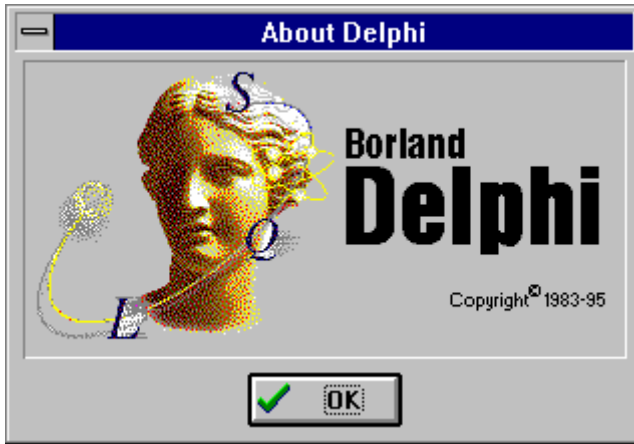


Những ví dụ trong lập trình Delphi



Quyển 2

Delphi Programming

Bạn cần phải đọc quyển I trước khi đọc những ví dụ trong quyển này. Quyển sách này giành cho bạn, hãy bấm và bấm, bạn sẽ thông thạo Delphi.

Hướng dẫn này sẽ giúp bạn điều khiển Delphi. Có vài chương trình ví dụ cùng với những nhiệm vụ tương tự. Bạn hãy thử làm các nhiệm vụ đó dựa vào các ví dụ.

Tất cả các bài làm của các ví dụ, bạn nên bỏ vào thư mục của mình.

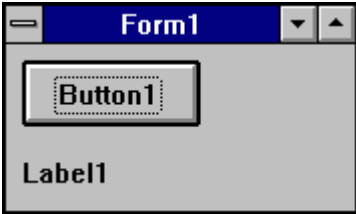

D:\Examples\ME\

Nếu bạn muốn “phê” thì trên Website của tui có các chương trình ví dụ này (bao gồm tất cả các file cần thiết)

Phụ Lục

Ví dụ	Trang	Nội dung ví dụ
1) Output	3	Hiển thị text trong 1 label.
2) Input	4	Gộp các chuỗi text.
3) Add	5	Add, Subtract, Multiply, Divide
4) RTotal	6	Running Total
5) Picture	8	Đồ họa trong Delphi
6) Lệnh If	10	Quyết định thực hiện
7) Animation	12	Animation đơn giản
8) Lệnh For	14	Sự lặp lại sử dụng For
9) Lệnh While	15	Sự lặp lại sử dụng While
10) Số số	17	Số số kiên thiết!!!
11) Delay	19	Hoãn thời gian - Giờ hệ thống
12) Calculator	21	Một chiếc Calculator (máy tính)
13) Fractal	24	Đồ họa Fractal
Advanced\Spiral	26	Đồ họa tiên tiến hơn

1) Ví dụ về OutPut (Xuất) dữ liệu

Biểu đồ IPO	Button1 Click
Đầu vào	Click vào button.
Quá trình	'Hello World!' chép vào label1.
Dữ liệu nhập thử	Không.
Chờ đợi đầu ra	'Hello World!' hiện trên label.
Đầu ra thực tế	'Hello World!'
Bảng chứng	
BEFORE CLICK 	AFTER CLICK 
Hành động cần thiết	Không.

{ Ví dụ Chương trình này minh họa việc hiện text trên label khi ta bấm chuột.

Nhiệm vụ Thêm button và label khác vào chương trình này. Thiết đặt sao cho phù hợp với tình huống. }

```

unit Unit1;
interface                                     { Delphi tạo dòng này }
uses                                           { Delphi tạo dòng này }
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics,
  Controls, Forms, Dialogs, StdCtrls;

type                                           { Delphi tạo dòng này }
  TForm1 = class(TForm)
    Button1: TButton;
    Label1: TLabel;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var                                           { Delphi tạo dòng này }
  Form1: TForm1;

implementation                               { Delphi tạo dòng này }

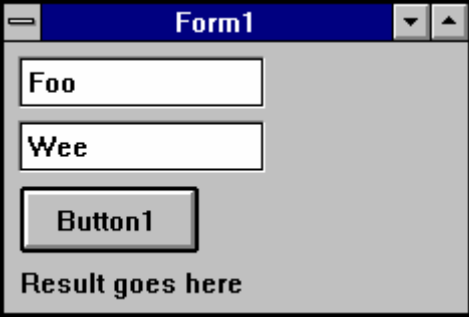
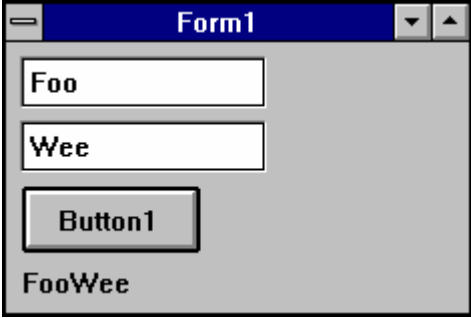
{$R *.DFM}

procedure TForm1.Button1Click(Sender: TObject);
begin                                         { Delphi tạo dòng này }
  label1.caption := 'Hello World!';          { BẠN ĐÁNH Ở DÒNG NÀY }
end;                                         { Delphi tạo dòng này }

end.                                         { Delphi tạo dòng này }

```

2) Ví dụ về việc Input (Nhập) dữ liệu

IPO Chart	Button1 Click
Đầu vào	Click vào button.
Quá trình	Nối các chuỗi văn bản được nhập. Hiển thị kết quả
Nhập dữ liệu thử	Foo Wee
Đợi đầu ra	FooWee
Đầu ra thực tế	FooWee
Bảng chứng	<div> <div>BEFORE CLICK</div>  </div> <div> <div>AFTER CLICK</div>  </div>
Hoạt động cần thiết	Không.

