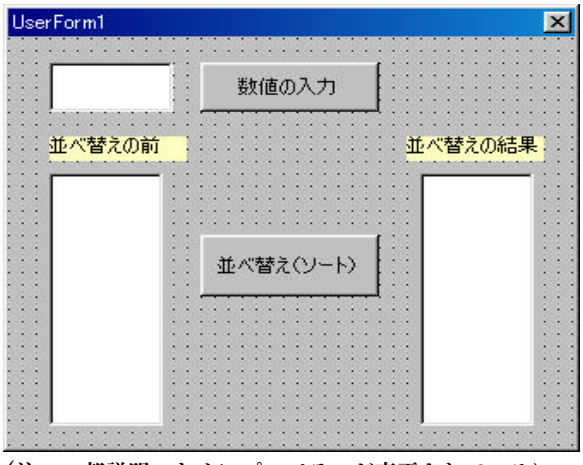
(注：一部説明のためにプロパティが変更されている)

（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。

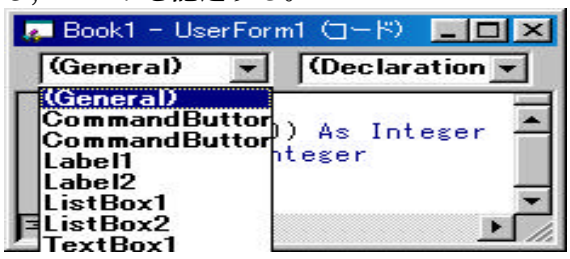
```
Dim kazu(50) As Integer
Dim n As Integer
```

```
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
```

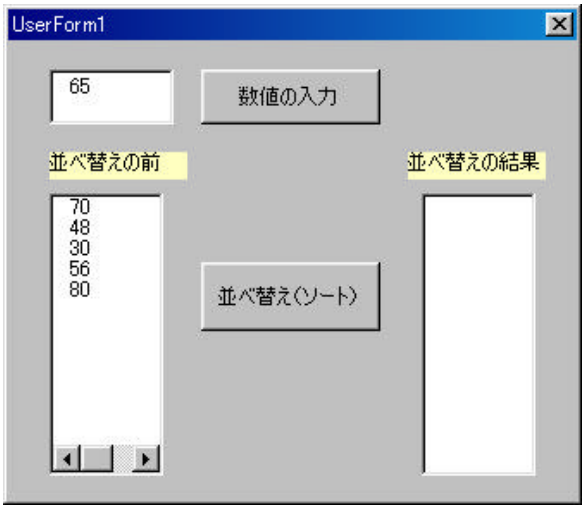
上記のプログラムコードの上二行は，図のようにコマンド名の部分を（General）に変更し，コードを記述する。



