

はじめてみよう！

*Visual Basic*

平成12年3月21日  
北陸工業専門学校

# もくじ

1. <i>Visual Basic</i> ってなに？	...	3
2. まず、さわってみよう！		
2-1. <i>Visual Basic</i> の画面と名称	...	4
2-2. 文字を表示してみよう	...	5
2-3. ボタンを押すと英語が表示されるようにしよう	...	8
3. いろいろな機能を使ってみよう		
3-1. 絵を表示して変化するようにしよう	...	11
3-2. I F 文を使って絵を交互に変化させよう	...	13
3-3. タイマーを使って絵を自動的に変化させよう	...	14
3-4. 計算させてみよう	...	16
3-5. オプションボタンを使って処理を選択しよう	...	18
4. ゲームを作ってみよう		
4-1. ゲームの概要	...	21
4-2. 毎回違う文字を表示させよう	...	22
4-3. 得点を表示させよう	...	24
4-4. 時間がきたら GAME OVER と表示させよう	...	27
5. ゲームに機能を追加してみよう		
5-1. 文字サイズが変わるようにしよう	...	31
5-2. 文字の表示位置が変わるようにしよう	...	32
5-3. 当たったら音を鳴らそう	...	33
5-4. まとめ	...	34

# 1. Visual Basic ってなに？

Visual Basic とは、Microsoft社が開発した、Windows上で実行できるプログラムを開発するための言語です。

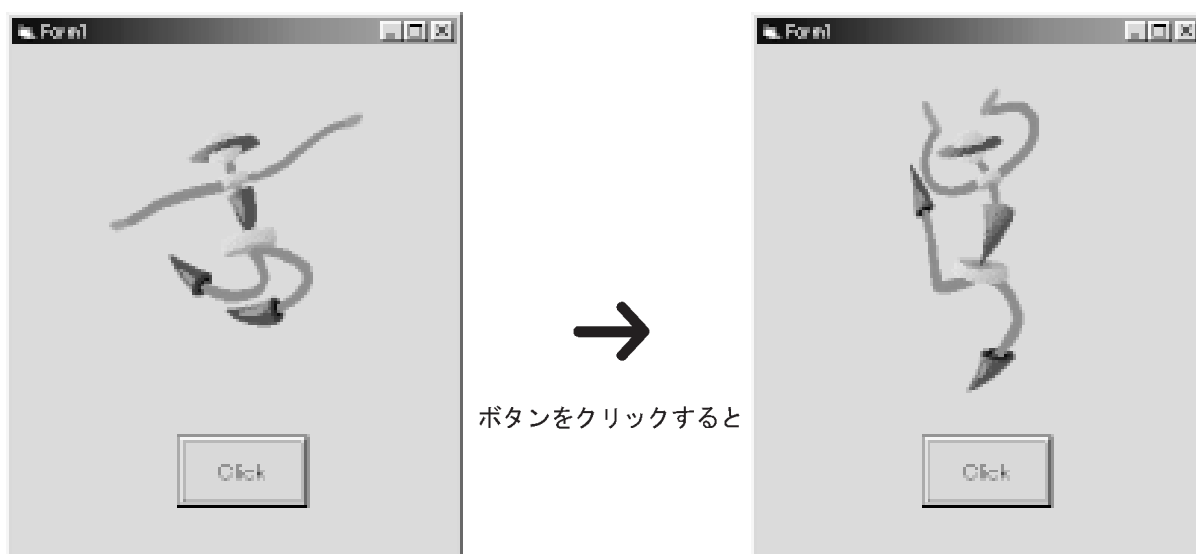
皆さんが使っている Windows ソフト(ワープロ、ゲームなど)の画面には、いろいろなボタンが付いています。そのボタンをマウスでクリックするといろいろな動作が起こります。それは、ボタンごとに「クリックしたらこういう動作をなさい」というプログラムが書いてあるからです。

この、「ボタンが押される」とか「マウスがクリックされる」などの事象が発生することを「**イベントが発生する**」といいます。

Visual Basic は、プログラムをこのイベントごとに作成していきます。

これだけ説明すると、難しそうに聞こえるかも知れませんが、Basic でプログラムを組んだ経験のある人には、とても簡単に理解することができます。また、プログラミングを経験したことがない人でも、容易に理解できるようになります。

とにかくまずはさわってみましょう！



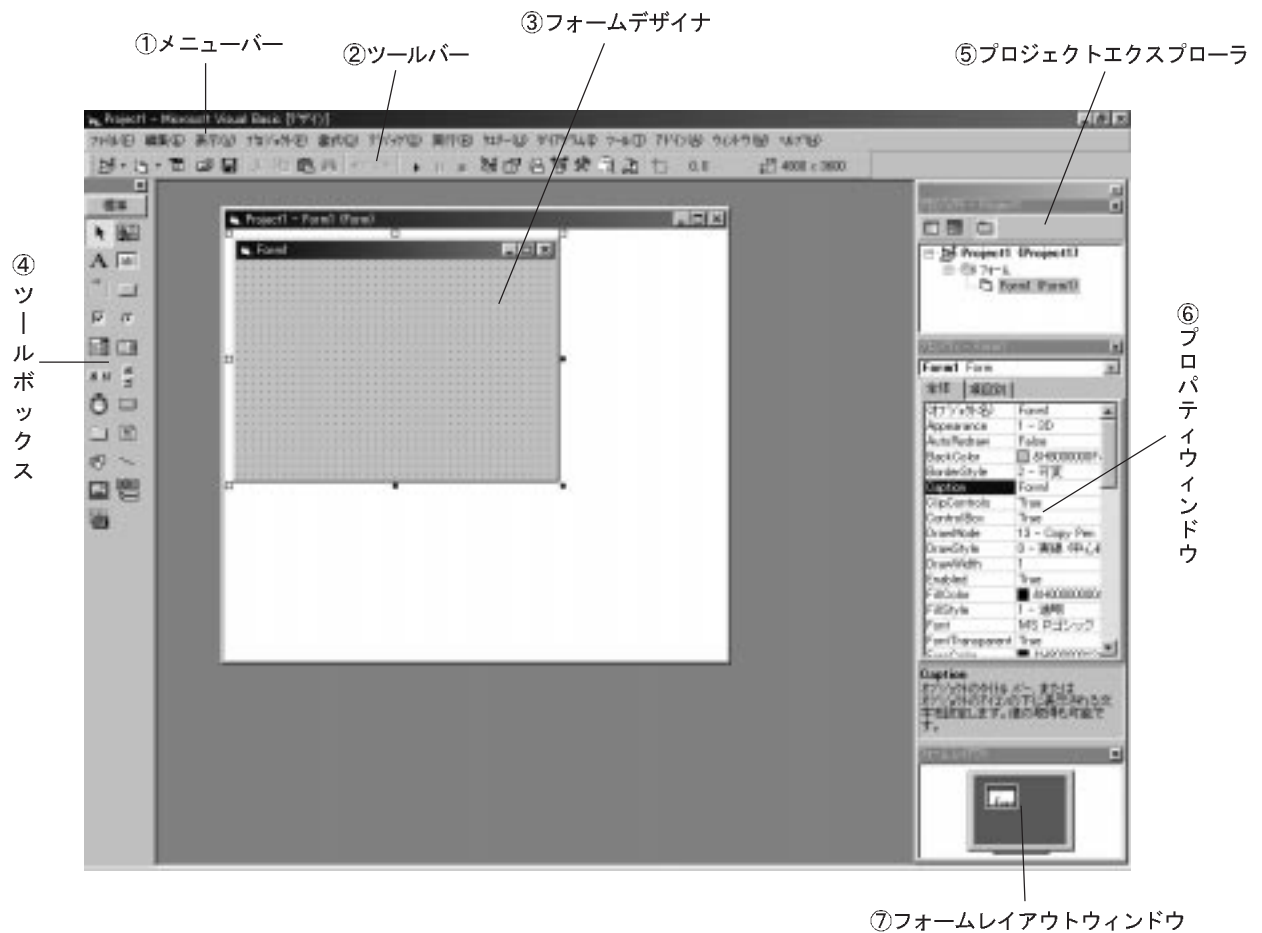
上の処理は下記の4行のプログラムでできています！

```
Private Sub Command1_Click()  
Image1.Visible = False  
Image2.Visible = True  
End Sub
```

