

MÔN : LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Bài thực hành số 7.1 : Xây dựng chương trình đồng hồ có giao diện truyền thống

I. Mục tiêu :

- Giúp SV làm quen với cách thức viết code để hiển thị 3 loại dữ liệu cơ bản cấu thành bất kỳ thông tin cần xuất : chuỗi văn bản, hình bitmap, hình đồ họa toán học.

II. Nội dung :

- Viết code để hiển thị 3 loại dữ liệu cơ bản : chuỗi văn bản, hình bitmap, hình đồ họa toán học và quan sát kết quả của từng đoạn code.



III. Chuẩn đầu ra :

- Sinh viên nắm vững và lập trình thành thạo các đoạn code để hiển thị 3 loại dữ liệu cơ bản : chuỗi văn bản, hình bitmap, hình đồ họa toán học.



IV. Qui trình :

0. Dùng chương trình soạn thảo đồ họa như Paint, CorelDraw, Photoshop vẽ ảnh bitmap của khung đồng hồ, mặt đồng hồ như hình dưới đây rồi cắt lên file c:\Images\DonghoTho.bmp.



1. Chạy VS .Net, chọn menu File.New.Project để hiển thị cửa sổ New Project.
2. Mở rộng mục Visual C# trong TreeView "Project Types", chọn mục Windows, chọn icon "Windows Forms Application" trong listbox "Templates" bên phải, thiết lập thư mục chứa Project trong listbox "Location", nhập tên Project vào textbox "Name:" (td. Form_Dongho), click button OK để tạo Project theo các thông số đã khai báo.
3. Form đầu tiên của ứng dụng đã hiển thị trong cửa sổ thiết kế, lúc này form hoàn toàn trống, chưa chứa đối tượng giao diện nào.
4. Chọn form để hiển thị cửa sổ Properties của nó, duyệt tìm mục BackgroundImage, click button ... bên phải để hiển thị cửa sổ "Select Resource", đánh dấu chọn RadioButton "Project Resource File", Click button Import, duyệt tìm file chứa ảnh nền của đồng hồ rồi chọn button Open để máy


import tài nguyên này vào Project. Xem kết quả hiển thị, nếu đúng thì chọn button OK để máy import thực sự tài nguyên vào Project.

5. File ảnh nền sẽ được import vào cây Project và được truy xuất theo đường dẫn nội bộ là Form_Dongho.Resources.DonghoTho.bmp. Chọn mục miêu tả ảnh nền vừa import để hiển thị cửa sổ Properties của nó, thay đổi trị thuộc tính Build Action = Embedded Resource.
6. Nếu kích thước ảnh nhỏ hơn form thì ta sẽ thấy ảnh được hiển thị lặp lại theo chiều ngang và dọc. Hãy hiệu chỉnh thuộc tính BackgroundImageLayout = None để máy hiển thị ảnh nền 1 lần ở vị trí trên trái Form.
7. Nếu cửa sổ ToolBox chưa hiển thị chi tiết, chọn menu View.Toolbox để hiển thị nó (thường nằm ở bên trái màn hình). Click chuột vào button  (Auto Hide) nằm ở góc trên phải của cửa sổ ToolBox để chuyển nó về chế độ hiển thị thường trực. Duyệt tìm phần tử Timer (trong nhóm Components hay nhóm All Window Forms), chọn nó, dờ chuột vào trong form (ở vị trí nào cũng được vì đối tượng này không được hiển thị) và vẽ nó với kích thước tùy ý. Hiệu chỉnh thuộc tính (Name) = myTimer, Interval = 1000 (1giây).
7. Chọn đối tượng myTimer, cửa sổ thuộc tính của nó sẽ hiển thị, click icon  để hiển thị danh sách các sự kiện của đối tượng, ấn kép chuột vào comboBox bên phải sự kiện Tick để máy tạo tự động hàm xử lý cho sự kiện này.

8. Viết code cụ thể cho hàm như sau :

```
//hàm phục vụ Timer
```

```
private void myTimer_Tick(object sender, EventArgs e) {
    myTimer.Stop();           //dừng đếm timer
    this.Refresh(); //vẽ lại form theo giờ hiện hành
}
```