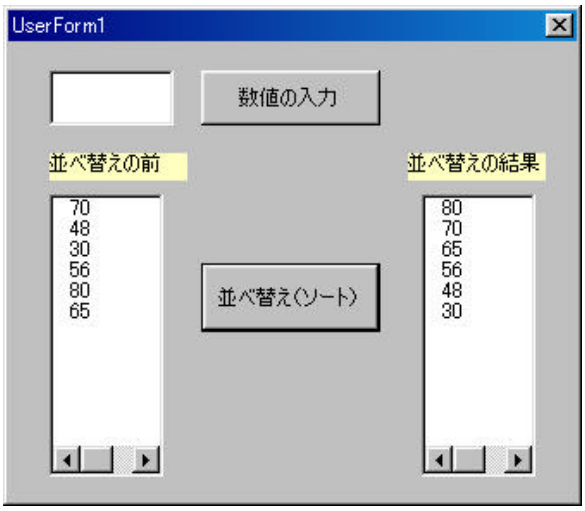
（４）プログラムの実行

①メニューバー[実行]-[ﾕｰｻﾞｰﾌｫｰﾑの実行]
またはツールバーの ▶ をクリックする。

②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。



③[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。



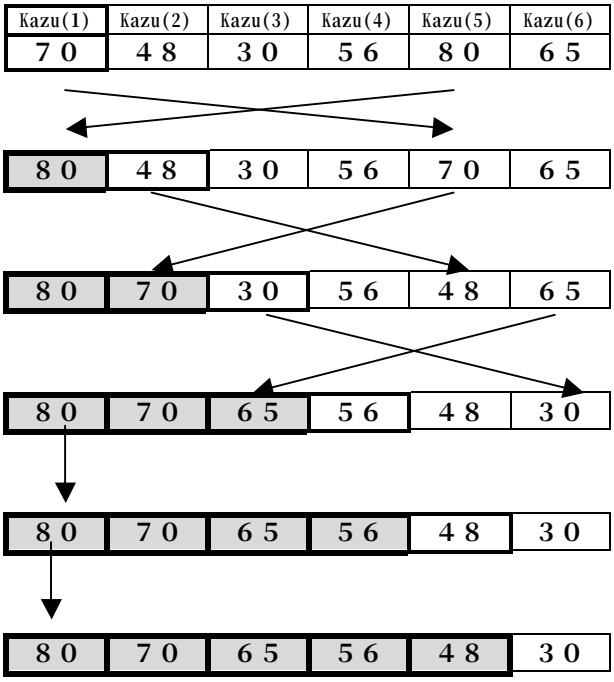
（５）プログラムの解説

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。

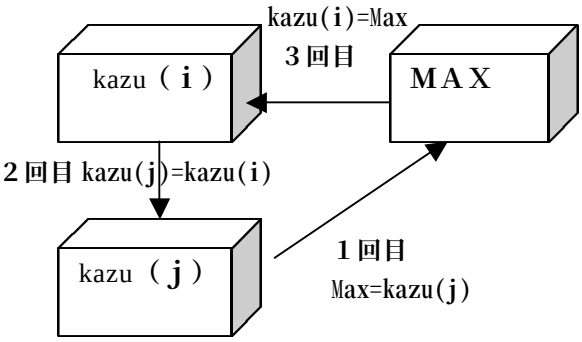
①配列変数 kazu(1)に代入された値を基準に，右側にある数値の中から最大のものを見つけ，データを入れ替える。これで，最大値が確定する。

②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。

③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。



④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。

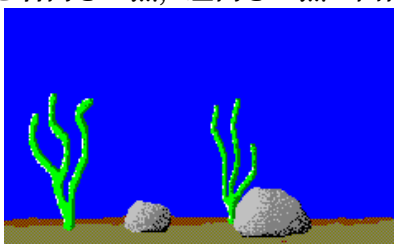
例
```
For k = 1 To n
    ListBox2.AddItem kazu(k), k - 1
Next k
```

