
LẬP TRÌNH VISUAL BASIC

BÀI TẬP THỰC HÀNH SỐ 1

MẢNG MỘT CHIỀU

1. Chuẩn đầu ra.

Sau khi học xong chương này, sinh viên có thể:

- Khai báo mảng 1 chiều (mảng tĩnh, mảng động).
- Viết đoạn code nhập các giá trị cho mảng sử dụng Textbox hoặc InputBox.
- Viết đoạn code xuất các giá trị trong mảng ra Label, Textbox hoặc sử dụng MsgBox.
- Viết 1 chương trình VB có sử dụng cấu trúc mảng.
- Viết 1 chương trình VB có sử dụng cấu trúc mảng và mảng các đối tượng điều khiển
- Có thái độ làm việc chăm chỉ, hình thành phong cách lập trình.

2. Chuẩn bị

SV phải làm (trên giấy) các bài tập về mảng đã làm trên lớp giờ lý thuyết và làm thêm ở nhà.

3. Phương tiện

Phòng máy có cài chương trình Microsoft Visual Basic 6.0

4. Thời lượng

4 tiết.

5. Tóm tắt lý thuyết

Mảng là tập hợp các phần tử cùng kiểu dữ liệu được đánh thứ tự. Số thứ tự của mỗi phần tử được gọi là *chỉ số*. Chỉ số mặc định bắt đầu từ 0. Nếu muốn chỉ số bắt đầu từ 1, sử dụng khai báo *Option Base 1*.

5.1 Khai báo mảng

Dim tên_biến(số phần tử-1) **As** kiểu_dữ_liệu

Ví dụ: Dim a(5) As Integer

Chỉ số	0	1	2	3	4	5
Giá trị	0	0	0	0	0	0

Dim b(3) As Double

Chỉ số	0	1	2	3
Giá trị	0.0	0.0	0.0	0.0

Dim c(3) As String

Chỉ số	0	1	2	3
Giá trị	""	""	""	""

Dim d(3) As Boolean

Chỉ số	0	1	2	3
Giá trị	False	False	False	False

Bắt đầu từ chỉ số bất kỳ:

Dim tên_biến(chỉ_số_đầu_tiên **To** chỉ_số_cuối_cùng) **As** kiểu_dữ_liệu

Ví dụ: Dim a(2 To 5) As Integer

Chỉ số	2	3	4	5
Giá trị	0	0	0	0

5.2 Truy xuất mảng

tên_biến(chỉ_số)

Ví dụ 1: Dim temp As Integer

temp=a(2)

Chỉ số	0	1	2	3	4	5
Giá trị	3	-1	10	20	7	34

Sau khi thực hiện đoạn lệnh trên thì temp có giá trị là 10.

Ví dụ 2: Dim temp As Integer

Dim k As Integer

k=1

temp=a(k)

Sau khi thực hiện đoạn lệnh trên thì temp có giá trị là -1.

* Gán giá trị cho phần tử trong mảng:

tên_biến(chỉ_số) = giá trị cần gán

Cho mảng sau:

Chỉ số	0	1	2	3	4	5
Giá trị	3	-1	10	20	7	34

Ví dụ 1: a(3) = 7

Chỉ số	0	1	2	3	4	5
Giá trị	3	-1	10	7	7	34

Ví dụ 2: Dim k As Integer

k=1

Dim temp As Integer

temp=15

a(k)=temp

Chỉ số	0	1	2	3	4	5
Giá trị	3	15	10	20	7	34

5.3 Duyệt mảng

Nếu mảng có chỉ số bắt đầu từ 0 và mảng có n phần tử

For i=0 To n-1 Step 1

<đoạn lệnh>

Next

Nếu mảng a có chỉ số bắt đầu từ một số bất kỳ:

For i=LBound(a) To UBound(a) Step 1

<đoạn lệnh>

Next

*Hàm LBound(tên_biến_mảng) trả về chỉ số nhỏ nhất của mảng.