## 2. まず、さわってみよう！

### 2-1. Visual Basicの画面と名称

Visual Basic を立ち上げると、次のような画面が表示されます。



各ウィンドウの名称と機能は次のとおりです。

#### ①メニューバー

ファイルの読み込みや保存、ウィンドウの表示変更などを行うときに利用します。

#### ②ツールバー

メニューバーの中にある命令(コマンド)で使用頻度の高いものをボタンに割り付けてあります。

#### ③フォームデザイナー

実際に実行(表示)される画面を設計するウィンドウです。

#### ④ツールボックス

ボタン、テキストボックス、図形などのツールを選択してフォームに貼り付けます。

#### ⑤プロジェクトエクスプローラ

プログラムの中で使用されるファイルの一覧を表示します。

#### ⑥プロパティウィンドウ

フォームの色や文字のフォント、ボタンの表示メッセージなど、各種の初期設定内容が表示されています。設定を変更することもできます。

#### ⑦フォームレイアウトウィンドウ

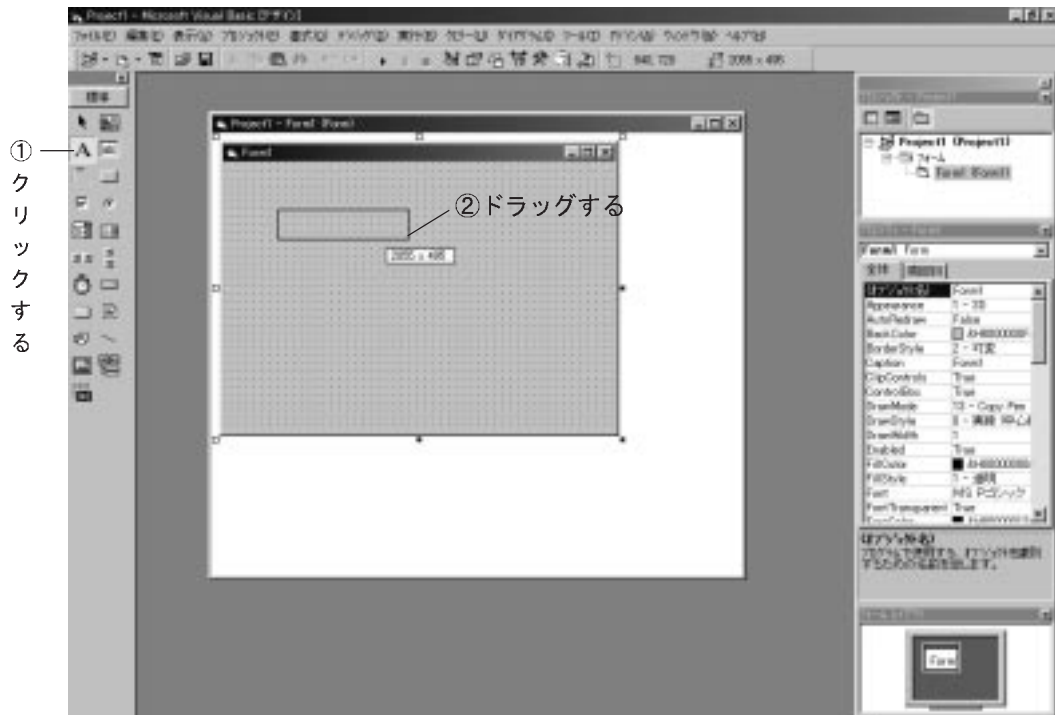
作成したプログラムを実行したときにフォーム(ウィンドウ)をどこに表示するかを指定します。

## 2-2. 文字を表示してみよう

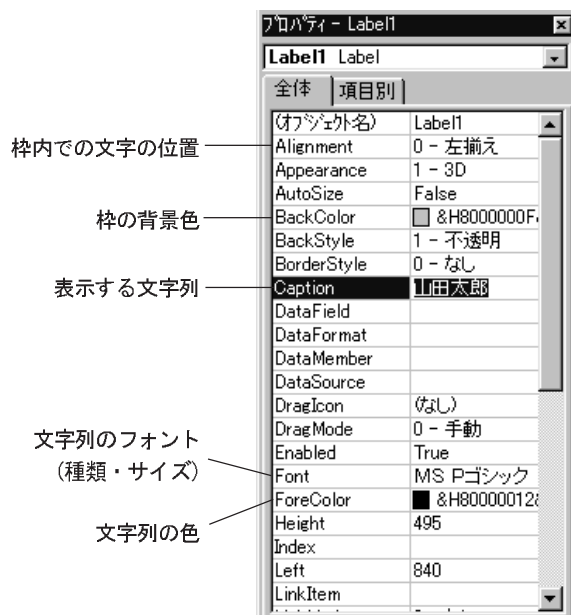
ではさっそく作ってみましょう。

まず、あなたの名前を表示することにします。

- ① 左側のツールボックスのラベルボタン **A** をクリックします。
- ② フォームデザイナーの中で、名前を表示したい範囲をマウスでドラッグします。



- ③ プロパティウィンドウが「Label1」になっていることを確認して、表示したい文字列や色を設定します。




- ④ 同じようにForm1 の中でマウスを1度クリックして、プロパティウィンドウが「Form1」になっていることを確認して、BackColor を設定します。

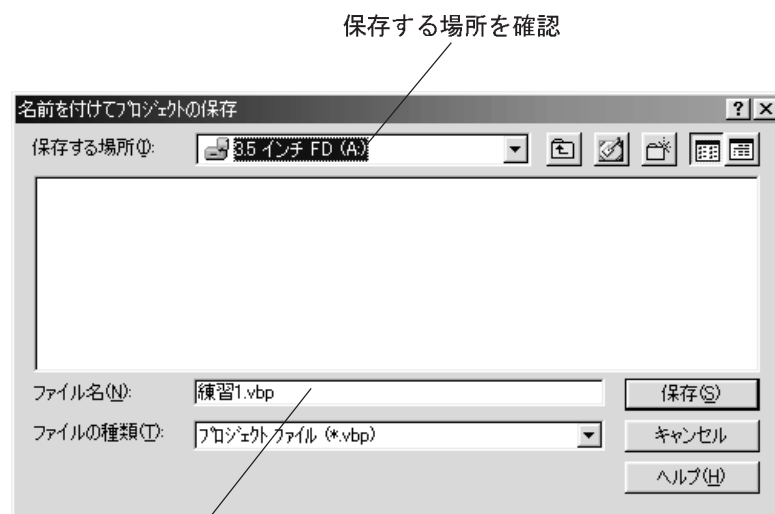


- ⑤ ①～③を繰り返して、住所、趣味を表示してみましょう。
- ⑥ また、「氏名」、「住所」、「趣味」といった項目名も表示してみましょう。  
これらは全て Label で設定します。



### 出来上がり例



- ⑦ ファイルを保存します。
- メニューバーの **ファイル** - **プロジェクトの保存** を選択するか、ツールバーの  ボタンを押して、フォームとプロジェクトを保存します。
- フォームは画面ごとの設定です。  
プロジェクトはプログラム(システム)全体の設定です。
- 「保存する場所」が「3.5インチFD」になっていることを確認して、自由にファイル名を設定して保存してください。



ファイル名を入力

- ⑧ 作成した画面を実行してみます。  
ツールバーの開始ボタン  を押します。  
プログラムが実行されます。  
プログラムを停止する時は終了ボタン  を押します。

### 実行画面




The screenshot shows a window titled "Form1" with a white background. It contains three rows of text, each with a label on the left and a value in a light gray box on the right:

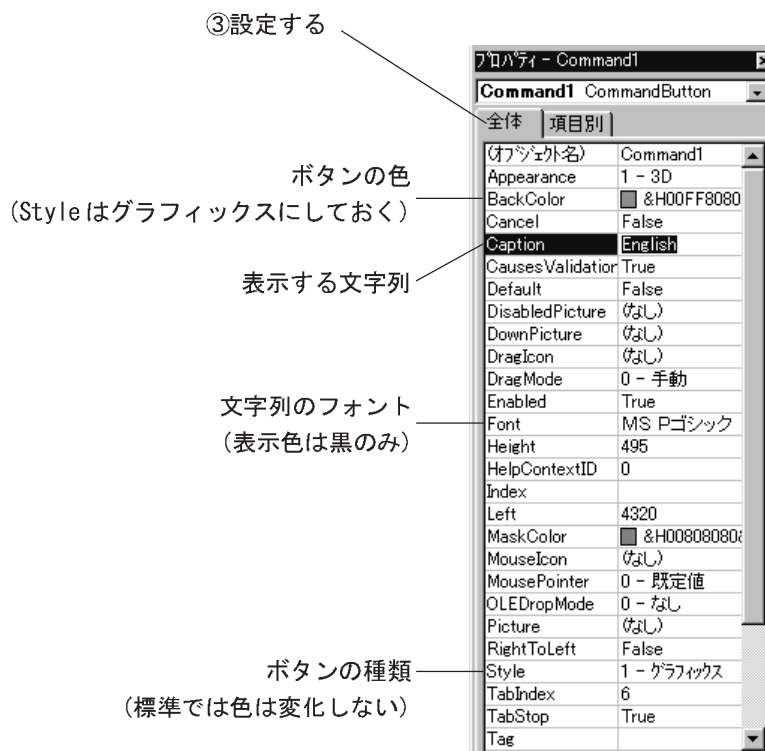
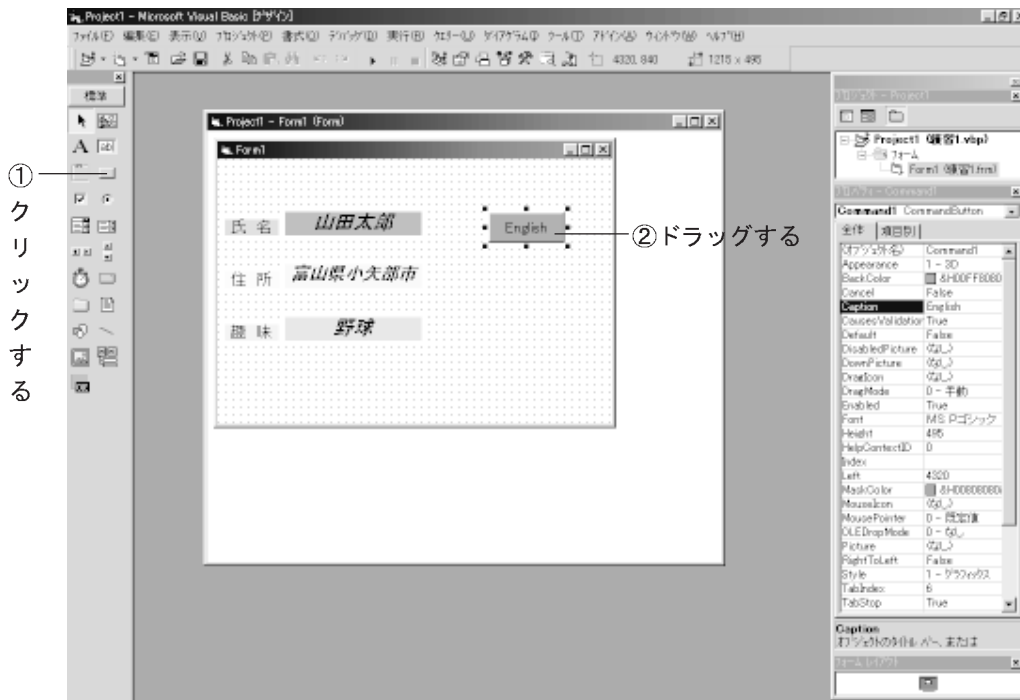
- 氏名 (Name): 山田太郎 (Yamada Taro)
- 住所 (Address): 富山県小矢部市 (Toyama Prefecture, Oyabe City)
- 趣味 (Hobby): 野球 (Baseball)

これだけでは、文字を配置しただけですから、実行しても何も変化しません。  
では次に、プログラムを作成して、画面に変化が起きるようにしましょう。

## 2-3. ボタンを押すと英語が表示されるようにしよう

ボタンを押すと、氏名がローマ字表示になりますようにします。

- ① ツールボックスのコマンドボタン  をクリックします。
- ② Label と同じように、フォームデザイナーの中で、ボタンを配置したい範囲をマウスでドラッグします。
- ③ プロパティウィンドウが「Command1」になっていることを確認して、ボタンの上に表示したい文字列やボタンの色などを設定します。





- ④ 作成したボタンをダブルクリックして、コードエディタウィンドウを開きます。

### コードエディタウィンドウ



- ⑤ 次のようにプログラムを入力します。




プログラムの意味は次のとおりです。

<code>Private Sub Command1_Click()</code>	...	Command1 のボタンがクリックされたときに実行されるサブプログラム
<code>Label1.Caption = "Taro Yamada"</code>	...	Label1 の Caption に "Taro Yamada" を代入
<code>End Sub</code>	...	サブプログラムの終了

Visual Basic は、このような小さいサブプログラムを組み合わせて構成します。また、「Label1.Caption」のように、プロパティを変数の様に扱い、設定値をプログラムで変更できます。

- ⑥ プログラムを実行します。  
前回と同じように ▶ をクリックします。
- ⑦ English ボタンをクリックすると名前がローマ字に変わります。



- ⑧ ■ ボタンをクリックするとプログラムが終了します。
- ⑨ ツールバーの上書き保存ボタン  をクリックし、フォームとプロジェクトを保存します。

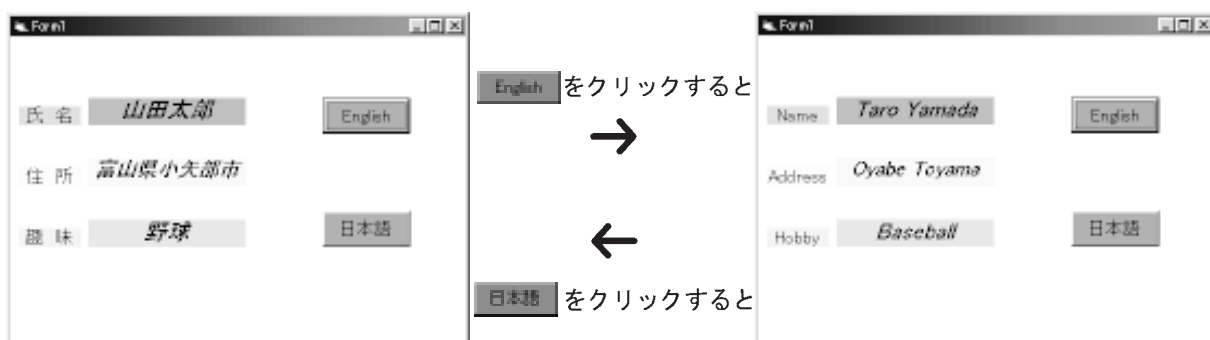
### 【練習問題1】

プログラムを追加して、次のように動作させてみましょう。



### 【練習問題2】


さらにボタンとプログラムを追加して、次のように動作させてみましょう。



### 3. いろいろな機能を使ってみよう

#### 3-1. 絵を表示して変化するようにしよう


絵(画像)を表示させて、ボタンを押すと表示されている絵が変わるようにしましょう。

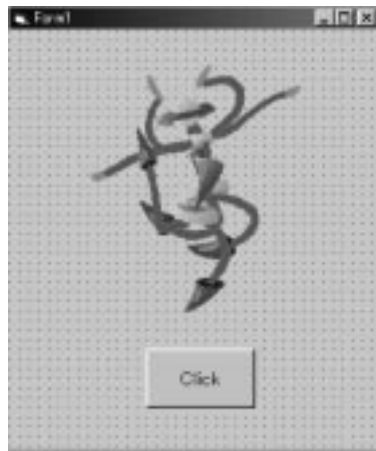
- ① フロッピーディスクに表示したい絵を準備(作成)します。
- ② メニューバーの **ファイル** - **新しいプロジェクト** を選択します。
- ③ フォームのサイズ、背景色を指定します。
- ④ ツールボックスの左下にあるイメージボタン  をクリックします。
- ⑤ フォームデザイナーの中で、絵を配置したい範囲をマウスでドラッグします。
- ⑥ プロパティウィンドウが「Image1」になっていることを確認します。
- ⑦ Picture の右側をクリックして、最初に表示したい絵のファイル名を指定します。



- ⑧ ボタンをクリックした後に表示する絵を「Image2」として同じ場所に配置します。
- ⑨ 「Image2」のプロパティの「Visible」(目に見える)を **False**(偽)にします。  
これで、「Image2」の絵はプログラムを実行したときには表示されません。



- ⑩ ツールボックスの Command ボタン  をクリックします。
- ⑪ フォームデザイナーの中で、ボタンを配置したい範囲をマウスでドラッグします。
- ⑫ プロパティウィンドウが「Command1」になっていることを確認して、ボタンの上に表示したい文字列やボタンの色などを設定します。



- ⑬ 作成したボタンをダブルクリックして、コードエディタウィンドウを開きます。
- ⑭ 次のようにプログラムを入力します。

```

Private Sub Command1_Click()
    Image1.Visible = False
    Image2.Visible = True
End Sub

