BÀI THỰC HÀNH 6 PictureBox, Vẽ Đồ Thị

1. Hãy <u>ĐỌC CẨN THÂN TẤT CẢ NỘI DUNG</u> trong bài thực hành trước khi làm bài. 2. Sinh viên <u>TỰ GÕ CODE, KHÔNG COPY</u> từ bài có sẵn, để nắm được việc lập trình.

1 Chuẩn đầu ra

Sau bài này, người học có thể:

- Chuyển đổi được tọa độ giữa hệ tọa độ toán (Oxy) và hệ tọa độ của PictureBox hoặc Form để vẽ đồ thị hàm số.
- Sử dụng được các phương thức vẽ: PSet, Line, Circle.
- Vẽ được các biểu đồ tròn, cột và các đồ thị hàm số được yêu cầu.

2 Chuẩn bị

Đọc lại về control: PictureBox.

Tài liệu tham khảo gọi ý: Chương 9, Giáo trình Visual Basic 6.0, Nguyễn Đăng Quang, Đại học Sư Phạm Kỹ Thuật TP. Hồ Chí Minh.

3 Phương tiện

Máy tính có cài đặt Visual Studio 6.0.

4 Kiến thức bổ trợ

• Hệ màu RGB:

- Hệ màu: là một sự mô tả các màu sắc trong máy tính. Ví dụ trong thực tế, ta mô tả màu bằng từ ngữ, như: Đỏ, Cam, Vàng, Trắng, Đen..., còn trong máy tính, có rất nhiều màu cần phải thể hiện, do đó người ta thường đánh số để chỉ các màu, ví dụ: RGB(255, 0, 0): màu đỏ, RGB(0, 255, 0): màu xanh lá, RGB(255, 255, 255): màu trắng.
- Hệ màu RGB: là hệ màu được dùng phổ biến trong máy tính. Hệ màu này dùng 3 thông số: R (Red Đỏ), G (Green Xanh lá), B (Blue Xanh dương) để chỉ màu sắc. Mỗi thông số có giá trị từ 0-255. Các thông số này kết hợp với nhau tạo thành các màu sắc, giống như khi ta trộn các màu lại được một màu tổng hợp.
 Ví dụ: <u>Red = 255, Green = 0, Blue = 0</u>: màu đỏ, thường được ghi là: RGB(255, 0, 0); hoặc <u>Red = 0, Green = 0, Blue = 255</u>: màu xanh dương (RGB(0, 0, 255)); hay <u>Red = 255, Green = 0</u>: màu vàng (RGB(255, 255, 0)).

• Hệ tọa độ trong PictureBox:

Trong PictureBox, gốc tọa độ là **góc trên, bên trái** của PictureBox với các trục tọa độ có chiều như hình bên dưới. Ta gọi tên hệ tọa độ này là *Ost* (để phân biệt với hệ tọa độ *Oxy* trong toán học (có trục Oy hướng lên)):



Như vậy, trong PictureBox:

- ✓ Tọa độ theo chiều ngang (*Os*) có giá trị từ $0 \rightarrow ScaleWidth$.
- ✓ Tọa độ theo chiều dọc (Ot) có giá trị từ 0 → *ScaleHeight*.

(ScaleWidth và ScaleHeight là thuộc tính chỉ bề rộng và bề cao của PictureBox, xem phần Các thuộc tính của PictureBox)

• Đổi hệ tọa độ sang hệ tọa độ của PictureBox:

Hệ tọa độ trong toán là Oxy (có trục tung Oy hướng lên), còn hệ tọa độ Ost của PictureBox có trục tung Ot hướng xuống. Ngoài ra, 2 hệ tọa độ này còn có điểm đặt gốc tọa độ khác nhau. Vì vậy, để vẽ đồ thị hàm số, ta phải chuyển từ hệ Oxy sang hệ Ost. Xem 2 ví dụ chuyển đổi sau.

Ví dụ 1: Vẽ hàm y = Asin(x), với A là một số dương, x từ $0 \rightarrow 2\pi$.



