1.4.4. MonthCalendar và DateTimePicker

Điều khiển MonthCalendar biểu diễn một giao diện trực quan dạng lưới chứa các ngày trong tháng được biểu diễn theo tuần theo thời gian thực, cho phép người dùng chọn một ngày hay một vài ngày bất kỳ trong giới hạn thời gian thiết lập sẵn (Hình 1.37). Một số thuộc tính của MonthCalendar được mô tả trong Bảng 1.24. Điều khiển MonthCalendar thường sử dụng sự kiện DateSelected và DateChanged để xử lý hành động khi người dùng click chọn ngày.

Tương tự, điều khiển DateTimePicker cho phép người dùng chọn ngày trong khoảng xác định thông qua giao diện đồ họa dạng lịch. Giao diện của DateTimePicker là sự kết hợp của hai điều khiển là ComboBox và MonthCalendar. DateTimePicker biểu diễn ngày tháng bởi các đối tượng thuộc lớp DateTime (Hình 1.38). Một số thuộc tính của DateTimePicker được mô tả trong Bảng 1.25. Điều khiển DateTimePicker thường sử dụng sự kiện ValueChanged để xử lý hành động khi người dùng click chọn.

1.4.5. Timer và ProgressBar Điều khiển Timer cho phép thực thi hành động nào đó sau một khoảng thời gian xác định thông qua sự kiện Tick (Hình 1.39). Nghĩa là Timer sẽ tick một cách tự động (theo số miligiây của thuộc tính Interval) khi đã gọi phương thức Start(), khi Timer tick, người dùng có thể cho chương trình thực hiện một công việc nào đó. Một số thuộc tính và phương thức của Timer được mô tả trong bảng 1.27.

Điều khiển ProgressBar dùng để hiển thị tiến độ thực hiện của một công việc nào đó (Hình 1.40), điều khiển này thường được kết hợp với Timer. Một số phương thức và thuộc tính được mô tả như trong Bảng 1.28.

Ví dụ, để thiết kế Form như Hình 1.41, thiết lập để khi bấm vào nút OK thì ProgressBar sẽ chạy theo Step được thiết lập là 10, Minimum = 0, Maximum = 20. Tiến độ chạy của ProgressBar sẽ được thể hiện trên Label thông qua sự kiện Tick của Timer, với Interval của Timer là 1000.

A screenshot of a computer

Description automatically generated

PictureBox là điều khiển dùng để hiển thị dữ liệu dạng hình ảnh như: Bitmap, icon, jpeg, gif… . Điều khiển này chủ yếu sử dụng thuộc tính Image để thiết lập hình ảnh khi thiết kế và chạy chương trình (Hình 1.42). Bảng 1.29 giới thiệu một số thuộc tính cơ bản của điều khiển này.

ImageList là một điều khiển đặc biệt, là một tập hợp hình ảnh có kích thước xác định để cung cấp cho các điều khiển khác sử dụng, ví dụ như: Button, ComboBox, ListView, TreeView… (Hình 1.43)

Lập trình viên tiến hành thêm hoặc xóa hình ảnh trong ImageList thông qua cửa sổ Images Collection Editor, bằng cách click chọn Add hoặc Remove. Chọn thuộc tính Images của điều khiển ImageList để hiển thị cửa sổ Images Collection Editor (Hình 1.44). Một số thuộc tính thường dùng trong ImageList được mô tả trong Bảng 1.30.

Để sử dụng ImageList cho một điều khiển nào đó, ta khai báo thuộc tính Image của điều khiển này là ImageList đã tạo, sau đó thiết lập các item/node của điều khiển này với các ImageIndex tương ứng trong ImageList. Ví dụ, xây dựng Form như Hình 1.45: Khi người dùng chọn hình muốn hiển thị chứa trong ComboBox thì hình tương ứng sẽ xuất hiện trong PictureBox. Kéo các điều khiển ComboBox, PictureBox, và ImageList vào giao diện, thiết lập các thuộc tính cho các điều khiển như sau:

A computer screen shot of a computer screen

Description automatically generated

Các dạng truyền tham số cho phương thức Show thường dùng được minh họa như sau: public static DialogResult Show(string text); public static DialogResult Show(string text, string caption); public static DialogResult Show(string text, string caption, MessageBoxButtons buttons); public static DialogResult Show(string text, string caption, MessageBoxButtons buttons, MessageBoxIcon icon); public static DialogResult Show(string text, string caption, MessageBoxButtons buttons, MessageBoxIcon icon, MessageBoxDefaultButton defaultButton, MessageBoxOptions option);