```

ボタンをダブルクリックすると自動的に表示される

ボタンが押されると Image1 の表示をやめる

Image2 を表示する

- ⑮ プログラムを実行し、 ボタンを押すと、絵が変わります。



- ⑯ フォームとプロジェクトを新しい名前前でフロッピーディスクに保存します。

## 3-2. IF文を使って絵を交互に変化させよう

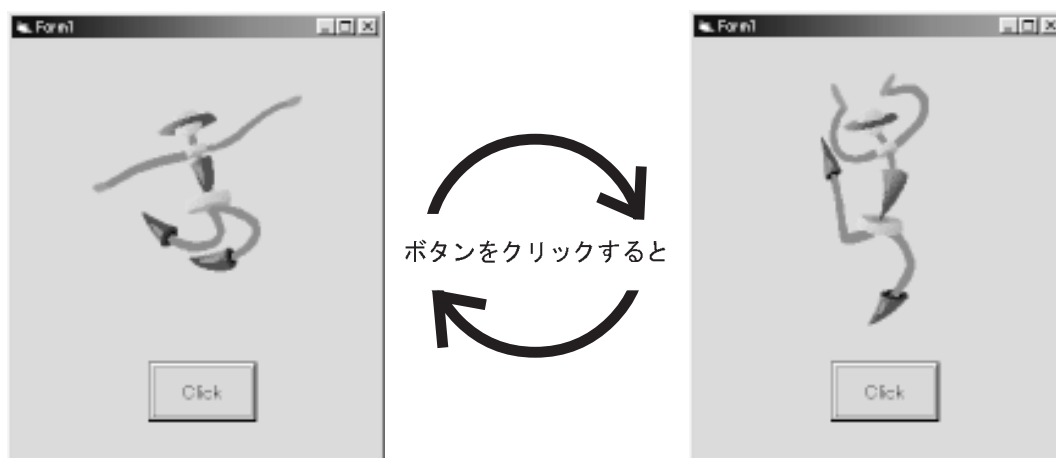
3-1のプログラムを元にして、ボタンを押すと絵が交互に表示されるようにしましょう。

① 次のようにプログラムを改良します。

```
Private Sub Command1_Click()  
    If Image1.Visible = True Then  
        Image1.Visible = False  
        Image2.Visible = True  
    Else  
        Image1.Visible = True  
        Image2.Visible = False  
    End If  
End Sub
```

もし、Image1が表示されていれば  
Image1を表示しない  
Image2を表示する  
そうでないならば  
Image1を表示する  
Image2を表示しない

② プログラムを実行すると、ボタンをクリックするたびに絵が交互に表示されます。



### 3-3. タイマーを使って絵を自動的に変化させよう

3-2のプログラムを元にして、ボタンを押さなくても、自動的に絵が交互に表示されるようにしましょう。

- ① メニューバーの **ファイル** - **新しいプロジェクト** を選択します。
- ② 前回と同じように、変化させたい絵を「Image1」「Image2」として、フォームに貼り付けます。
- ③ 「Image2」のプロパティのVisibleをFalseにしておきます。
- ④ ツールボックスのタイマーボタン  をクリックし、フォーム内のどこかに貼り付けます(実行すると見えなくなりますから、どこに貼り付けてもかまいません。)
- ⑤ 貼り付けたタイマーボタンをダブルクリックして、コードエディタウィンドウを開きます。
- ⑥ 次のようにプログラムを入力します。

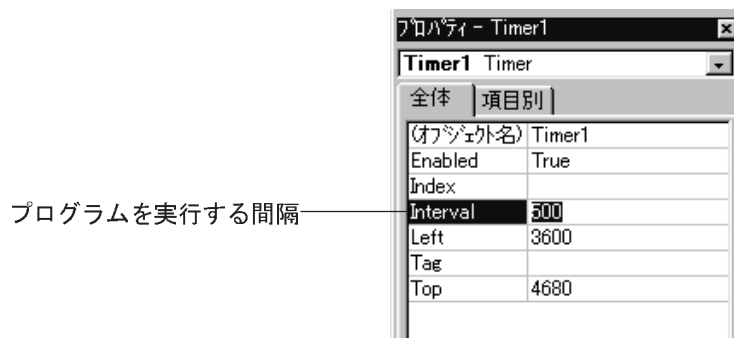
```
Private Sub Timer1_Timer ()  
    If Image1.Visible = True Then  
        Image1.Visible = False  
        Image2.Visible = True  
    Else  
        Image1.Visible = True  
        Image2.Visible = False  
    End If  
End Sub
```

Timer1 の指定した時間ごとの処理

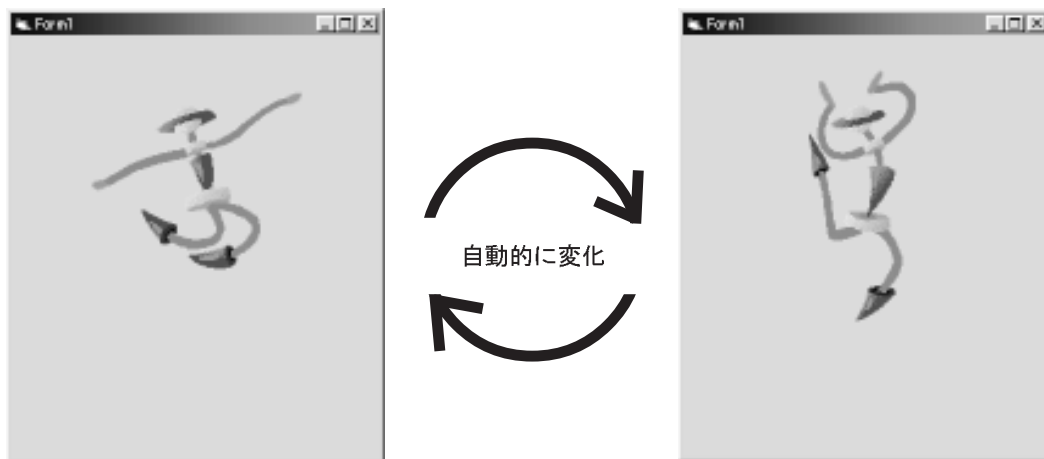
3-2のプログラムと同じ

サブプログラムの終了

- ⑦ 上記のプログラムを何秒ごとに実行させるかを指定します。  
「Timer1」のプロパティウィンドウのIntervalを設定します。  
1000を入力すると、1秒ごとに上記プログラムを繰り返し実行します。  
500を入力すると、0.5秒ごとになります。  
動作させたい周期を自由に設定します。



- ⑧ プログラムを実行すると、自動的に指定した間隔で絵が交互に表示されます。

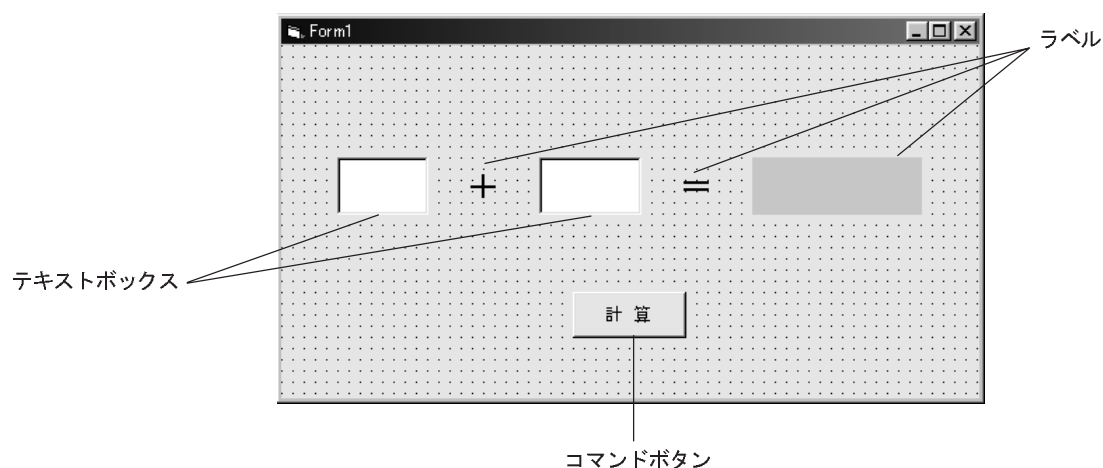


- ⑨ フォームとプロジェクトを新しい名前でフロッピーディスクに保存します。

### 3-4. 計算させてみよう

画面に2つの数字を入力し、ボタンを押すと合計の計算ができるようにしましょう。

- ① メニューバーの **ファイル** - **新しいプロジェクト** を選択します。
- ② ツールボックスのテキストボックスボタン **lab1** をクリックし、フォームデザイナーの中で、テキストを配置したい範囲をマウスでドラッグします。
- ③ テキストボックス2つとラベル3つとコマンドボタン1つを下記のように配置し、各設定を行います。