Đổi tọa độ trục hoành:

 $x: 0 \rightarrow 2\pi$ s: 0 \rightarrow width (width = PictureBox.*ScaleWidth*) $x/2\pi = s/width \rightarrow x = s*2\pi/width$ Đổi tọa độ trục tung:

- y: $-A \rightarrow A$
- t: height $\rightarrow 0$ (height = PictureBox.*ScaleHeight*)
- → Ta thấy y tăng (-A → A), t giảm (height → 0), vì vậy ta nhân t với -1 để t cũng tăng:

y: $-A \rightarrow A$ -t: -height $\rightarrow 0$

→ Bây giờ ta đưa một đầu mút của y và -t về cùng một giá trị (trong bài này ta chọn đưa về 0), bằng cách cộng y với -A:

y-A: $-2A \rightarrow 0$ -t: -height $\rightarrow 0$

 \rightarrow y-A/-2A = -t/-height \rightarrow t = (A-y)*height/2A

Nếu ta chọn A = height/2 thì t = A-y

Ví dụ 2: Vẽ hàm y = -x³ + 3x² - 2, với x từ -a → a hoặc y từ -a → a (tức là gốc tọa độ Oxy nằm chính giữa PictureBox)



Đổi tọa độ trục hoành:

 $x: -a \rightarrow a$

s: $0 \rightarrow$ width (width = PictureBox.*ScaleWidth*)

→ Bây giờ ta đưa một đầu mút của x và s về cùng một giá trị (trong bài này ta chọn đưa về 0), bằng cách cộng x với a:

 $a+x: 0 \rightarrow 2a$ s: 0 \rightarrow width

 $\Rightarrow \qquad (a+x)/2a = s/width \rightarrow x = s*2a/width - a$

Đổi tọa độ trục tung:

->

y: -a → a

t: height $\rightarrow 0$ (height = PictureBox.*ScaleHeight*)

→ Ta thấy y tăng (-a → a), t giảm (height → 0), vì vậy ta nhân t với -1 để t cũng tăng:

y: -a \rightarrow a -t: -height \rightarrow 0

Bây giờ ta đưa một đầu mút của y và –t về cùng một giá trị (trong bài này ta chọn đưa về 0), bằng cách công y với -a:

y-a: $-2a \rightarrow 0$ -t: -height $\rightarrow 0$

 $y-a/-2a = -t/-height \rightarrow t = (a-y)*height/2a$

• Điểm hiện tại (current graphics position) của PictureBox:

Khi vẽ các hình trong PictureBox, VB sẽ tự lưu lại vị trí của lần vẽ sau cùng và gọi nó là Điểm hiện tại. Tọa độ của điểm này được lưu trong thuộc tính *CurrentX, CurrentY* của PictureBox.

Ví dụ: Ban đầu, điểm hiện tại sẽ là gốc tọa độ của PictureBox (CurrentX=0, CurrentY=0). Sau đó, ta vẽ một điểm lên PictureBox thì điểm hiện tại sẽ là tọa độ của điểm này. Sau đó, vẽ tiếp một điểm khác, thì điểm hiện tại là tọa độ điểm mới vẽ. Nếu ta sẽ một đường thẳng thì điểm hiện tại sẽ là điểm đầu cuối của đường thẳng (xem thêm về CurrentX hoặc CurrentY trong MSDN).

- Các thuộc tính của PictureBox:
 - Cls: xóa nội dung đã có trên PictureBox.
 - AutoRedraw = True: tự động vẽ lại nội dung trong PictureBox (ví dụ như sau khi PictureBox bị che khuất hay điều chỉnh kích thước...).
 - ForeColor: xác lập màu dùng để vẽ hình (màu đường viền hình).
 - DrawWidth: chỉ định độ dày của đường viền khi vẽ hình.
 - FillColor: xác lập màu dùng để tô vào trong các hình. Thường dùng hàm RGB() để chỉ định màu.
 - FillStyle: xác lập kiểu tô trong các hình. Ví dụ: VbFSTransparent: trong suốt (**mặc định** của VB), vbFSSolid: tô kín bằng màu, vbVerticalLine: tô bằng các đường kẻ dọc, vbCross: tô bằng đường ca rô (xem thêm trong MSDN).
 - ScaleHeight, ScaleWidth: chiều cao và chiều rộng bên trong của PictureBox (không tính viền của PictureBox).
 - Height, Width: chiều cao và chiều rộng của PictureBox (có tính cả viền).
- Các phương thức vẽ của PictureBox:
 - *PSet* (*x*, *y*), [*color*]: vẽ một điểm (**P**oint **Set**) tại tọa độ (*x*, *y*) với màu *color*.
 - Line (x1, y1) (x2, y2), [color], [B][F]: vẽ đường thẳng từ điểm (x1, y1) đến điểm (x2, y2) với màu color. Nếu không chỉ định điểm (x1, y1) thì VB sẽ lấy điểm hiện tại (CurrentX, CurrentY) làm điểm (x1, y1). Nếu chỉ định thêm B: vẽ hình chữ nhật (Box) với 2 đỉnh là (x1, y1) và (x2, y2). Nếu chỉ định BF: vẽ hình chữ nhật và tô màu vào bên trong (Fill), màu tô là màu được chỉ định bởi thuộc tính FillColor và FillStyle của PictureBox.