{ Ví dụ Chương trình nối 2 chuỗi văn bản lại.
N.Vụ Thay đổi chương trình sao cho bất cứ cái gì bạn nhập được nối lại và lặp lại 2 lần. }

```
unit Unit1;
interface
uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls;
```

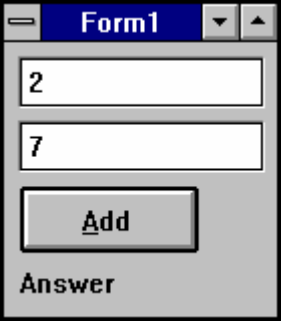
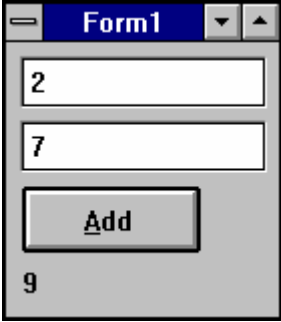
```
type
  TForm1 = class(TForm)
    Edit1: TEdit;
    Edit2: TEdit;
    Button1: TButton;
    Label1: TLabel;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

```
var Form1: TForm1;
```

```
implementation
{$R *.DFM}
```

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  { Nhập 2 mảnh của text lại. Dấu + nghĩa là nhập. }
  label1.caption := edit1.text + edit2.text;
end;
end.
```

3)Thực hiện phép tính cộng (Add)

IPO Chart	ButtonAdd Click
Nhập	Click the button.
Quá trình	Cộng những số được nhập vào. Hiện kết quả trong labelAnswer
Nhập dữ liệu thử	2 và 7
Chờ đầu ra	9
Đầu ra thực tế	9
Bảng chứng	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>BEFORE CLICK</p>  </div> <div style="text-align: center;"> <p>AFTER CLICK</p>  </div> </div>
Hành động cần	Không.

{ Ví dụ Cộng 2 số với nhau
N.Vụ Sử dụng - * div để trừ, nhân và chia lấy dư.
Điều gì sẽ xảy ra khi bạn div bằng 0? }

```

unit Unit1;
interface
uses SysUtils, WinTypes, WinProcs, Messages, Classes,
    Graphics, Controls, Forms, Dialogs, StdCtrls;
type
    TForm1 = class(TForm)
        Edit1: TEdit;
        Edit2: TEdit;
        ButtonAdd: TButton;
        LabelAnswer: TLabel;
        procedure ButtonAddClick(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

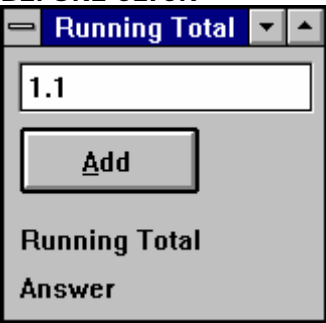

var
    Form1: TForm1;

implementation
{$R *.DFM}

procedure TForm1.ButtonAddClick(Sender: TObject);
begin
    { intToStr để chuyển số nguyên thành chuỗi text }
    { strToInt để chuyển chuỗi text thành số nguyên }
    LabelAnswer.Caption := intToStr(strToInt(edit1.text) +
                                    strToInt(edit2.text));
end;
end.

```

4) Ví dụ về Running Total

IPO Chart	ButtonAdd Click
Nhập	Click vào button.
Quá trình thực hiện	Thêm số được nhập vào giá trị của runningTotal (giá trị đầu là 0).
Dữ liệu nhập thử	1.1
Chờ đầu ra	1.1
Đầu ra thực tế	1.100000000000036
Bảng chứng	
BEFORE CLICK 	AFTER CLICK 
Hành động cần làm	Làm tròn một số lỗi nhỏ. Không cần làm gì trong ví dụ này.

```
{ Ví dụ      Thêm các số được nhập vào và hiển thị running total.
                Ghi chú làm tròn các lỗi!
    N.Vụ 1      Đếm các số khi chúng được nhập vào.
                Hiện thị kết quả đếm.
    N.Vụ 2      Tiến triển trung bình của những số được nhập vào.
                Hiện thị kết quả trung bình.
}
```

```
unit Unit1;
```

```
interface
```

```
uses
```

```
SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,  
Forms, Dialogs, StdCtrls;
```

```
type
```

```
TFormMain = class(TForm)
    EditInput: TEdit;
    ButtonAdd: TButton;
    LabelRunningTotal: TLabel;
    LabelAnswer: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure ButtonAddClick(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }

    runningTotal : Real;    { Số thực là số dạng 123.456 }
end;
```

```
var
```