#### ラベルとテキストボックスの違い

- ・ ラベルは主に文字を表示するときに使用
- ・ テキストボックスは主に文字を入力する時に使用
- ・ ラベルからキー入力できない
- ・ テキストボックスはキー入力も表示もできる



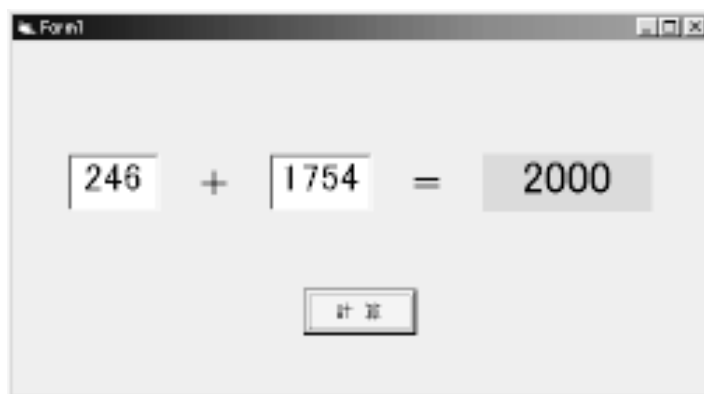
- ④ コマンドボタンをダブルクリックして、次のようにプログラムを入力します。

```
Private Sub Command1_Click()  
    Label3.Caption = Val(Text1.Text) + Val(Text2.Text)  
End Sub
```

Text1とText2を  
足して、  
Label1.Captionに  
代入する

答えを表示するラベル名      Text1の内容を数値化した値      Text2の内容を数値化した値


- ⑤ プログラムを実行し、テキストボックスに数値を入力して  ボタンをクリックすると、足し算の答えが表示されます。

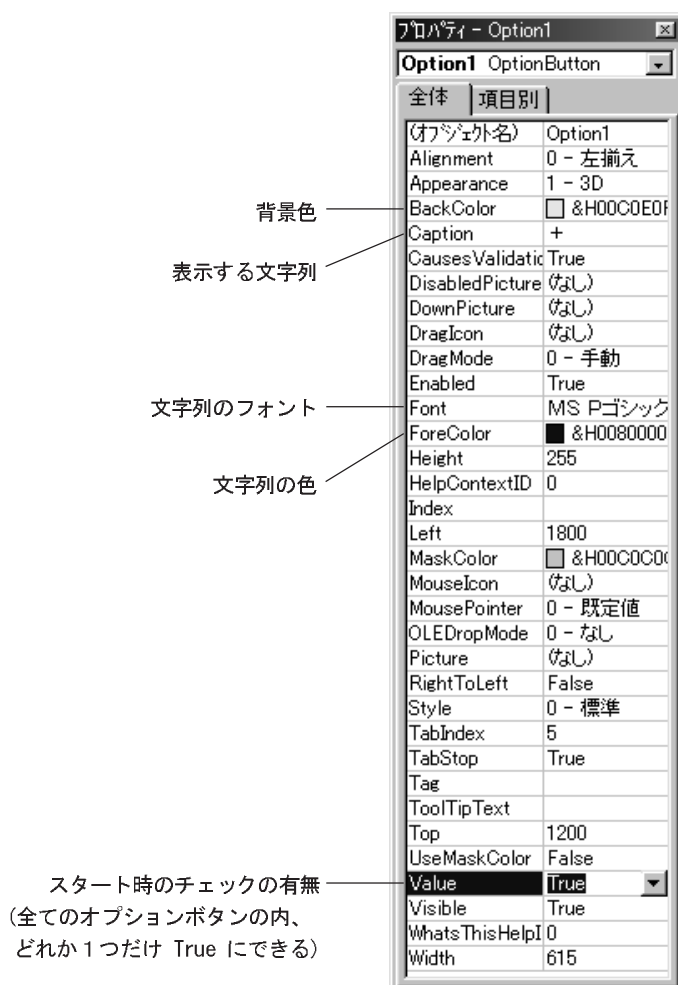
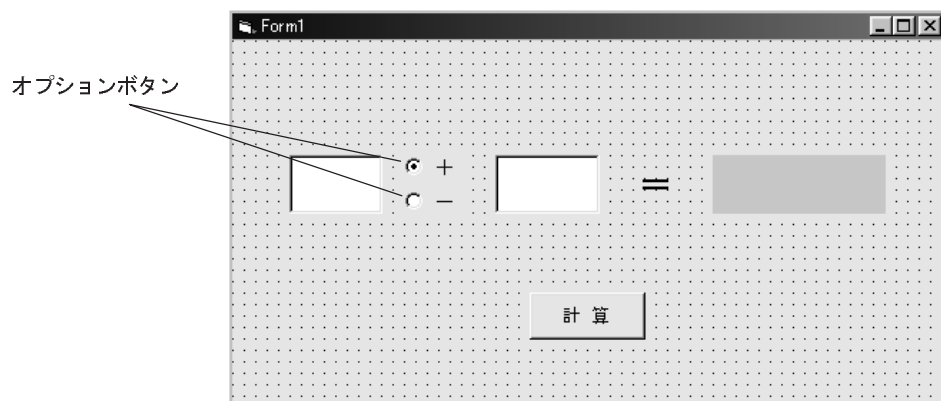


- ⑥ フォームとプロジェクトを新しい名前でもフロッピーディスクに保存します。

### 3-5. オプションボタンを使って処理を選択しよう

3-4のフォームとプログラムを元にして、足し算か引き算を選べるようにしましょう。

- ① 「+」を表示したラベルを選択し、削除する。
- ② ツールボックスのオプションボタン  をクリックし、「+」のあった場所に配置する。
- ③ オプションボタン2つをプロパティを設定して、次のように並べる。



項目別	値
(オブジェクト名)	Option1
Alignment	0 - 左揃え
Appearance	1 - 3D
BackColor	<input type="color" value="#H00C0E0F"/>
Caption	+
CausesValidatic	True
DisabledPicture	(なし)
DownPicture	(なし)
DragIcon	(なし)
DragMode	0 - 手動
Enabled	True
Font	MS Pゴシック
ForeColor	<input type="color" value="#H0080000"/>
Height	255
HelpContextID	0
Index	
Left	1800
MaskColor	<input type="color" value="#H00C0C00"/>
MouseIcon	(なし)
MousePointer	0 - 既定値
OLEDropMode	0 - なし
Picture	(なし)
RightToLeft	False
Style	0 - 標準
TabIndex	5
TabStop	True
Tag	
ToolTipText	
Top	1200
UseMaskColor	False
Value	<input checked="" type="checkbox"/>
Visible	True
WhatsThisHelpI	0
Width	615

スタート時のチェックの有無  
(全てのオプションボタンの内、  
どれか1つだけ True にできる)

- ④ コマンドボタンをダブルクリックして、次のようにプログラムを入力します。

```
Private Sub Command1_Click()
```

```
    If Option1.Value = True Then
```

```
        Label3.Caption = Val(Text1.Text) + Val(Text2.Text)
```

```
    ElseIf Option2.Value = True Then
```

```
        Label3.Caption = Val(Text1.Text) - Val(Text2.Text)
```

```
    End If
```

```
End Sub
```