Circle (x, y), *radius*, [*color, start, end, aspect*]: vẽ cung tròn hoặc hình tròn, hình ellipse. (x, y) là tọa độ tâm.
 Nếu vẽ hình tròn: *radius* là bán kính, *aspect* = 1.
 Nếu vẽ ellipse: *radius* là bán kính lớn hơn trong 2 bán kính (xem hình bên dưới), *aspect*

= bán kính trục đứng / bán kính trục ngang. Hai hình sau vẽ ellipse có phương trình: $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$



 $a>b \rightarrow radius = a, aspect = b/a$



 $b>a \rightarrow radius = b, aspect = b/a$

Tham số *start, end*: góc bắt đầu và kết thúc của cung tròn (tính bằng radian). VB luôn luôn tính góc theo **chiều dương** (chiều ngược chiều kim đồng hồ) cho dù góc *start* và *end* là số âm. Tuy nhiên, khi chỉ định góc là số âm, VB sẽ vẽ một phần **hình tròn đóng** (có thể tô màu), còn nếu góc là số dương VB chỉ vẽ cung tròn (xem hình bên dưới).





start = 2π , end = $\pi/2$

start = - 2π , *end* = - $\pi/2$

- Các hàm cần biết:
 - RGB(*do*, *xanhla*, *xanhduong*): trả về màu sắc theo hệ RGB với các thông số màu sắc có giá trị *do*, *xanhla*, *xanhduong*. Ví dụ: RGB(0, 0, 255): màu xanh dương.

5 Thời lượng: 4 tiết

6 Nội dung thực hành

Bài thực hành có hướng dẫn chi tiết:

6.1	Vẽ đường ellipse	6
6.2	Vẽ ellipse và tô màu	7
6.3	Vẽ biểu đồ tròn	9
6.4	Vẽ biểu đồ cột	12
6.5	Vẽ đồ thị hàm sin	14
6.6	Vẽ đồ thị hàm bậc 3	15
Bài thự	c hành tự thực hiện:	
7.1	Vẽ đồ thị hàm số lượng giác	17

6.1 VÊ ĐƯỜNG ELLIPSE

6.1.1 Mô tả

Dữ liệu vào: các bán kính a, b của đường ellipse.

Dữ liệu ra: đường ellipse tương ứng.

Điều khiển: nút cmdVe: thực hiện vẽ ellipse.

6.1.2 Thực hiện

- 0. Mở Visual Basic, chọn New Project/Standard EXE, và save vào thư mục Bai27_PictureBox_VeDuongEllipse (tự tạo trước).
- 1. Tạo form như hình:



2. Nhấp đúp vào nút *cmdVe* và nhập code:

```
Private Sub cmdVe_Click()
    Dim a As Integer, b As Integer
    Dim xc As Integer, yc As Integer
    Dim BanKinh As Integer
    a = Val(txtA.Text)
    b = Val(txtB.Text)
    If a > b Then
        BanKinh = a
    Else
        BanKinh = b
    End If
    xc = picl.ScaleWidth / 2
    yc = picl.ScaleHeight / 2
    picl.Cls
    picl.Cls
    picl.Circle (xc, yc), BanKinh, RGB(0, 0, 255), , , b / a
```

End Sub



6.2 VẼ ELLIPSE VÀ TÔ MÀU

6.2.1 Mô tả

Dữ liệu vào: các chỉ định màu sắc và kiểu tô.

Dữ liệu ra: hình ellipe với màu sắc và kiểu tô tương ứng.

Điều khiển: các CheckBox và OptionButton.