Những ví dụ về lập trình bằng Delphi; Biên dịch bởi Lê Khắc Như; website: <http://www.lkn.b4.to>
2/21/04

```
FormMain: TFormMain;
```

implementation

```
{ $R *.DFM }
```

```
{ Thủ tục này hoạt động khi chương trình được khởi động.  
  Nó khởi tạo biến runningTotal tới 0. }
```

```
procedure TFormMain.FormCreate(Sender: TObject);
```

```
begin
```

```
    runningTotal := 0;
```

```
end;
```

```
{ Thủ tục xử lý khi nhấn nút. }
```

```
procedure TFormMain.ButtonAddClick(Sender: TObject);
```

```
begin
```

```
{ strToFloat converts một chuỗi text tới dấu phẩy động của số thực }
```

```
{ floatToStr converts một dấu phẩy động của số thực tới 1 chuỗi text }
```

```
    runningTotal := runningTotal + strToFloat(editInput.Text);
```

```
    labelAnswer.caption := floatToStr(runningTotal);
```

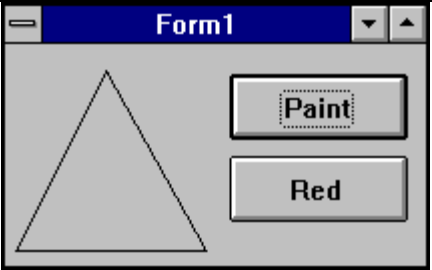
```
    editInput.setFocus; { Làm editInput sẵn sàng cho số tiếp theo. }
```

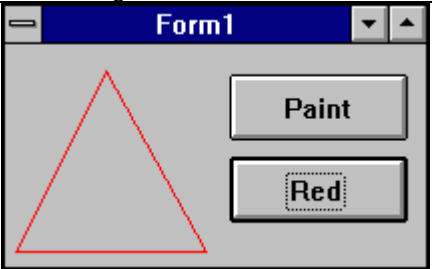
```
    editInput.selectAll; { Tất cả dữ liệu editInput đã được bôi đen. }
```

```
end;
```

```
end.
```

5) Ví dụ về Picture (Hình ảnh)

IPO	ButtonPaint Click
Input	Click vào ButtonPaint
Input Test Data	Không
Process	Sử dụng Canvas.Pen.Color, Canvas.MoveTo và Canvas.LineTo để vẽ tam giác màu đen.
Expected Output	Hình tam giác màu đen cần được vẽ.
Actual Output	Hình tam giác được vẽ.
Evidence	
Action Needed	Không

IPO	ButtonRed Click
Input	Click vào ButtonRed
Input Test Data	không
Process	Sử dụng Canvas.Pen.Color, Canvas.MoveTo và Canvas.LineTo để vẽ tam giác màu đỏ.
Expected Output	Red triangle should be drawn.
Actual Output	Red triangle is drawn.
Evidence	
Action Needed	Không

```
{
    EXAMPLE    Vẽ đồ họa đơn giản.

    TASK       Vẽ hình vuông màu vàng và xanh lục. Sử dụng Delphi help
               Kiểm đối tượng TCanvas. Xem phương pháp
               làm. Và có lẽ bạn sẽ vẽ được hình tròn, textOut,
               cung và nhiều nữa.
}
unit Unit1;

interface

uses
    SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
    Forms, Dialogs, StdCtrls, ExtCtrls;
```


Những ví dụ về lập trình bằng Delphi; Biên dịch bởi Lê Khắc Như; website: <http://www.lkn.b4.to>
2/21/04

type

```
TForm1 = class(TForm)
  PaintBox1: TPaintBox;
  ButtonPaint: TButton;
  ButtonRed: TButton;
  procedure ButtonPaintClick(Sender: TObject);
  procedure ButtonRedClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
```

var

```
Form1: TForm1;
```

implementation

```
{ $R *.DFM }
```

```
procedure TForm1.ButtonPaintClick(Sender: TObject);
```

```
begin
```

```
  paintBox1.Canvas.Pen.Color := clBlack;
```

```
  paintBox1.Canvas.MoveTo(50, 10);
```

```
  paintBox1.Canvas.LineTo(100, 100);
```

```
  paintBox1.Canvas.LineTo(5, 100);
```

```
  paintBox1.Canvas.LineTo(50, 10);
```

```
end;
```

```
procedure TForm1.ButtonRedClick(Sender: TObject);
```

```
begin
```

```
  paintBox1.Canvas.Pen.Color := clRed;
```

```
  paintBox1.Canvas.MoveTo(50, 10);
```

```
  paintBox1.Canvas.LineTo(100, 100);
```

```
  paintBox1.Canvas.LineTo(5, 100);
```

```
  paintBox1.Canvas.LineTo(50, 10);
```

```
end;
```

```
end.
```

6) Ví dụ về cách sử dụng lệnh IF

{ Ví dụ Quyết định sử dụng lệnh IF.