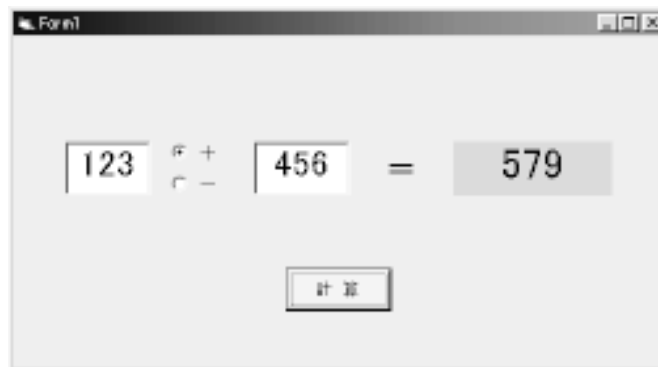
Option1 が選ば  
れていたたら

足し算の計算

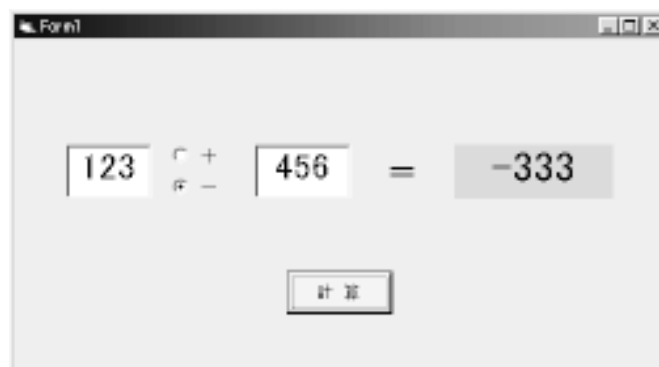
Option2 が選ば  
れていたたら

引き算の計算

- ⑤ プログラムを実行し、テキストボックスに数値を入力して「+」のオプションボタンを  
チェックして、 ボタンをクリックすると、足し算の答えが表示されます。



- ⑥ プログラムを実行し、テキストボックスに数値を入力して「-」のオプションボタンを  
チェックして、 ボタンをクリックすると、引き算の答えが表示されます。



- ⑦ フォームとプロジェクトを新しい名前でもフロッピーディスクに保存します。

### 【練習問題3】

「×」と「÷」のオプションボタンを追加して、四則演算ができるようにしましょう。

The screenshot shows a window titled "Form1" with a calculator interface. On the left, there is a text box containing the number "53". To its right is a vertical stack of four radio buttons, each with an operator: "+", "-", "x", and "÷". The "x" radio button is selected. To the right of the operators is another text box containing the number "4". Further right is an equals sign "=" and a text box containing the result "13.25". Below the operators and the "4" box is a button labeled "計算" (Calculate).

#### オプションボタン とチェックボックス の違い

- ・ オプションボタンは、選択肢の中からどれか1つを選ぶときに使用
- ・ チェックボックスは、選択肢の中から複数選択するときに使用

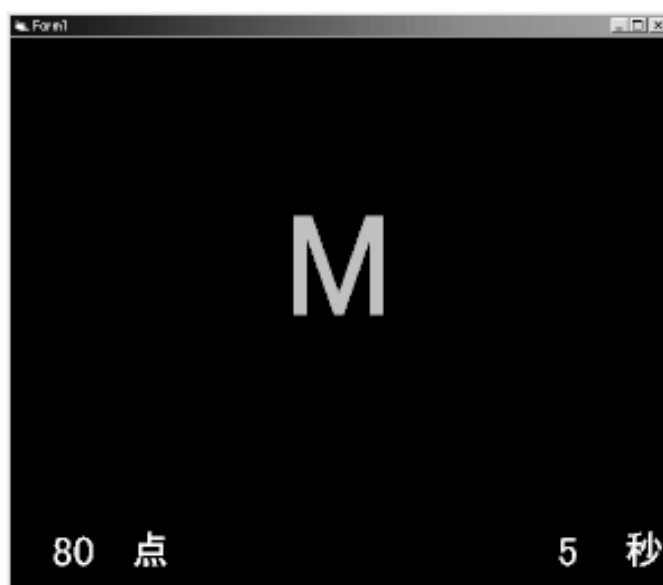
## 4. ゲームを作ってみよう

### 4-1. ゲームの概要

次のようなゲームを作ってみましょう。

- ① 画面上に、A～Zまでの文字が、1つ表示される。
- ② 表示された文字をキーボードから入力すると、得点が加算される。
- ③ ある時間になると「GAME OVER」と表示される。

プログラムをスタートさせると



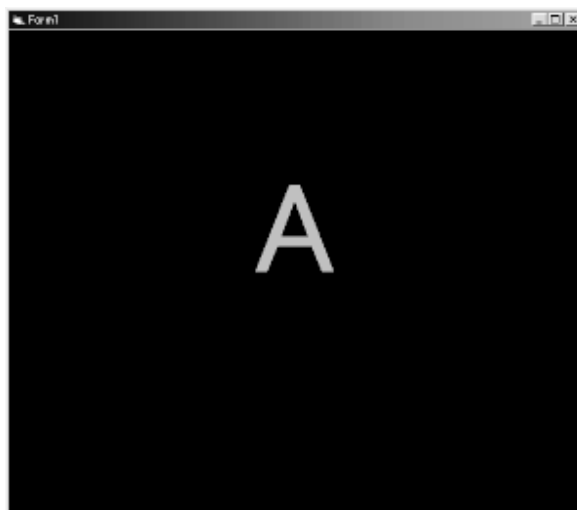
時間になると



## 4-2. 毎回違う文字を表示させよう

1秒ごとに毎回異なる文字が表示されるようにしましょう。

- ① メニューバーの **ファイル** - **新しいプロジェクト** を選択します。
- ② フォームのサイズ、色を設定します。
- ③ 文字を表示させるラベルを作成し、「A」と表示させておきます。



- ④ タイマーを1つ配置します。
- ⑤ 配置したタイマーをダブルクリックして下記のプログラムを入力します。

```
Private Sub Timer1_Timer ()  
    moji = Int(Rnd() * 26) + Asc("A")  
    Label1.Caption = Chr(moji)  
End Sub
```

文字コード計算用変数

“A”の文字コード

0~25までの整数の乱数

moji 番目の文字をラベル1に表示する

- ⑥ フォームの背景部分をダブルクリックして下記のプログラムを入力します。

```
Private Sub Form_Load ()  
    Randomize  
End Sub
```

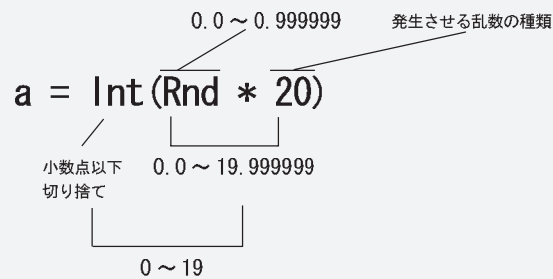
フォームが最初に表示されたときに実行される処理

乱数の初期化

- ⑦ Timer1のプロパティの Interval を 1000 に設定します。
- ⑧ プログラムを実行すると、画面上に異なる文字が1秒ごとに表示されます。

## 乱数の初期化と発生

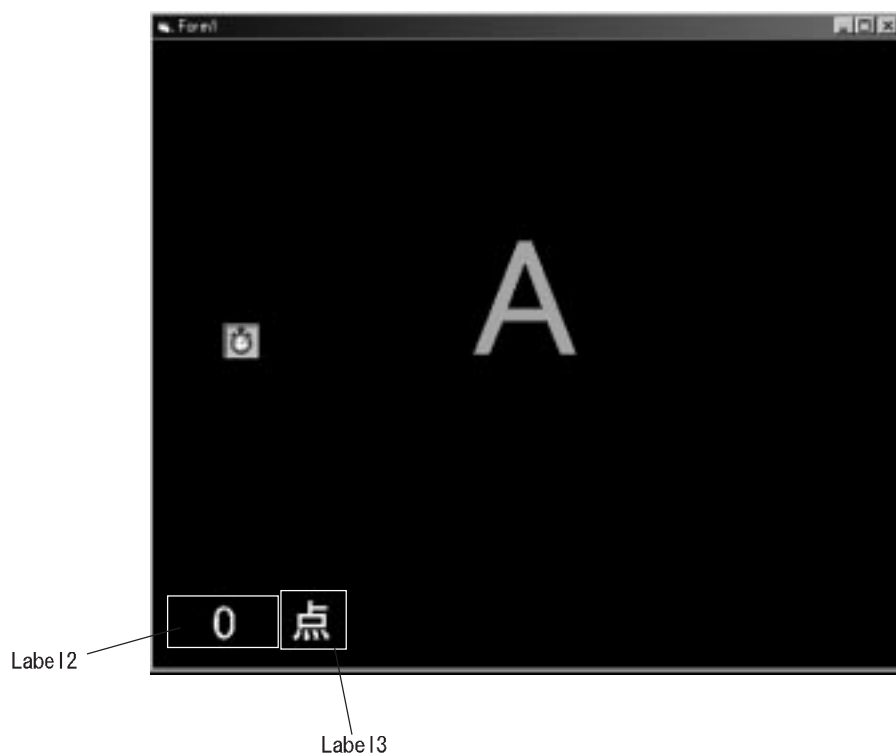
- ・ 任意の数字を発生させたいとき、乱数を利用します。
- ・ プログラム全体の中で1回、「Randomize」を実行します。
- ・ 「Randomize」は乱数を初期化する命令で、トランプをシャフルするようなものです。
- ・ 「Rnd」は「Randomize」された乱数の中から順に1つずつ乱数を発生させる命令です。
- ・ 「Rnd」は0.0～0.999999(1未満)の間の実数の乱数を発生させます。
- ・ 例えば、「a」という変数に0～19の20種類の数字の乱数を発生させたい場合は次のような計算をさせます。