Hàm UBound(tên_biến_mảng) trả về chỉ số lớn nhất của mảng.

Ví dụ 1: ta có mảng a sau:

Chỉ số	0	1	2	3	4	5
Giá trị	3	15	10	20	7	34

LBound(a) trả về 0.

UBound(a) trả về 5.

Ví dụ 2: ta có mảng b sau:

Chỉ số	3	4	5	6	7	8
Giá trị	3	15	10	20	7	34

LBound(b) trả về 3.

UBound(b) trả về 8.

5.4 Mảng động

Là mảng có số phần tử có thể thay đổi tùy ý trong lúc chạy chương trình. Điều này phù hợp hơn vì trong thực tế người lập trình không thể tiên liệu trước số phần tử thực tế.

Sử dụng mảng động gồm 2 bước:

- Khai báo hiện diện với từ khóa Dim nhưng số phần tử để rỗng.
- Tạo mảng thực sự khi cần thiết bằng phát biểu ReDim.

Ví dụ: Dim a() As Double

Redim a(n) As Double

Mảng động có thể được tạo lại nhiều lần khi cần thiết.

Ví dụ: Dim a() As Double

Redim a(n) As Double

Redim a(m) As Double

Tuy nhiên lệnh cấp phát mới sẽ xóa rỗng nội dung (chuỗi) hoặc gán bằng 0 (số) mọi phần tử đã có giá trị trước đó. Để bảo toàn giá trị các phần tử, sử dụng phát biểu ReDim Preserve.

Ví dụ:

ReDim Preserve a(500) As Double

6. Nội dung thực hành

6.1 Cho mảng một chiều gồm n số nguyên, hãy viết hàm tính tổng của các phần tử trong mảng.

Ví dụ: mảng: 6 1 100 -5 3 -> hàm trả về kết quả: 105

Function TongMang (ByVal a As Variant, ByVal n As Integer) As Long

Hướng dẫn: Duyệt mảng và cộng dồn bằng công thức $s = s + a(i)$

6.2 Cho mảng một chiều gồm n số nguyên, hãy viết hàm tính giá trị trung bình cộng của các phần tử trong mảng.

Ví dụ: mảng: 6 1 100 -5 3 -> hàm trả về kết quả: 21

Function TrungBinhCong (ByVal a As Variant, ByVal n As Integer) As Long

Hướng dẫn: Sau khi tính tổng xong, chia tổng cho số phần tử của mảng.

6.3 Cho mảng một chiều gồm n số nguyên, hãy viết hàm tìm phần tử lớn nhất trong mảng.

Ví dụ: mảng: 6 1 100 -5 3 -> hàm trả về kết quả: 100

Function Max (ByVal a As Variant, ByVal n As Integer) As Long

6.4 Cho mảng một chiều các số nguyên, hãy viết hàm tính số lần xuất hiện của x.

Ví dụ: mảng: 6 3 100 -5 3 6 6 3 100 ; x=6 -> hàm trả về kết quả: 3

Function TanSuat (ByVal a As Variant, ByVal x As Integer) As Integer

Hướng dẫn: So sánh x với a(i), nếu bằng nhau tăng biến đếm lên một đơn vị.

6.5 Sắp xếp mảng a tăng dần. Hàm trả về là một chuỗi chứa mảng a đã được sắp xếp, mỗi phần tử cách nhau khoảng trắng.

Ví dụ: mảng a: 6 100 -3 78 34 -> hàm trả về kết quả: "-3 6 34 78 100"

Function SapXep (ByVal a As Variant) As String

Hướng dẫn: chạy hai vòng for.

6.6 Kiểm tra một mảng một chiều a gồm các phần tử nguyên có phải là cấp số cộng hay không, nếu phải hàm trả về công sai của cấp số cộng, nếu không hàm trả về -1.