9. Ấn phải chuột vào mục Form1.cs trong cửa sổ Solution Explorer rồi chọn option View Designer để hiển thị lại cửa sổ thiết kế Form. Chọn Form, cửa sổ thuộc tính của nó sẽ hiển thị, click icon  để hiển thị danh sách các sự kiện của Form, duyệt tìm sự kiện Paint, ấn kép chuột vào comboBox bên phải sự kiện Paint để máy tạo tự động hàm xử lý cho sự kiện này. Viết code cụ thể cho hàm như sau :

```
private void Form1_Paint(object sender, PaintEventArgs e) {
    //xác định đối tượng mục tiêu
    Control control = (Control)sender;
    //thay đổi kích thước form theo ảnh khung đồng hồ
    control.ClientSize = new Size(bgimg.Width, bgimg.Height);
    //xác định đối tượng graphics (đối tượng vẽ) của đối tượng
    Graphics g = e.Graphics;
    //vẽ bitmap miêu tả nền đồng hồ nếu chưa khai báo ảnh nền cho Form
    g.DrawImage(bgimg, 0, 0);
    //định nghĩa các biến cần dùng
    Pen hPen;
    int x, y;
    double goc;
    //thiết lập tâm đồng hồ
    xo = bgimg.Width / 2 - 1; yo = xo + 33;
    //thiết lập bán kính cần lắc, kim giờ/phút/giây
    rh = bgimg.Width / 2 - 49; rm = bgimg.Width / 2 - 44; rs = bgimg.Width / 2 - 40;
    //tạo Brush và vẽ tâm đồng hồ
    Brush hBrush = new SolidBrush(Color.FromArgb(177, 37, 156));
    g.FillEllipse(hBrush, xo - 4, yo - 4, 8, 8);
}
```

```

//tạo pen để vẽ kim giờ
hPen = new Pen(Color.FromArgb(0, 0, 255), 3);
DateTime now = DateTime.Now;
//tính góc của kim giờ
goc = 270 + 360 * (now.Hour + (double)now.Minute / 60) / 12;
//đổi góc từ độ ra radian
goc = goc * 3.1416 / 180;
//xác định tọa độ đỉnh thứ 2 của kim giờ
x = xo + (int)(rh * Math.Cos(goc));
y = yo + (int)(rh * Math.Sin(goc));
//vẽ kim giờ
g.DrawLine(hPen, xo, yo, x, y);
//tạo pen để vẽ kim phút
hPen = new Pen(Color.FromArgb(0, 255, 0), 2);
//tính góc của kim phút
goc = 270 + 360 * now.Minute / 60;
//đổi góc từ độ ra radian
goc = goc * 3.1416 / 180;
//xác định tọa độ đỉnh thứ 2 của kim phút
x = xo + (int)(rm * Math.Cos(goc));
y = yo + (int)(rm * Math.Sin(goc));
//vẽ kim phút
g.DrawLine(hPen, xo, yo, x, y);
//tạo pen để vẽ kim giây
hPen = new Pen(Color.FromArgb(237, 5, 220), 1);
//tính góc của kim giây
goc = 270 + 360 * now.Second / 60;
//đổi góc từ độ ra radian
goc = goc * 3.1416 / 180;
//xác định tọa độ đỉnh thứ 2 của kim giây
x = xo + (int)(rs * Math.Cos(goc));
y = yo + (int)(rs * Math.Sin(goc));
//vẽ kim giây
g.DrawLine(hPen, xo, yo, x, y);
//tạo chuỗi miêu tả giờ/phút/giây hiện hành
String buf = String.Format("{0:d2}:{1:d2}:{2:d2}", now.Hour, now.Minute, now.Second);
//tạo đối tượng font chữ cần dùng
Font myFont = new Font("Helvetica", 11);
//tạo biến miêu tả chế độ canh giữa khi xuất chuỗi
StringFormat format1 = new StringFormat(StringFormatFlags.NoClip);
format1.Alignment = StringAlignment.Center;
//xuất chuỗi miêu tả giờ/phút/giây
g.DrawString(buf, myFont, System.Drawing.Brushes.White, xo, bgimg.Height - 29, format1);
//cho phép timer chạy tiếp
myTimer.Start();
}

```

10. Tương tự như bước 9, hãy tạo hàm xử lý sự kiện Load cho Form rồi viết code cho hàm này như sau :

```

private void Form1_Load(object sender, EventArgs e)
{
    //tạo đối tượng quản lý assembly

```

```

        System.Reflection.Assembly myAssembly =
System.Reflection.Assembly.GetExecutingAssembly();
        //tạo đối tượng Stream miêu tả ảnh bitmap
        Stream myStream =
myAssembly.GetManifestResourceStream("Form_Dongho.Resources.DonghoTho.bmp");
        //tạo đối tượng ảnh bitmap nền để dùng khi cần
        bgimg = new Bitmap(myStream);
    }

```

11. Dời cursor nhập liệu về đầu class Form1 rồi viết các lệnh định nghĩa các thuộc tính cần dùng như sau :

```

//định nghĩa các biến cần dùng để quản lý Form
Bitmap bgimg; //định nghĩa biến chứa ảnh nền
int xo, yo; //tọa độ tâm đồng hồ
int rh, rm, rs; //bán kính các kim giờ/phút/giây

```

12. Thêm lệnh import namespace sau vào đầu file :

```
using System.IO;
```

13. Chọn menu Debug.Start Debugging để dịch và chạy ứng dụng. Xem kết quả và đánh giá kết quả. Nếu tọa độ hay kích thước các cây kim và chuỗi chưa hiển thị đúng vị trí thì hiệu chỉnh lại giá trị cho tốt hơn.