6.2.2 Thực hiện

- 0. Mở Visual Basic, chọn New Project/Standard EXE, và save vào thư mục Bai28_PictureBox_VeEllipseVaToMau (tự tạo trước).
- 1. Tạo form như hình:

5	🖏 Ve ellipse va to mau					
	PictureBox:	i i Mau To	Kieu To			
	pic1	∷ 🔽 chkRed	• optSolid			
		∴	C optVerticalLine			
		∴ ∴ □ chkBlue	C optCross			

2. Nhấp đúp vào *form* và nhập code:

Private Sub Ve()

```
Dim R As Integer, G As Integer, B As Integer
Dim kieuto As Integer
Dim w As Integer, h As Integer
```

```
If chkRed.Value = vbChecked Then
        R = 255
    Else
        R = 0
    End If
    If chkGreen.Value = vbChecked Then
        G = 255
    Else
        G = 0
    End If
    If chkBlue.Value = vbChecked Then
       B = 255
    Else
        B = 0
    End If
    If optSolid.Value = True Then
        kieuto = vbFSSolid
    ElseIf optVerticalLine.Value = True Then
        kieuto = vbVerticalLine
    Else
        kieuto = vbCross
    End If
   pic1.Cls
   pic1.AutoRedraw = True
   pic1.ForeColor = RGB(0, 0, 255)
   pic1.DrawWidth = 3
   pic1.FillColor = RGB(R, G, B)
   pic1.FillStyle = kieuto
   w = pic1.ScaleWidth
   h = pic1.ScaleHeight
    pic1.Circle (w / 2, h / 2), w / 2, , , , h / w
End Sub
Private Sub Form_Load()
    Call Ve
End Sub
```

3. Lần lượt nhấp đúp vào các CheckBox và nhập code:

```
Private Sub chkBlue_Click()
Call Ve
End Sub
```

```
Private Sub chkGreen Click()
       Call Ve
   End Sub
   Private Sub chkRed Click()
       Call Ve
   End Sub
4. Lần lượt nhấp đúp vào các OptionButton và nhập code:
   Private Sub optCross Click()
       Call Ve
   End Sub
   Private Sub optSolid Click()
       Call Ve
   End Sub
   Private Sub optVerticalLine Click()
       Call Ve
   End Sub
```

🔁. Ve ellipse		
	Mau To ✓ chkRed chkGreen chkBlue	Kieu To C optSolid C optVerticalLine C optCross

6.3 VĨ BIỂU ĐỒ TRÒN

6.3.1 Mô tả

Dữ liệu vào: số liệu cho 4 phần của biểu đồ.

Dữ liệu ra: biểu đồ tròn với 4 phần tương ứng.

Điều khiển: nút cmdVe: thực hiện vẽ biểu đồ.

6.3.2 Thực hiện

0. Mở Visual Basic, chọn New Project/Standard EXE, và save vào thư mục Bai29_PictureBox_VeBieuDoTron (tự tạo trước).

1. Tạo form như hình:

🖏 Ve bieu do tron	
PictureBox:	Phan 1 txtPhan1
pic	Phan 2 Jut Phan 2
	Phan 3 LutDhav2
	Phan 4 ItxtPhan4
	cmdVe

2. Nhấp đúp vào nút *cmdVe* và nhập code:

```
Private Sub cmdVe_Click()
```

```
Dim phan1 As Integer, phan2 As Integer, phan3 As Integer,
phan4 As Integer
    Dim tong As Integer
    Dim gocdau As Double, goccuoi As Double
    Dim PI As Double
    Dim xc As Integer, yc As Integer
    PI = 4\# * Atn(1\#)
    xc = pic.ScaleWidth / 2
    yc = pic.ScaleHeight / 2
    phan1 = Val(txtPhan1.Text)
   phan2 = Val(txtPhan2.Text)
   phan3 = Val(txtPhan3.Text)
   phan4 = Val(txtPhan4.Text)
    tong = phan1 + phan2 + phan3 + phan4
   pic.Cls
   pic.AutoRedraw = True
   pic.ForeColor = RGB(0, 0, 255)
   pic.DrawWidth = 2
   pic.FillStyle = vbFSSolid
    'Ve Phan 1
   pic.FillColor = RGB(255, 0, 0)
    gocdau = 2 * PI
    goccuoi = (phan1 / tong) * 2 * PI
```

```
pic.Circle (xc, yc), pic.ScaleWidth / 2 - 100, , -gocdau, -
goccuoi
    'Ve Phan 2
   pic.FillColor = RGB(0, 255, 0)
    gocdau = goccuoi
    goccuoi = gocdau + (phan2 / tong) * 2 * PI
   pic.Circle (xc, yc), pic.ScaleWidth / 2 - 200, , -gocdau, -
goccuoi
    'Ve Phan 3
   pic.FillColor = RGB(255, 255, 0)
    gocdau = goccuoi
    goccuoi = gocdau + (phan3 / tong) * 2 * PI
   pic.Circle (xc, yc), pic.ScaleWidth / 2 - 200, , -gocdau, -
goccuoi
    'Ve Phan 4
   pic.FillColor = RGB(255, 0, 255)
    gocdau = goccuoi
   goccuoi = gocdau + phan4 / tong * 2 * PI
   pic.Circle (xc, yc), pic.ScaleWidth / 2 - 200, , -gocdau, -
qoccuoi
```