Ví dụ: mảng a: 4 7 10 13 16 19 -> hàm trả về kết quả: 3

mảng a: 3 4 3 7 5 0 -> hàm trả về kết quả: -1

Function CapSoCong (ByVal a As Variant) As Integer

6.7 Cho một mảng một chiều a gồm các số thực, kiểm tra mảng a có đối xứng hay không, nếu đối xứng trả về True, nếu không trả về False.

Ví dụ: mảng a: 0.1 3.4 5 3.4 0.1 -> hàm trả về kết quả: True

mảng a: 2.3 5 6 7 5 6.7 -> hàm trả về kết quả: False

Function DoiXung (ByVal a As Variant) As Boolean

6.8* Sắp xếp mảng a có các số lẻ tăng dần(số chẵn giữ nguyên). Hàm trả về là một chuỗi chứa mảng a có các số lẻ đã được sắp xếp, mỗi phần tử cách nhau khoảng trắng.

Ví dụ: mảng a: 6 101 -3 78 35 -> hàm trả về kết quả: "6 -3 35 78 101"

Function SapXep (ByVal a As Variant) As String

7. Bài tập thêm

7.1 Cho mảng một chiều các số nguyên dương có n phần tử. Hãy viết hàm tìm số nhỏ nhất trong mảng. Nếu mảng có số nhỏ nhất thì trả về số nhỏ nhất, nếu mảng không có số nhỏ nhất thì trả về 0.

Ví dụ: mảng 5 5 5 5 5 -> hàm trả về kết quả: 0

mảng 3 2 4 1 5 -> hàm trả về kết quả: 2

Function SoNhoNhi (ByVal a As Variant) As Integer

7.2 Cho mảng một chiều gồm n số nguyên, hãy viết hàm tìm số âm lớn nhất trong mảng, nếu mảng không có số âm thì trả về số 0.

Ví dụ: mảng 4 2 7 9 3 5 6 -> hàm trả về kết quả: 0

mảng 4 -5 8 -11 12 18 -2 -> hàm trả về kết quả: -2

Function AmLonNhat (ByVal a As Variant) As Integer

7.3* Sắp xếp mảng a tăng dần, sau đó chèn thêm một phần tử k vào mảng sao cho mảng vẫn giữ thứ tự tăng dần. Hàm trả về kết quả là một mảng.

Ví dụ: mảng a: 3 4 3 -7 5 0 ; k=1 -> kết quả: -7 0 1 3 3 4 5

Function SapXepChen (ByVal a As Variant, ByVal k As Integer) As Variant

Function SapXepChen (ByVal a As Variant, ByVal k As Integer) As Integer()

7.4** Liệt kê tần suất xuất hiện của các số trong mảng a.

Ví dụ: mảng a: 6 101 -3 -3 35 6 -3

-> hàm trả về kết quả:

“Có 2 số 6, có 1 số 101, có 3 số -3, có 1 số 35”

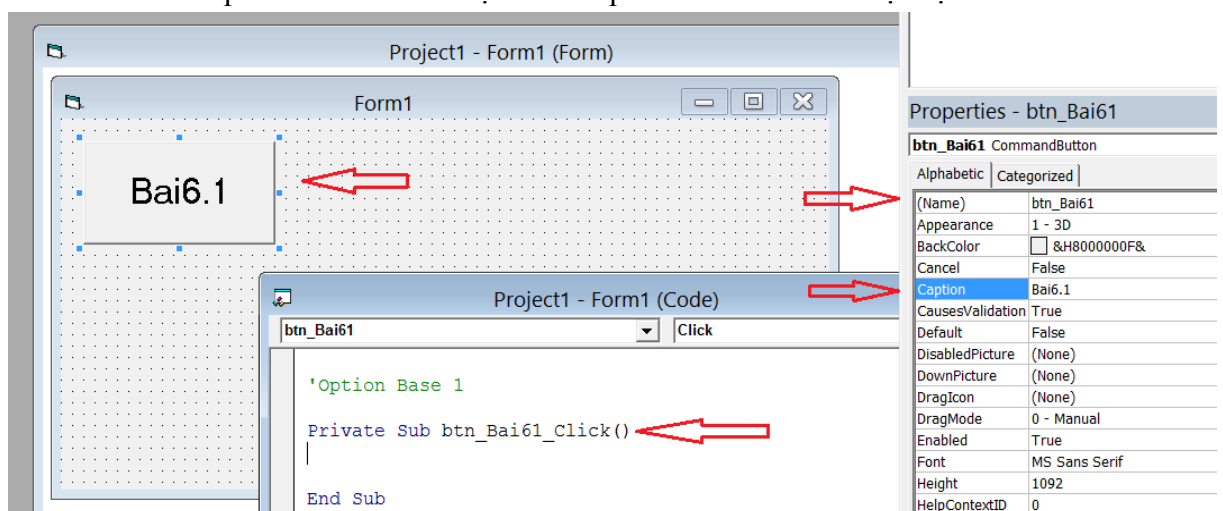
Function TanSuatXuatHien (ByVal a As Integer) As String