### 4-3. 得点を表示させよう

画面に表示された文字と同じ文字をキーボードから入力すると、得点されるようにしましょう。

- ① 下記のように、得点と「点」を表示するラベルを作成します。



- ② フォームをダブルクリックしてコードエディタウィンドウを開きます。
- ③ 右側のプロシージャの中から「KeyDown」を選択します。  
(キーボードが押されたときに実行されるプログラムを作成します。)





- ④ 下記のプログラムを入力します。

```
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = Asc(Label1.Caption) Then
        ten = ten + 10
        Label2.Caption = ten
    End If
End Sub
```

入力された文字コード

フォーム上でキーが押されたときの処理

押された文字コードが表示された文字コードと同じなら

点数を10点プラス

点数を表示

- ⑤ プログラムを実行します。

表示と同じキーを入力すると、点数が「10点」と表示されます。

しかし、何度正解しても10点より大きくなりません。

これは、各サブプログラム内でプログラムが実行されるたびに変数は毎回「0」に初期化されるからです。

つまり、キーが押されるたびに、「ten」は毎回0に戻ってしまい、そこに10点が加算されるので10点より大きくなることはないのです。

そこで次のプログラムを追加します。

- ⑥ コードエディタウィンドウの左上のオブジェクトを (General) にします。

カーソルがエディタウィンドウの一番上に移動します。

そこに、「Dim ten As Integer」と入力します。

これは、変数「ten」をフォーム中の全てのサブプログラム内で共通に使うようにし、変数の値をそのまま残すようにする命令です。



- ⑦ プログラムを実行します。  
今度は得点が正しく加算されていくはずです。



- ⑧ フォームとプロジェクトを新しい名前でフロッピーディスクに保存します。

## 4-4. 時間がきたら GAME OVER と表示させよう

ある時間になったら、ゲームをストップさせましょう。  
ゲームをストップさせるには、次の作業が必要です。

1秒ごとに

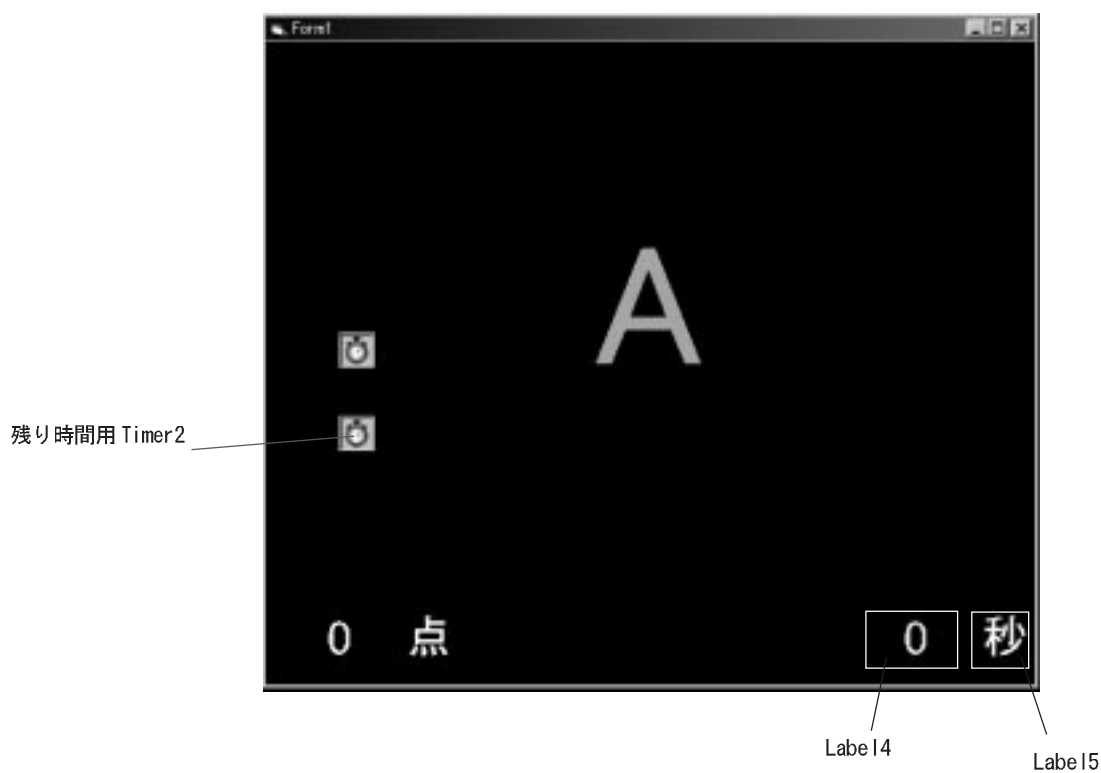
- ・残り時間を減らす

時間になったら

- ・タイマーを止める
- ・表示されている文字を消す
- ・GAME OVER と表示する

順に作成していきましょう。

- ① 残り時間を表示させるラベルと残り時間用のタイマーを作成します。



② 1秒ごとに残り時間が1ずつ減っていくプログラムを追加します。

```
Project1 - Form1 (C#)
Timer2 Timer
Timer

Dim ten As Integer
Dim byo As Integer

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = Asc(Label1.Caption) Then
        ten = ten + 10
        Label2.Caption = ten
    End If
End Sub

Private Sub Form_Load()
    Randomize
    byo = 20
End Sub

Private Sub Timer1_Timer()
    moji = Int(Rnd() * 26) + Asc("A")
    Label1.Caption = Chr(moji)
End Sub

Private Sub Timer2_Timer()
    byo = byo - 1
    Label4.Caption = byo
End Sub
```

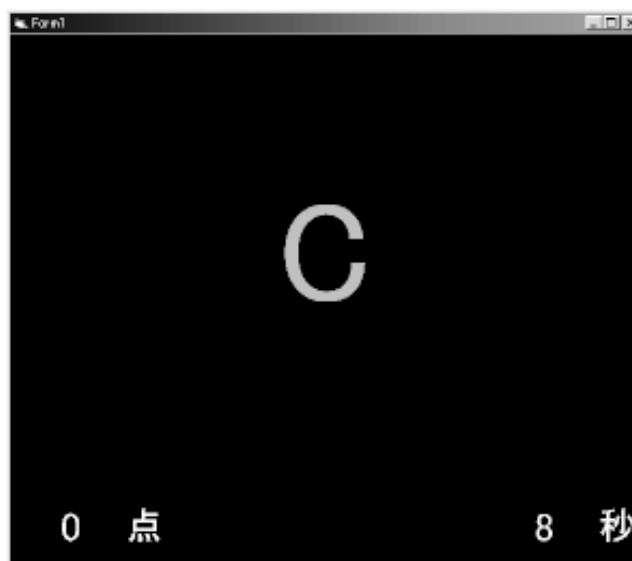
残り時間カウント用変数

残り時間カウント変数  
初期設定

カウントを1減らし  
表示する(1秒ごと)

③ Timer2 のプロパティの Interval を 1000 (1 秒) にします。  
(1 秒ごとにプログラムを実行させます。)

④ 実行すると、残り時間が表示されます。  
ただしこのままでは、残り時間はマイナスになっても止まりません。



- ⑤ 残り時間が0になった時、カウントダウンが止まり、文字の表示が消えるプログラムを追加します。

```
Private Sub Timer2_Timer ()
    byo = byo - 1
    Label4.Caption = byo

    If byo = 0 Then
        Timer1.Interval = 0
        Label1.Visible = False
        Timer2.Interval = 0
    End If
End Sub
```

追加

もし、0秒になったら

文字表示のタイマーを止める

文字を見えなくする

カウントダウンのタイマーを止める

- ⑥ 更に「GAME OVER」と表示させましょう。  
まず、「GAME OVER」と書いたラベルを作成し、VisibleをFalseにしておきます。



- ⑦ 上記の IF 文の中に次の命令を追加します。

```
Label6.Visible = True . . . Label6 (GAME OVER) を見えるようにする
```