Nút nhấn sẽ làm form thành màu đỏ.
Không quyết định gì nữa.

Nút thứ hai làm form thành màu xanh lục
nếu màu của form lúc đó là màu đỏ.

Nút thứ ba.
Nếu form màu đỏ, làm nó thành màu xanh lục.
Nếu form màu xanh lục, làm nó thành màu vàng.
Nếu form màu vàng, làm nó thành màu đỏ.
Không trả lời tới mọi màu khác.

N.Vụ Thêm nút thứ tư. Làm nó thay đổi các
màu trên bất cứ lúc nào bạn thích.
Bạn nên tìm xem có những màu nào có
sẵn bằng các làm theo cách sau đây:

Help | Topic Search
Đánh màu vào TColor
Bấm Enter 2 lần.
Click vào TColor
Click vào Color Property

}

unit Unit1;

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, StdCtrls;

type

TForm1 = **class**(TForm)
 Button1: TButton;
 Button2: TButton;
 Button3: TButton;
 procedure Button1Click(Sender: TObject);
 procedure Button2Click(Sender: TObject);
 procedure Button3Click(Sender: TObject);
 private
 { Private declarations }
 public
 { Public declarations }
end;

var

Form1: TForm1;

implementation

{ \$R *.DFM }

{ Làm form thành màu đỏ. Quyết định N làm cái này. }

Những ví dụ về lập trình bằng Delphi; Biên dịch bởi Lê Khắc Như; website: <http://www.lkn.b4.to>
2/21/04

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
    Form1.color := clRed;  
end;  
  
{ Nếu form màu đỏ, làm nó thành xanh lục. Không làm khác nữa. }  
procedure TForm1.Button2Click(Sender: TObject);  
begin  
    if Form1.color = clRed then  
    begin  
        Form1.color := clGreen  
    end  
end;  
  
{ Nếu form màu đỏ, làm nó thành màu xanh lục.  
  Nếu form màu xanh lục, làm nó thành màu vàng.  
  Nếu form màu vàng, làm nó thành màu đỏ.  
  Không làm nữa nếu form đã đổi đủ màu. }  
procedure TForm1.Button3Click(Sender: TObject);  
begin  
    if Form1.color = clRed then  
    begin  
        Form1.color := clGreen  
    end  
    else if Form1.color = clGreen then  
    begin  
        Form1.color := clYellow  
    end  
    else if Form1.color = clYellow then  
    begin  
        Form1.color := clRed  
    end  
  
end;  
  
end.
```

7) Ví dụ sử dụng Animation (Ảnh động)

Ví dụ Chương trình này sử dụng 2 control và timers. Sử dụng windows paintbrush để vẽ 2 hình. Một hoàn đảo hoang và 1 tấm lợp.

Thiết bị hẹn giờ sẽ chỉ định thời gian để tấm lợp di chuyển là 3 giây.

mặt người đàn ông biến đổi trong từng giây.

N.Vụ Thiết lập hoạt cảnh cho mình. Nên tìm hiểu cách dùng Timer để hiển thị ảnh.

}

unit Unit1;

interface

uses

SysUtils, WinTypes, WinProcs,
Messages, Classes, Graphics,
Controls, Forms, Dialogs,
ExtCtrls, StdCtrls;

type

TForm1 = **class**(TForm)
Timer1: TTimer;
Image2: TImage;
Image1: TImage;
Timer2: TTimer;
Button1: TButton;
procedure Timer1Timer(Sender: TObject);
procedure Timer2Timer(Sender: TObject);
procedure Button1Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;

var

Form1: TForm1;

implementation

{ \$R *.DFM }

{ image2 được di chuyển theo chỉ định của timer1. }

procedure TForm1.Timer1Timer(Sender: TObject);

begin

image2.Left := image2.Left + 2;

if image2.Left > 169 **then** { Đóng chương trình trình khi ảnh }
begin { tới phía phải màn hình. }



Những ví dụ về lập trình bằng Delphi; Biên dịch bởi Lê Khắc Như; website: <http://www.lkn.b4.to>
2/21/04

```
        close
    end
end;

{ Làm cho man nhảy mắt trong mỗi biến đổi của timer2. }
procedure TForm1.Timer2Timer(Sender: TObject);
begin
    { Thay đổi màu mắt của man. }
    if image1.canvas.brush.color = clGray then
    begin
        image1.canvas.brush.color := clWhite
    end
    else
    begin
        image1.canvas.brush.color := clGray
    end;

    { Vẽ một elip nhỏ. Đó là con mắt. }
    image1.canvas.ellipse(85, 74, 89, 78)
end;

{ Đóng chương trình nếu nút nhấn được bấm. }
procedure TForm1.Button1Click(Sender: TObject);
begin
    close
end;

end.
```

8) Ví dụ sử dụng For Loop (Vòng lặp FOR)

{ **EXAMPLE** Sự lặp lại, sử dụng 1 For Loop, được trình diễn ở đây. Vẽ văn bản lên một canvas cũng được biểu diễn.

Một biến gọi là Counter được sử dụng.
Đó là một biến Integer (số nguyên).

TASK ONE (Beginners)

Làm sự lặp lại đó trong bất cứ
Chương trình nào mà bạn muốn làm.

TASK TWO (Advanced)

Hiển thị 'Hello' thành một hình tròn
Trong form. Sử dụng hàm SIN và COS.