8. Hướng dẫn thực hành:

8.1 **Quy ước:** Đối với các bài thực hành chỉ yêu cầu viết hàm tính toán... ta sẽ tạo một giao diện đơn giản có một nút bấm(Button) và dùng sự kiện nhấn nút (Click) để gọi hàm tính toán cần viết. Các giá trị input cần thiết sẽ khai báo trực tiếp hoặc sử dụng InputBox trong code xử lý sự kiện nhấn nút. Tương tự, giá trị output sẽ sử dụng MsgBox để thông báo.

Ví dụ: bài thực hành 6.1 (xem phần 6-Nội dung thực hành)

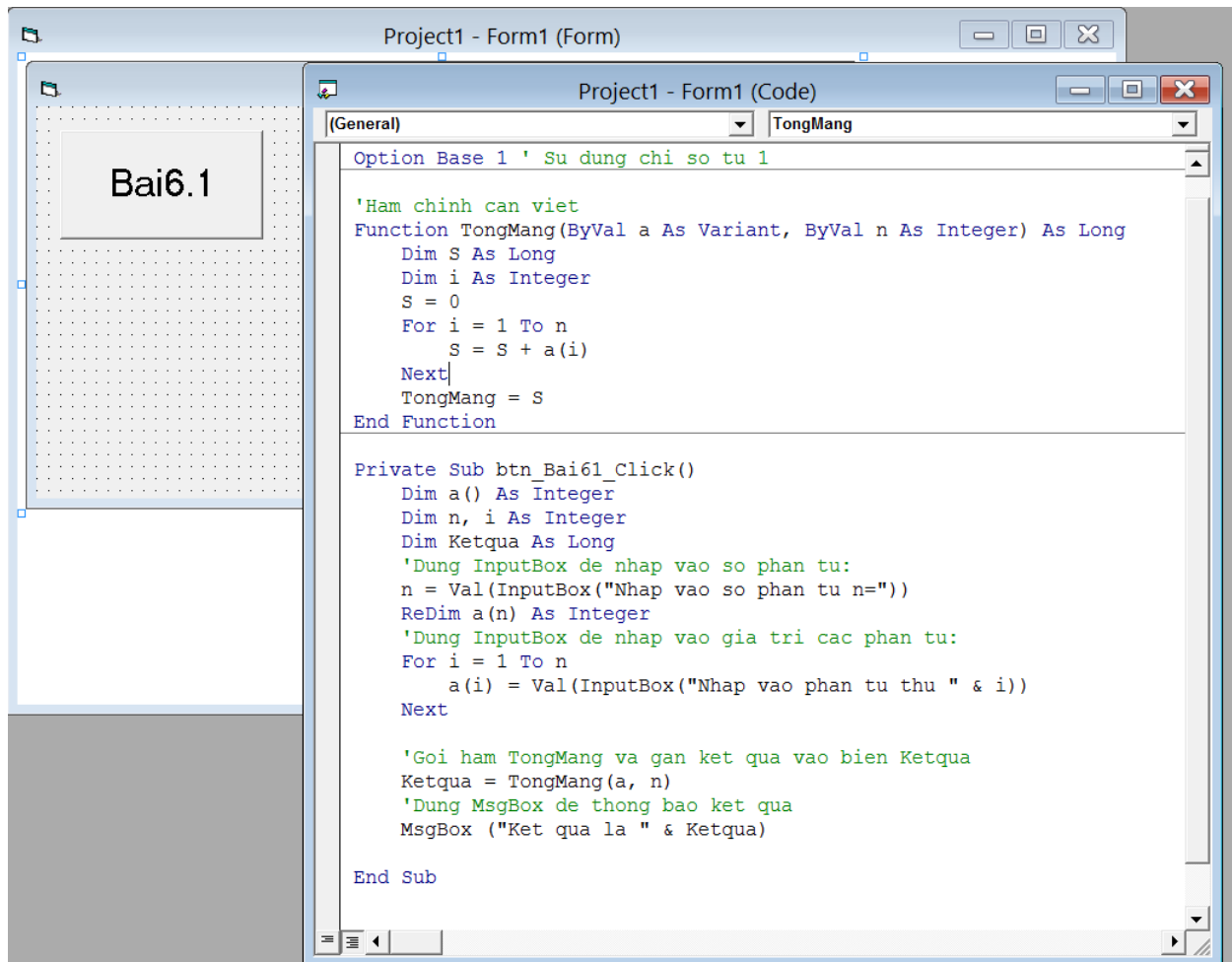
- Kéo thả một nút bấm (Button) vào Form1.
- Sửa thuộc tính: *Name: btn_Bai61, Caption: Bai 6.1.*
- Click đúp vào nút bấm vừa tạo để vào phần viết code xử lý sự kiện Click