⑧ プログラムを実行してみましょう。

残り時間が0秒になったとき、文字が消えて、「GAME OVER」と表示されれば完成です！

⑨ ファイルを保存しましょう。



## 5. ゲームに機能を追加してみよう

### 5-1. 文字のサイズが変わるようにしよう

乱数を使って文字のサイズが毎回変わるようにします。

- ① 文字を表示させている、「Timer1」のプログラムに次の命令を追加します。

```
Label1.FontSize = Int(Rnd() * 100) + 20
```

0 ~ 99 までの整数

20 ~ 119 までの整数

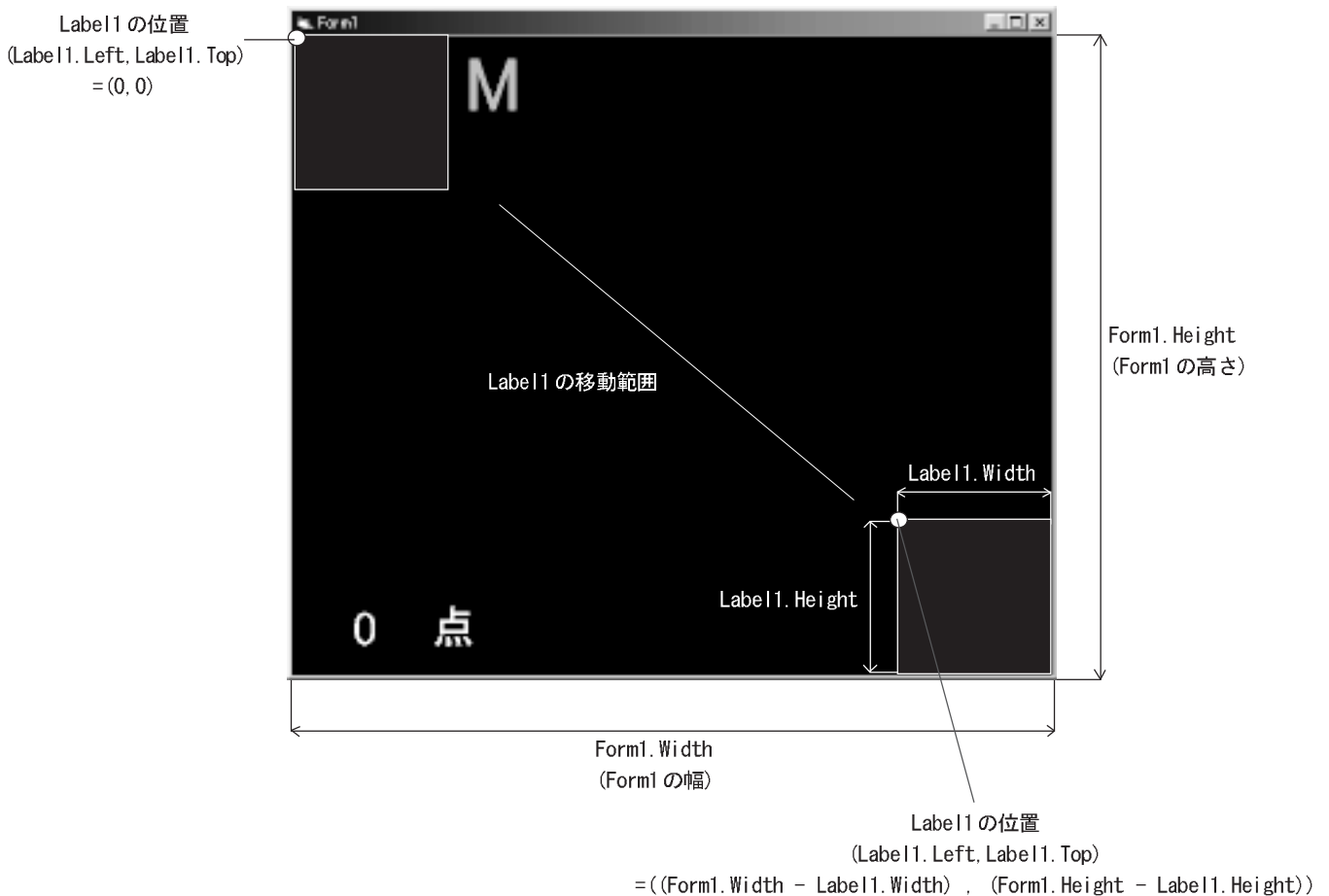
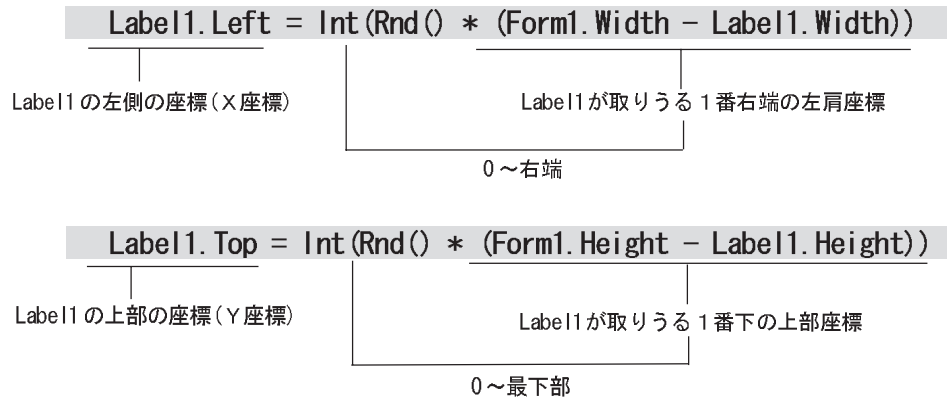
- ② 実行すると、毎回文字のサイズが変わります。



## 5-2. 文字の表示位置が変わるようにしよう

乱数を使って文字の表示位置が毎回変わるようにします。

- ① 文字を表示させている、「Timer1」のプログラムに次の命令を追加します。



- ② 実行すると、毎回文字の表示する位置が変わります。



### 5-3. 当たったら音を鳴らそう

表示されている文字と同じ文字が押されたら、当たりの音(wave データ)を出します。

- ① 鳴らしたい音を、あらかじめフロッピーディスクに保存しておきます。
- ② コードエディタウィンドウのオブジェクト(左上)を(General)にして、下記のプログラムを入力します。  
これは音を鳴らすための準備プログラムです。  
このまま入力してください。

```
Private Declare Function sndPlaySound Lib "winmm" _
Alias "sndPlaySoundA" (ByVal WaveFile As Any, _
ByVal uFlags As Long) As Long
```

- ③ 得点を加算しているところと同じところに下記のプログラムを追加してください。

```
i% = sndPlaySound("a:¥oto.wav", 1)
```

↑  
鳴らしたいサウンドデータのファイル名

- ④ プログラムを実行すると、得点すると同時にサウンドが鳴ります。

## 5-4. まとめ

ゲームの全プログラムは下記の通りです。

```
Private Declare Function sndPlaySound Lib "winmm" _
Alias "sndPlaySoundA" (ByVal WaveFile As Any, _
ByVal uFlags As Long) As Long

Dim ten As Integer
Dim byo As Integer

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = Asc(Label1.Caption) Then
        ten = ten + 10
        Label2.Caption = ten
        i% = sndPlaySound("a:¥oto.wav", 1)
    End If
End Sub

Private Sub Form_Load()
    Randomize
    byo = 20
End Sub

Private Sub Timer1_Timer()
    moji = Int(Rnd() * 26) + Asc("A")
    Label1.Caption = Chr(moji)
    Label1.FontSize = Int(Rnd() * 100) + 20
    Label1.Left = Int(Rnd() * (Form1.Width - Label1.Width))
    Label1.Top = Int(Rnd() * (Form1.Height - Label1.Height))
End Sub

Private Sub Timer2_Timer()
    byo = byo - 1
    Label4.Caption = byo
    If byo = 0 Then
        Timer1.Interval = 0
        Timer2.Interval = 0
        Label1.Visible = False
        Label6.Visible = True
    End If
End Sub
```

サウンド初期設定

変数 ten 宣言

変数 byo 宣言

キーが押された時の処理

もし、表示された文字と  
押されたキーが同じなら  
点数を加算して表示

音を鳴らす

フォームを呼び出したときの  
最初の処理

乱数の初期化

残り時間数の初期設定

文字表示のタイマー

A~Zのある文字を発生させる

文字の表示

表示サイズの変更 (20 ~ 119)

表示位置の変更

残り時間計算のタイマー

byo を 1 減らす

byo の表示

残り時間 (byo) が 0 になったら

文字表示タイマー停止

残り時間計算タイマー停止

文字を消す

「GAME OVER」を表示