```
}  
unit Unit1;  
  
interface  
  
uses  
    SysUtils, WinTypes, WinProcs,  
    Messages, Classes, Graphics,  
    Controls, Forms, Dialogs,  
    StdCtrls;  
  
type  
    TForm1 = class(TForm)  
        Button1: TButton;  
        procedure Button1Click(Sender: TObject);  
    private  
        { Private declarations }  
    public  
        { Public declarations }  
    end;  
  
var  
    Form1: TForm1;  
  
implementation  
  
{$R *.DFM}  
  
procedure TForm1.Button1Click(Sender: TObject);  
var Counter : Integer;  
begin  
    For Counter := 1 To 10 Do  
        Begin  
            Form1.Canvas.TextOut(Counter * 10, Counter * 10, 'Hello');  
        End  
    end;  
end.
```



9) Ví dụ sử dụng WHILE LOOP (Vòng lặp WHILE)

```
{    EXAMPLE    Một vòng lặp WHILE là sự liên tục. Sự lặp lại
                tiếp tục miễn là giá trị của X1 không phải
                là zero.

                TASK    Viết chương trình chứa đựng một while loop.

                Vòng lặp "while" lặp lại zero or nhiều lần.
                Nó được sử dụng khi số lượng việc lặp lại
                Chưa được biết trước.                                }

unit Unit1;

interface

uses
    SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
    Forms, Dialogs, ExtCtrls, StdCtrls;

type
    TForm1 = class(TForm)
        Image1: TImage;
        ButtonPaint: TButton;
        procedure ButtonPaintClick(Sender: TObject);
        procedure FormCreate(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.ButtonPaintClick(Sender: TObject);
var X1, Y1, X2, Y2 : Integer;
    Colour : TColor;
begin
    X1 := 1; { Chắc chắn rằng X1 không là zero hoặc thủ tục sẽ
              { không bao giờ chạy. Dọn các hình vẽ trước đây. }
    image1.canvas.rectangle(0, 0, image1.width, image1.height);

    { Vẽ những đường ngẫu nhiên với màu ngẫu nhiên cho đến khi X1 = 1 }
    while X1 <> 0 do
    begin
        { Cho một vài giá trị ngẫu nhiên. }
        X1 := random(image1.width);
        Y1 := random(image1.height);
        X2 := random(image1.width);
        Y2 := random(image1.height);

        { Cho một màu ngẫu nhiên. }
        Colour := random($100) + $100 *
```

Những ví dụ về lập trình bằng Delphi; Biên dịch bởi Lê Khắc Như; website: <http://www.lkn.b4.to>
2/21/04

```
random($100) + $10000 *  
random($100);  
  
    { Đặt màu cho bút và vẽ các đường thẳng. }  
    image1.canvas.pen.color := Colour;  
    image1.canvas.moveTo(X1, Y1);  
    image1.canvas.lineTo(X2, Y2);  
end;  
end;  
  
    { Thủ tục này chạy một lần khi chương trình được khởi động. }  
procedure TForm1.FormCreate(Sender: TObject);  
begin  
    { Tạo một số thú ngẫu nhiên. }  
    randomize  
end;  
  
end.
```


10) Ví dụ tạo chương trình LOTO (Số số)

{ **EXAMPLE** Một chương trình chọn số ngẫu nhiên.

TASK Sử dụng chương trình này như ví dụ, viết chương trình để chọn số ngẫu nhiên. Chương trình này cho phép số trùng nhau. Nhìn thấy nếu bạn có thể làm nó. }

unit Unit1;

interface

uses

SysUtils, WinTypes, WinProcs, Messages,
Classes, Graphics, Controls, Forms, Dialogs,
ExtCtrls, StdCtrls;

type

TForm1 = class(TForm)
 Image1: TImage;
 Button1: TButton;
 Timer1: TTimer;
 procedure Button1Click(Sender: TObject);
 procedure Timer1Timer(Sender: TObject);
 procedure FormCreate(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

Procedure MyShow(aNumber : Integer);
end;

var

Form1: TForm1;

implementation

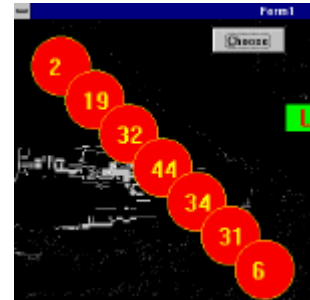
{ \$R *.DFM }

{ ----- }
{
 This procedure selects and displays one
 lottery number. Its position is determined by
 aNumber.
}

procedure TForm1.MyShow(aNumber : Integer);

begin

Image1.Canvas.Pen.Color := clYellow;	{ Gives yellow outline }
Image1.Canvas.Brush.Color := clRed;	{ Gives red circle }
Image1.Canvas.Font.Color := clYellow;	{ Gives yellow text }
Image1.Canvas.Font.Size := 24;	{ Gives bigger text }
	{ Draw the ellipse }
Image1.Canvas.Ellipse(aNumber * 40 - 20, aNumber * 40 - 20,	
aNumber * 40 + 50, aNumber * 40 + 50);	
	{ Display the text }



Những ví dụ về lập trình bằng Delphi; Biên dịch bởi Lê Khắc Như; website: <http://www.lkn.b4.to>
2/21/04