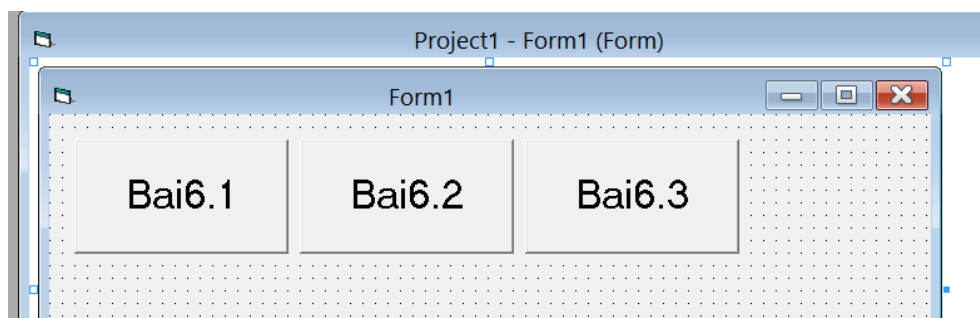
- Viết hàm xử lý yêu cầu tính tổng các phần tử trong mảng theo hướng dẫn

```
'Ham chinh can viet
Function TongMang(ByVal a As Variant, ByVal n As Integer) As Long
    Dim S As Long
    Dim i As Integer
    S = 0
    For i = 1 To n
        S = S + a(i)
    Next
    TongMang = S
End Function
```

- Trong hàm xử lý sự kiện Click, dùng InputBox để khởi tạo mảng a có n phần tử và gọi hàm TongMang vừa viết, gán kết quả vào biến Ketqua và dùng MsgBox để thông báo kết quả.



Ta có thể làm tương tự cho các bài 6.2, 6.3...



Sinh viên cũng có thể sử dụng một Label để xuất kết quả thay vì dùng MsgBox.

Đối với các bài thực hành yêu cầu thiết kế giao diện riêng, thực hiện theo hướng dẫn riêng của từng bài.