Ví dụ: Để hỏi người dùng có muốn thoát chương trình hay không, nếu người dùng nhấn OK thì thoát (Hình 1.48), đoạn mã được viết như sau:

if(MessageBox.Show("Lỗi chương trình, bạn muốn thoát?", "Error",

MessageBoxButtons.OKCancel, MessageBoxIcon.Error)== DialogResult.OK)

{

Application.Exit();

A screen shot of a computer

Description automatically generated}

Tạo sự kiện FormLoad để thêm các Item vào ListView như sau, chạy chương

trình để xem kết quả như Hình 1.52.

private void frmHocSinh\_Load(object sender, EventArgs e)

{

ListViewItem lvitem = new ListViewItem(new string[] { "001",

"Nguyễn Văn Hùng", "18", "Nam", "CTK43" });

this.lvDanhSach.Items.Add(lvitem);

lvitem = new ListViewItem(new string[] { "002", "Trương Thị Lệ",

"20", "Nữ", "CTK41" });

this.lvDanhSach.Items.Add(lvitem);

lvitem = new ListViewItem(new string[] { "003", "Nguyễn Hoàng

Nam", "19", "Nam", "CTK42" });

this.lvDanhSach.Items.Add(lvitem);

lvitem = new ListViewItem("004");

lvitem.SubItems.Add("Lê Thị Lan Hương");

lvitem.SubItems.Add("22");

lvitem.SubItems.Add("Nữ");

lvitem.SubItems.Add("CTK40"); lvDanhSach.Items.Add(lvitem); }

A computer screen shot of a program

Description automatically generated

private void btnOpen\_Click(object sender, EventArgs e)

{

openFiledlg.Title = "Open Dialog";

openFiledlg.Multiselect = true;

openFiledlg.Filter = "Image Files (JPEG, GIF, BMP, etc.)|"

+ "\*.jpg;\*.jpeg;\*.gif;\*.bmp;\*.tif;\*.tiff;\*.png|"

+ "JPEG files (\*.jpg;\*.jpeg)|\*.jpg;\*.jpeg|"

+ "GIF files (\*.gif)|\*.gif| BMP files (\*.bmp) | \*.bmp | "

+ "TIFF files (\*.tif;\*.tiff)|\*.tif;\*.tiff|"

+ "PNG files (\*.png)|\*.png| All files (\*.\*)|\*.\*";

if (openFiledlg.ShowDialog() == DialogResult.OK)

{

string[] filenames = openFiledlg.FileNames;

for (int i = 0; i < filenames.Length; i++)

this.lvListFile.Items.Add(new ListViewItem(filenames[i]));

}

}

private void btnSave\_Click(object sender, EventArgs e)

{

saveFiledlg.Title = "Save file ..";

saveFiledlg.Filter = "Image Files (JPEG,BMP,GIF,..)|(\*.jpeg;\*.jpg;)| "

+ "Bitmap files (\*.bmp)|\*.bmp| All files (\*.\*) | \*.\* ";

if (saveFiledlg.ShowDialog() == DialogResult.OK)

{

try

{

System.Drawing.Image im = pbImage.Image;

im.Save(saveFiledlg.FileName, ImageFormat.Bmp);

}

catch { }

}

}

private void listView1\_SelectedIndexChanged(object sender, EventArgs e)

{

int i = lvListFile.SelectedItems.Count - 1;

if (i >= 0)

{

ListViewItem lvitem = lvListFile.SelectedItems[0];

this.pbImage.ImageLocation = lvitem.Text;

}

}

}

}A screenshot of a computer

Description automatically generated

Ví dụ, viết chương trình hiển thị phím và các thông tin phím bổ trợ mà người dùng bấm trên bàn phím (Hình 1.73), trước tiên ta lấy thông tin phím bấm, sau đó lấy thông tin phím bỗ trợ. Hai A computer screen shot of a computer program

Description automatically generatedthao tác trên được viết trong các sự kiện sau:

Chương này đã mang đến cho người đọc cái nhìn tổng quan về lập trình giao diện, cụ thể là ứng dụng Windows Form trong bộ công cụ Visual Studio .NET, cung cấp cho người đọc những kiến thức cơ bản về các điều khiển, sự kiện thường dùng trong Windows Application, đi cùng với đó là những ví dụ minh họa đơn giản và cụ thể giúp cho người đọc có thể đọc hiểu và thực hành một cách dễ dàng.