```
Image1.Canvas.TextOut(aNumber * 40,  
                      aNumber * 40, intToStr(random(49) + 1))  
  
end;  
  
{ ----- }  
  
procedure TForm1.Button1Click(Sender: TObject);  
Var Counter : Integer;  
begin  
    For Counter := 1 to 7 Do      { For all seven lottery numbers DO }  
    Begin  
        MyShow(Counter)          { Display the ball                }  
    End  
end;  
  
{ ----- }  
  
procedure TForm1.Timer1Timer(Sender: TObject);  
{  
    This procedure runs once every ten seconds.  
    It displays the word Lottery.  It also changes the  
    colour of the text.  
}  
begin  
    Image1.Canvas.Pen.Color := clLime;  
    Image1.Canvas.Brush.Color := clLime;  
  
    if Image1.Canvas.Font.Color = clRed then  
    begin  
        Image1.Canvas.Font.Color := clYellow  
    end  
    else  
    begin  
        Image1.Canvas.Font.Color := clRed  
    end;  
  
    Image1.Canvas.Font.Size := 24;  
  
    Image1.Canvas.TextOut(Image1.Width div 2, 100, ' Lottery ' )  
end;  
  
{ ----- }  
  
procedure TForm1.FormCreate(Sender: TObject);  
{  
    This procedure runs when the program first starts up.  
    Randomize causes the random number sequence to start  
    on a different random number on each run.  
}  
begin  
    randomize  
end;  
  
{ ----- }  
  
end.  
  
{ ----- }
```

11) Ví dụ về Time Delay (Trì hoãn thời gian)

```
{  EXAMPLE    This program shows how to do a time delay using
              a Delphi System timer.  When you press the button,
              after five seconds, the word Hello appears.

              TASK    Write a program that counts from one to ten at
                      half second time intervals and displays the count.
                      When it reaches ten, make something happen such as
                      a messageBeep(0) or a picture being drawn. }
```

```
unit Unit1;
```

```
interface
```

```
uses
```

```
  SysUtils, WinTypes,
  WinProcs, Messages,
  Classes, Graphics,
  Controls, Forms,
  Dialogs, StdCtrls,
  ExtCtrls;
```

```
type
```

```
  TForm1 = class(TForm)
    Button1: TButton;
    Timer1: TTimer;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
    count : Integer;

    procedure doSomething;
  end;
```

```
var
```

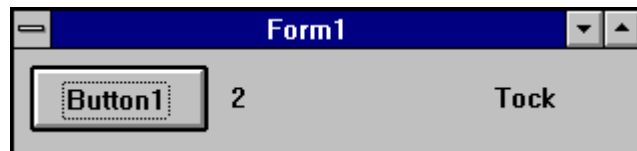
```
  Form1: TForm1;
```

```
implementation
```

```
{ $R *.DFM }
```

```
procedure TForm1.doSomething;
begin
  label1.caption := 'Hello'
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  count := 5;
```



Những ví dụ về lập trình bằng Delphi; Biên dịch bởi Lê Khắc Như; website: <http://www.lkn.b4.to>
2/21/04

```
    label1.caption := ''  
end;  
  
procedure TForm1.FormCreate(Sender: TObject);  
begin  
    count := 0;  
end;  
  
procedure TForm1.Timer1Timer(Sender: TObject);  
begin  
    if count <> 0 then  
        begin  
            count := count - 1;  
            if count = 0 then  
                begin  
                    doSomething  
                end  
            end;  
  
            label2.caption := intToStr(count);  
  
            if label3.caption = 'Tick' then  
                label3.caption := 'Tock'  
            else  
                label3.caption := 'Tick'  
            end;  
  
end.
```

12) Tạo chiếc máy tính đơn giản

```
{ EXAMPLE      Đây là ví dụ về chiếc máy tính đơn giản.
                Nó tính được Cộng và tính Trừ.

                TASK      Học chương trình để hiểu nó và sau đó
                          Thêm vào máy tính thành phần Times, Divide,
                          Square root và nếu muốn bạn làm từ X tới Y
                          Và các hàm khó hơn nữa.
}
{ ===== }
unit MainForm;

interface

uses
  SysUtils, WinTypes, WinProcs,
  Messages, Classes, Graphics,
  Controls, Forms, Dialogs,
  StdCtrls;

{ ===== }

type
  modeType = (none, plus, minus);           { Add more modes here. }

  TForm1 = class(TForm)
    EditDisplay: TEdit;
    ButtonPlus: TButton;
    ButtonMinus: TButton;
    ButtonEquals: TButton;
    ButtonClear: TButton;
    procedure FormCreate(Sender: TObject);
    procedure ButtonClearClick(Sender: TObject);
    procedure ButtonPlusClick(Sender: TObject);
    procedure ButtonEqualsClick(Sender: TObject);
    procedure ButtonMinusClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations } { You type these in. }
    mode      : modeType;
    resultSoFar : Real;

    procedure calculate;
  end;