8.2 Một số lưu ý khi làm việc với mảng:

- Nên sử dụng chỉ số từ 1 để dễ tính toán: khai báo *Option Base 1*

- Tham số là mảng luôn yêu cầu phải truyền ByRef (mặc định). Nếu muốn dùng ByVal, ta có thể dùng biến kiểu Variant để thay thế. Ví dụ ở bài 6.1, hàm xử lý có thể khai báo theo 1 trong 3 cách sau:

*Function TongMang(**ByRef** a() As Integer, ByVal n As Integer) As Long*

Function TongMang(a() As Integer, ByVal n As Integer) As Long

*Function TongMang(**ByVal** a As Variant, ByVal n As Integer) As Long*

Ở bài này do không cần phải thay đổi gì ở mảng a nên ta nên dùng cách 3 (ByVal) để tránh trường hợp có thể sai sót làm thay đổi mảng a. Cách 1 và 2 thực chất là như nhau do truyền ByRef là mặc định (không khai báo có nghĩa là ByRef)

- Ta có thể dùng đoạn code sau (được bôi đen-sử dụng cho bài 6.1) để khai báo và gán trực tiếp một số giá trị phần tử vào mảng không cần dùng hàm InputBox để nhập vào.

```
Private Sub btn_Bai61_Click()  
    Dim n, i As Integer  
    Dim Ketqua As Long  
    'Khai báo một mảng có 6 phần tử (n=6) và gán trực tiếp giá trị  
    Dim a As Variant  
    a = Array(10, 2, 30, 4, 5, 60)  
    n = 6  
    'Gọi hàm TongMang và gán kết quả vào biến Ketqua  
    Ketqua = TongMang(a, n)  
    'Dùng MsgBox để thông báo kết quả  
    MsgBox ("Ket qua la " & Ketqua)  
  
End Sub
```

9. Hướng dẫn phân mảng điều khiển:

Phần này sẽ hướng dẫn SV xây dựng ứng dụng Máy tính bỏ túi với các phép tính +, -, *, / đơn giản thông qua đó chỉ ra lợi ích của việc sử dụng mảng điều khiển. Máy tính sẽ có một Label hiển thị, 10 nút số (0-9), 4 nút phép tính (+, -, *, /) và nút =.

Ngoài Label và 5 nút chức năng, ta thấy 10 nút số từ 0-9 có chức năng giống nhau. Thay vì sử dụng 10 nút riêng biệt và copy, chỉnh sửa 10 đoạn code xử lý khác nhau, ta sẽ gom 10 nút này vào một mảng điều khiển để thuận lợi hơn trong việc viết và chỉnh sửa code xử lý.

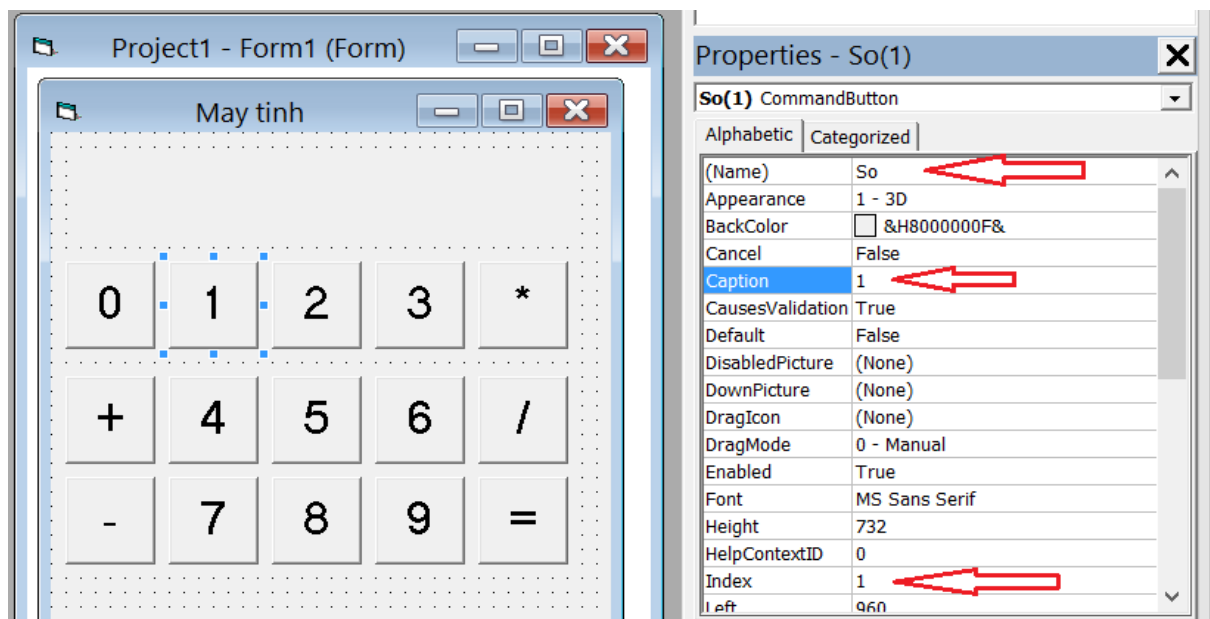
Các nút +, -, *, /, = và label hiển thị ta thiết kế bình thường:

- Nút Cộng: Caption: +, Name: Cong

- Nút Trừ: Caption: -, Name: Tru

- Nút Nhân: Caption: +, Name: Nhan
- Nút Chia: Caption: /, Name: Chia
- Nút Bằng: Caption: =, Name: Bang
- Label Hiển thị: Caption: 0, Name: Hienthi

Riêng 10 nút số ta chỉ cần thiết kế nút số 0 [Caption : 0, Name : So]. Sau đó copy/ paste ra các nút số 1-9, chỉnh sửa lại Caption cho phù hợp từ 1 đến 9. Khi copy/paste lần đầu, VB sẽ hỏi có muốn tạo mảng điều khiển không (...Do you want to create a control array?), chọn đồng ý (Yes).



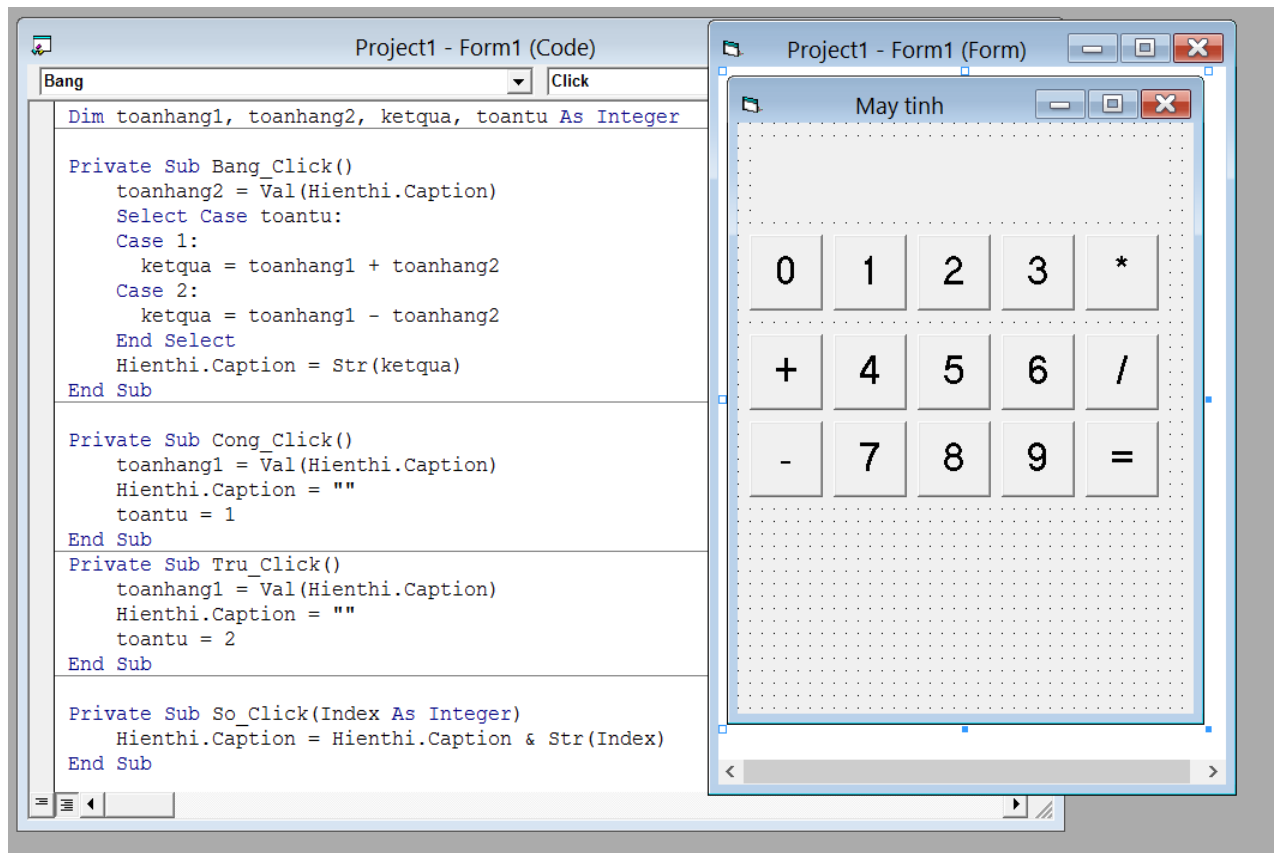
Kích đúp vào nút số 0 để viết code xử lý sự kiện bấm nút cho mảng điều khiển gồm 10 phần tử (10 số): khi bấm nút, giá trị nút đó sẽ hiển thị trên label hiển thị nối thêm vào bên phải). Ta thấy giá trị của mỗi nút cũng chính là giá trị Index của nó trong mảng điều khiển nên code xử lý cho mảng điều khiển sẽ là:

```
Private Sub So_Click(Index As Integer)
    Hienthi.Caption = Hienthi.Caption & Str(Index)
End Sub
```

Như vậy là đã xong phần hiển thị, khi chạy thử chương trình, bấm các nút số để hiển thị số trên label hiển thị.