End Sub



3. Lưu và nhấn nút Start hay F5 để chạy và xem kết quả:

6.4 VẼ BIỂU ĐỒ CỘT

6.4.1 Mô tả

Dữ liệu vào: số liệu cho 4 cột của biểu đồ.

Dữ liệu ra: biểu đồ cột với 4 cột tương ứng.

Điều khiển: nút cmdVe: thực hiện vẽ biểu đồ.

6.4.2 Thực hiện

- 0. Mở Visual Basic, chọn New Project/Standard EXE, và save vào thư mục Bai30_PictureBox_VeBieuDoCot (tự tạo trước).
- 1. Tạo form như hình:

🖏 Ve bieu do cot	_ • •
PictureBox: pic	Cot 1 [txtCot(0) Cot 2 [txtCot(1) Cot 3 [txtCot(2) Cot 4 [txtCot(3) cmdVe

2. Nhấp đúp vào nút *cmdVe* và nhập code:

Private Sub cmdVe_Click()

```
Dim c1 As Integer, c2 As Integer, c3 As Integer, c4 As
Integer
    Dim cmax As Integer
    Dim ChieuRongCot As Integer, ChieuCaoCot As Integer
    c1 = Val(txtCot(0).Text)
    c2 = Val(txtCot(1).Text)
    c3 = Val(txtCot(2).Text)
    c4 = Val(txtCot(3).Text)
    cmax = c1
    If c2 > cmax Then
        cmax = c2
    End If
    If c3 > cmax Then
        cmax = c3
    End If
    If c4 > cmax Then
        cmax = c4
    End If
```

```
pic.Cls
   pic.AutoRedraw = True
   pic.ForeColor = RGB(0, 0, 255)
   pic.DrawWidth = 3
    ChieuRongCot = pic.ScaleWidth / 4
   pic.FillColor = RGB(255, 0, 0)
    'Ve Cot 1
    pic.FillStyle = vbVerticalLine
    ChieuCaoCot = (c1 / cmax) * pic.ScaleHeight
    pic.Line (0, Height)-(ChieuRongCot, pic.ScaleHeight -
ChieuCaoCot), , B
    'Ve Cot 2
    pic.FillStyle = vbHorizontalLine
    ChieuCaoCot = (c2 / cmax) * pic.ScaleHeight
    pic.Line (ChieuRongCot, Height) - (ChieuRongCot * 2,
pic.ScaleHeight - ChieuCaoCot), , B
    'Ve Cot 3
   pic.FillStyle = vbCross
    ChieuCaoCot = (c3 / cmax) * pic.ScaleHeight
    pic.Line (ChieuRongCot * 2, Height)-(ChieuRongCot * 3,
pic.ScaleHeight - ChieuCaoCot), , B
    'Ve Cot 4
    pic.FillStyle = vbDiagonalCross
    ChieuCaoCot = (c4 / cmax) * pic.ScaleHeight
    pic.Line (ChieuRongCot * 3, Height)-(ChieuRongCot * 4,
pic.ScaleHeight - ChieuCaoCot), , B
```

End Sub

3. Lưu và nhấn nút Start hay F5 để chạy và xem kết quả:



6.5 VẼ ĐỒ THỊ HÀM SIN

6.5.1 Mô tả

Dữ liệu vào: không có.

Dữ liệu ra: đồ thị hàm sin.

Điều khiển: không có (đồ thị sẽ được vẽ khi sự kiện Form_Paint (vẽ form) xảy ra).