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

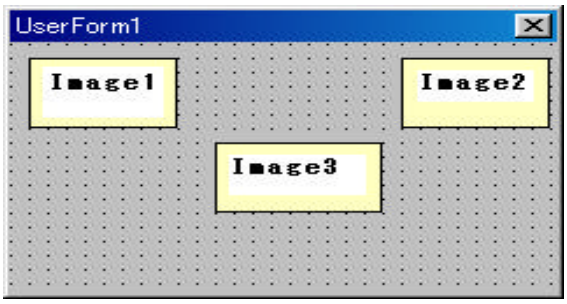
Fish_r.GIF

Fish_l.GIF

魚の絵は，透過ＧＩＦ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

①ユーザーフォームに水槽の絵を表示する。

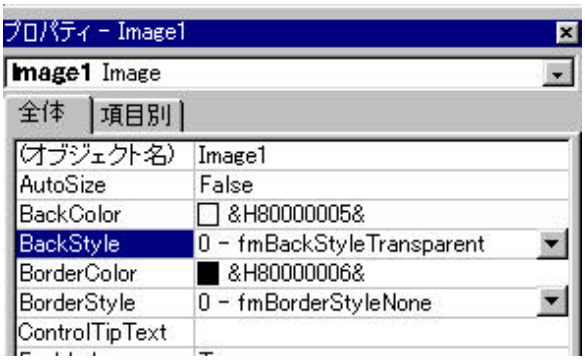
Picture プロパティを選択する。

作成した水槽のファイル名を選択する。

②Image1 コントロールに右向きの魚を表示

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。

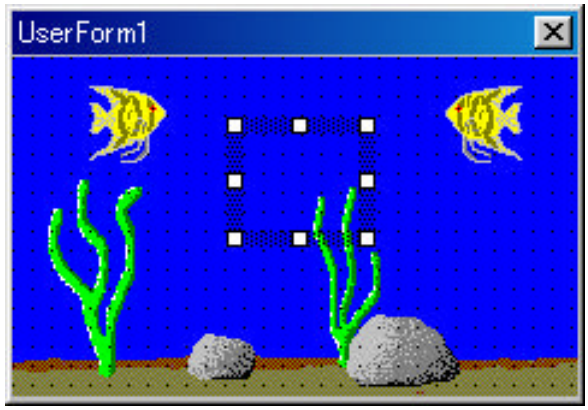
その他，バックスタイルおよびボーダースタイルのプロパティを次のように変更する。この変更は Image1～Image3 のコントロールすべてに設定する。



③Image2 コントロールに左向きの魚を表示

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。

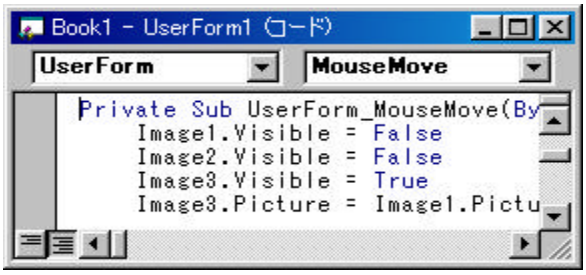
④次のような画面になることを確認する。



（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。

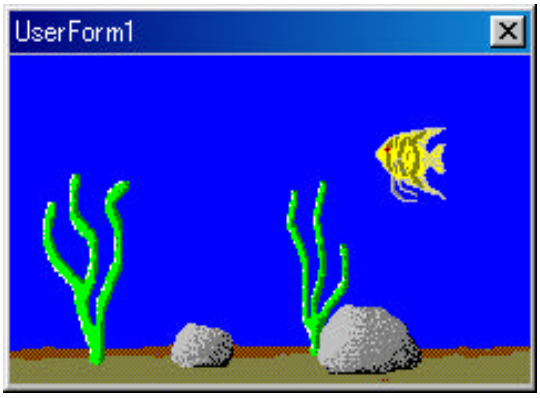



```
Private Sub UserForm_MouseMove(ByVal
  Button As Integer, ByVal Shift As
  Integer, ByVal X As Single, ByVal Y As
  Single)
    Image1.Visible = False
    Image2.Visible = False
    Image3.Visible = True
    Image3.Picture = Image1.Picture
    If X > Image3.Left Then
        Image3.Picture = Image1.Picture
        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

①メニューバー[実行]-[ﾕｰｻﾞｰﾌｫｰﾑの実行]
　またはツールバーの ▶ をクリックする。

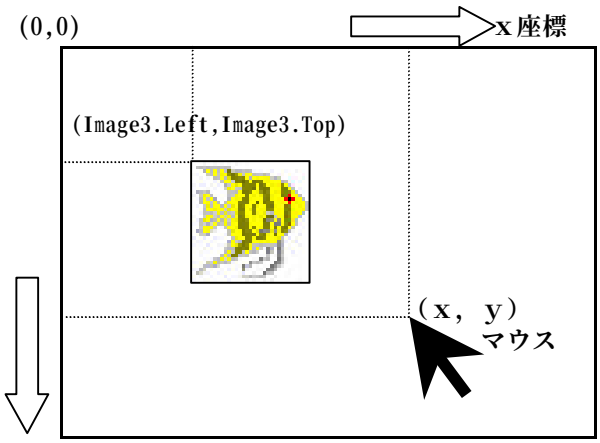
②実行画面



③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



①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。

②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。

③マウスが魚より右側にあれば、Image3 に右向きの魚の画像(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
```

④マウスが魚より左側にあれば、Image3 に左向きの魚の画像(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
```

⑤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
```