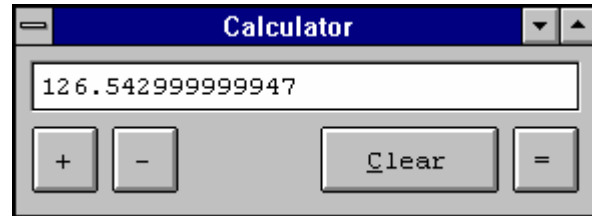
{ ===== }

var
  Form1: TForm1;

implementation

{$R *.DFM}

{ ===== }
```



```
{ ===== }

procedure TForm1.Calculate;
begin
    { Do a calculation based on the mode. }
    if mode = none then
    begin
        resultSoFar := strToFloat(editDisplay.text);
    end
    else if mode = plus then
    begin
        resultSoFar := resultSoFar + strToFloat(editDisplay.text);
    end
    else if mode = minus then
    begin
        resultSoFar := resultSoFar - strToFloat(editDisplay.text);
    end;

    { Refresh the answer / input display. }
    editDisplay.text := floatToStr(resultSoFar);
    editDisplay.setFocus;
    editDisplay.selectAll;
end;

{ ===== }
{ Ensure that the initial values are OK. }
{ This happens when the form is created. }
procedure TForm1.FormCreate(Sender: TObject);
begin
    resultSoFar := 0.0;
    mode       := none;
end;

{ ===== }
{ CLEAR BUTTON }
{ Clear the displays. }
{ Set the result so far to zero. }
procedure TForm1.ButtonClearClick(Sender: TObject);
begin
    mode := none;
    resultSoFar := 0.0;
    editDisplay.setFocus;
    editDisplay.clear;
end;

{ ===== }
{ PLUS BUTTON }
{ Perform the calculation so far and set the mode to PLUS. }
procedure TForm1.ButtonPlusClick(Sender: TObject);
begin
    calculate;
    mode := plus;
end;

{ ===== }
```

Những ví dụ về lập trình bằng Delphi; Biên dịch bởi Lê Khắc Như; website: <http://www.lkn.b4.to>
2/21/04

```
{ ===== }
{ EQUALS BUTTON }
{ Perform the calculation so far and set the mode to NONE. }
procedure TForm1.ButtonEqualsClick(Sender: TObject);
begin
    calculate;
    mode := none;
end;

{ ===== }
{ MINUS BUTTON }
{ Perform the calculation so far and set the mode to MINUS. }
procedure TForm1.ButtonMinusClick(Sender: TObject);
begin
    calculate;
    mode := minus;
end;

{ ===== }

{ ADD BUTTONS AND PROCEDURES FOR Times, Divide, Square root, Etc. }

end.

{ ===== }
```

13) Ví dụ về đồ họa Fractals

Fractals là một kiểu đồ họa phức tạp không giống với bất kỳ kiểu đồ họa nào. That means you can zoom in on the picture and the small scale looks the same as the large scale picture. This program produces a simple fractal. Nice fern or tree shapes can be produced by modifying the program below. By introducing a small degree of randomness, the shapes can look surprisingly like real trees or ferns.

```
{
```

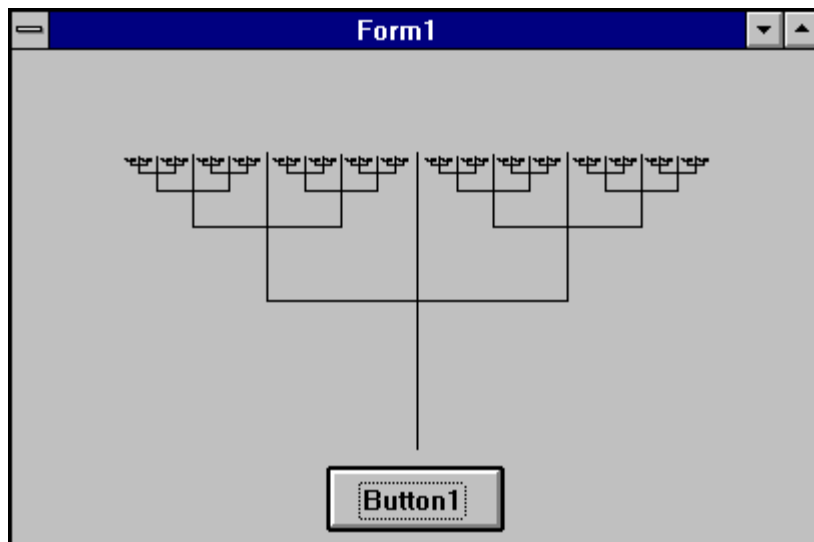
EXAMPLE

A much more complex fractal graphic.

TASK

Tinker with the maths to get different shaped fractals.