Phần hiện thực các phép tính ta sẽ sử dụng 4 biến toàn cục lần lượt theo tên gọi để lưu : toán hạng 1, toán hạng 2, toán tử và kết quả phép tính. Khi người dùng nhập vào toán hạng thứ nhất, bấm nút cộng, số thứ nhất sẽ được lưu vào toanhang1, giá trị label hiển thị sẽ được xóa để nhập số thứ 2 đồng thời biến toantu sẽ được gán bằng 1 (quy ước: 1- cộng, 2- trừ, 3-nhân, 4-chia). Khi người dùng nhập số thứ 2 xong, bấm nút bằng, chương trình sẽ lưu số thứ 2 vào toanhang2 và thực hiện phép tính sau đó hiển thị kết quả lên label hiển thị.

Bên dưới là code tham khảo cho các nút cộng, trừ và bằng. Phần viết code xử lý cho phép nhân và chia (nút nhân, chia) xem như bài tập để SV thực hiện.



Bài tập: Ta thấy 4 nút cộng trừ nhân chia cũng có chức năng khá giống nhau. Thay vì dùng 4 nút riêng biệt, hãy dùng một mảng điều khiển có bốn phần tử (4 nút) để hiện thực. Tham khảo thêm đoạn code xử lý bên dưới, giá trị Index phụ thuộc vào Option Base 1 hay 0.

```

Private Sub Tinh_Click(Index As Integer)
    toanhang1 = Val(Hienthi.Caption)
    Hienthi.Caption = ""
    toantu = Index
End Sub

```

---HẾT---