6.5.2 Thực hiện

- 0. Mở Visual Basic, chọn New Project/Standard EXE, và save vào thư mục Bai31_PictureBox_VeDoThiHamSin (tự tạo trước).
- 1. Tạo form như hình:

🖏 Ve do thi ham sin	
PictureBox:	:
: pic1	:
· ·	
	:
	:
	:
	:
	:
••	
	:
	:
	:
	:

2. Nhấp đúp vào *form* và nhập code:

```
Private Sub Form_Paint()
   Dim x As Double, y As Double
   Dim s As Integer, t As Integer
   Dim A As Integer
   Dim width As Integer, height As Integer
   Dim PI As Double
   Dim SoChuKy As Integer
   width = picl.ScaleWidth
   height = picl.ScaleHeight
   A = height \ 2
   PI = 4 * Atn(1)
   picl.AutoRedraw = True
```

```
'Ve truc Ox
picl.Line (0, height / 2)-(width, height / 2)
picl.DrawWidth = 3
'Ve 2 chu ky
SoChuKy = 2
For s = 0 To width Step 1
    x = SoChuKy * 2 * PI * s / width
    y = A * Sin(x)
    t = A - y
    picl.PSet (s, t), RGB(255, 0, 0)
Next
End Sub
```


6.6.1 Mô tả

Dữ liệu vào: hàm bậc 3: $y = -x^3 + 3x^2 - 2$

Dữ liệu ra: đồ thị hàm bậc 3: $y = -x^3 + 3x^2 - 2$, với x có giá trị từ $-a \rightarrow a$ hoặc y từ $-a \rightarrow a$ (a là số dương tùy ý).

Điều khiển: không có (đồ thị sẽ được vẽ khi sự kiện Form_Paint (vẽ form) xảy ra).

6.6.2 Thực hiện

- 0. Mở Visual Basic, chọn New Project/Standard EXE, và save vào thư mục Bai32_PictureBox_VeDoThiHamBac3 (tự tạo trước).
- 1. Không cần thiết kế form (vì vẽ trực tiếp lên form).

```
2. Nhấp đúp vào form và nhập code:
```

```
Private Sub Form Paint() 'CHÚ Ý: Sự kiện là Form Paint
    Dim x As Double, y As Double
    Dim a As Double
    Dim s As Long, t As Long
    Dim width As Integer, height As Integer
    width = Me.ScaleWidth 'Me: form dang chay.
    height = Me.ScaleHeight
    'Vẽ 2 trục tọa độ: Ox, Oy
    Me.ForeColor = RGB(0, 0, 0)
    Me.DrawWidth = 1
    'Ox
    Me.Line (0, height / 2)-(width, height / 2)
    'Ov
    Me.Line (width / 2, 0)-(width / 2, height)
    'Vẽ đồ thị hàm số: y = -x^3 + 3x^2 - 2,
    'với x từ -a \rightarrow a hoặc y từ -a \rightarrow a
    Me.ForeColor = RGB(0, 0, 255)
    Me.DrawWidth = 2
    a = 7
    'Dùng 1 trong 2 cách sau để vẽ:
    'CÁCH 1: cho s chạy (từ 0 -> width)
     For s = 0 To width Step 1
T
ı.
         x = 2 * a * s / width - a
         y = -x * x * x + 3 * x * x - 2
         t = (a - y) * height / (2 * a)
         Me.PSet (s, t)
1
     Next
    'CÁCH 2: cho x chay (từ -a -> a)
    For \mathbf{x} = -\mathbf{a} \mathbf{T} \mathbf{o} \mathbf{a} Step 0.001
        y = -x * x * x + 3 * x * x - 2
        s = (a + x) * width / (2 * a)
        t = (a - y) * height / (2 * a)
        Me.PSet (s, t)
    Next
End Sub
```


7 Bài tập đề nghị

7.1 Vẽ đồ thị hàm số lượng giác

Mô tả: Viết chương trình vẽ đồ thị hàm số lượng giác sau:

$$y = \frac{(x+1)\sin(x)}{\sin(x) + x\cos(x)}$$

với x từ $-2*2\pi \rightarrow 2*2\pi$ hoặc y từ $-2*2\pi \rightarrow 2*2\pi$

Hướng dẫn: Làm tương tự bài **6.6. Vẽ đồ thị hàm bậc 3**. Đồ thị kết quả được cho trong hình bên dưới.

_HÉT_____