```
}
```



```
unit Unit1;
```

```
interface
```

```
uses
```

```
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics,  
  Controls, Forms, Dialogs, StdCtrls, ExtCtrls;
```

```
type
```

```
  TForm1 = class(TForm)  
    PaintBox1: TPaintBox;  
    Button1: TButton;  
    procedure Button1Click(Sender: TObject);
```

```
  private
```

```
    { Private declarations }
```

```
  public
```

```
    { Public declarations }
```

```
    procedure fractal(X, Y, Size : Integer);  
  end;
```

```
var
```

```
  Form1: TForm1;
```


implementation

```
{ $R *.DFM }
```

```
{ Thủ tục này là ĐỆ QUY. Nó có nghĩa là sử dụng lại chính nó để làm  
công việc đó. Những thủ tục Đệ Quy có thể chạy ra khỏi điều khiển  
nếu thiết kế sai. Trước những thủ tục của chính nó, nó phải kiểm  
tra xem có đúng không rồi mới tiếp tục. Trong ví dụ này,  
những hình dạng nhỏ và nhỏ hơn cho đến khi kích thước nó là hai điểm.  
Đó là các điểm nhỏ bởi vì khi trình bày ra màn hình, nó không  
hiển thị các bức ảnh được. }
```

```
procedure TForm1.fractal(X, Y, Size : Integer);
```

```
begin
```

```
    paintBox1.Canvas.MoveTo(X, Y);
```

```
    paintBox1.Canvas.LineTo(X, Y - Size);
```

```
    paintBox1.Canvas.MoveTo(X - Size div 2, Y - Size div 2);
```

```
    paintBox1.Canvas.LineTo(X + Size div 2, Y - Size div 2);
```

```
    if Size > 2 then
```

```
    begin
```

```
        fractal(X - Size div 2, Y - Size div 2, Size div 2);
```

```
        fractal(X + Size div 2, Y - Size div 2, Size div 2)
```

```
    end
```

```
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
begin
```

```
    fractal(200, 200, 150)
```

```
end;
```

```
end.
```

Ví dụ vẽ Circle, Spiral (Vòng tròn và Xoắn ốc)

{ **EXAMPLE** Đồ họa thú vị nào đó được trình bày ở đây.

Những vòng tròn và các đường xoắn ốc được vẽ bằng các điểm được hiển thị trên màn hình.

TASK Vẽ những hình học khác như sóng sin và đồ thị bình phương, lập phương và các hàm khác. }

unit Unit1;

interface

uses

SysUtils, WinTypes, WinProcs,
Messages, Classes, Graphics,
Controls, Forms, Dialogs,
StdCtrls, ExtCtrls;

type

TForm1 = **class**(TForm)
 PaintBox1: TPaintBox;
 Panel1: TPanel;
 ButtonCircle: TButton;
 ButtonSpiral: TButton;

procedure ButtonCircleClick(Sender: TObject);

procedure ButtonSpiralClick(Sender: TObject);

private

 { Private declarations }

public

 { Public declarations }

end;

var

 Form1: TForm1;

implementation

{ \$R *.DFM }

procedure TForm1.ButtonCircleClick(Sender: TObject);

var X, Y, theta : Integer;

begin

for theta := 0 **to** 359 **do** { Bước độ một }

begin

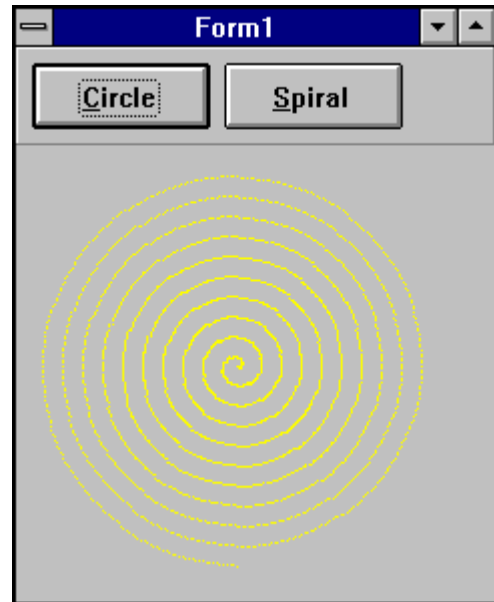
 X := 110 + round(100 * sin(2 * PI * theta / 360));

 Y := 110 + round(100 * cos(2 * PI * theta / 360));

 paintBox1.Canvas.Pixels[X, Y] := clRed;

end

end;



Những ví dụ về lập trình bằng Delphi; Biên dịch bởi Lê Khắc Như; website: <http://www.lkn.b4.to>
2/21/04

```
procedure TForm1.ButtonSpiralClick(Sender: TObject);  
var X, Y, theta : Integer;  
begin  
    for theta := 0 to 3600 do { Bước độ một }  
    begin  
        X := 110 + round((theta div 36) * sin(2 * PI * theta / 360));  
        Y := 110 + round((theta div 36) * cos(2 * PI * theta / 360));  
        paintBox1.Canvas.Pixels[X, Y] := clYellow;  
    end  
end;  
  
end.
```

Tác giả : Lê Khắc Như ; Ngày 24 tháng 1 năm 2004
Email: laptrinh04@yahoo.ca
Website: <http://www.lkn.b4.to>