

LẬP TRÌNH WINDOWS

Nội Dung

- Xử lý ngoại lệ
- Multiple Document Interface (MDI)
- ImageList
- TreeView

Xử lý ngoại lệ

- **Ngoại lệ (Exception):** các lỗi xuất hiện khi chương trình đang thực thi → **dừng chương trình**

```
ds = new DataSet();
daSinhVien = new SqlDataAdapter("SELECT * FROM SINH_VIEN", ChuoiKetNoi);
dalop = new SqlDataAdapter("SELECT * FROM LOP", ChuoiKetNoi);
// Đổ dữ liệu vào DataTable SinhVien
daSinhVien.Fill(ds, "SinhVien");
dalop.Fill(ds, "Lop");
```

```
TreeNode root = new TreeNode("Bộ  
treeView1.Nodes.Add(root);
```

```
foreach(DataRow row in ds.Tables[0].Rows)
{
    TreeNode node = new TreeNode(
        row[0].ToString());
    root.Nodes.Add(node);
}
```

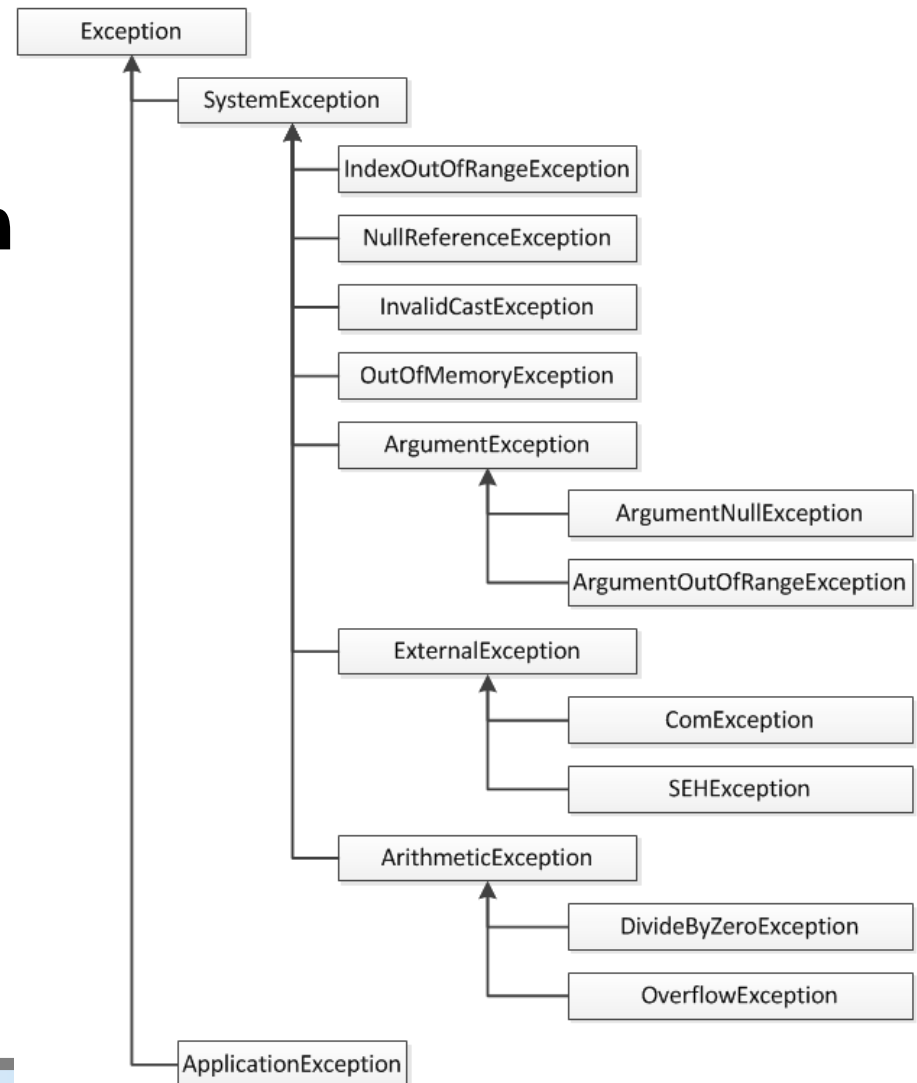
	Value
	{System.Data.SqlClient.SqlDataAdapter}
	{System.Data.DataSet}
	{DemoMDI.Form2, Text: Quản Lý Sinh Viên}

atch 1

The image shows a screenshot of the Visual Studio error dialog box. At the top, there is a yellow warning icon followed by the text "SqlException was unhandled". Below this, the main text reads: "An unhandled exception of type 'System.Data.SqlClient.SqlException' occurred in System.Data.dll". To the right of this text is a vertical scrollbar. Below the main text, there is a section titled "Additional information:" followed by the text: "A network-related or instance-specific error occurred while establishing". Below this section is another vertical scrollbar. The next section is titled "Troubleshooting tips:" and contains two blue links: "Get general help for this exception." and "Get general help for the inner exception.". To the right of these links is a vertical scrollbar. Below the troubleshooting tips is a blue link that says "Search for more Help Online...". The next section is titled "Exception settings:" and contains a checkbox labeled "Break when this exception type is thrown". Below this section is another vertical scrollbar. The final section is titled "Actions:" and contains three blue links: "View Detail...", "Copy exception detail to the clipboard", and "Open exception settings".

Xử lý ngoại lệ

- Lớp **Exception** và một số lớp con



Xử lý ngoại lệ

- Xử lý ngoại lệ trong C#:

```
try
{
    // Khối xử lý chính
}
catch (ExceptionName e1)
{
    // Khối xử lý loại ngoại lệ thứ 1
}
catch (ExceptionName e2)
{
    // Khối xử lý loại ngoại lệ thứ 2
}
catch (ExceptionName eN)
{
    // Khối xử lý loại ngoại lệ thứ N
}
[finally
{
    // Khối xử lý sau khi đã xử lý ngoại lệ
}]
```

Xử lý ngoại lệ

- **Xử lý ngoại lệ trong C#:**

```
try
{
    // Khối xử lý chính
}
catch (Exception ex)
{
    // Khối xử lý chung cho các loại ngoại lệ
}
[finally
{
    // Khối xử lý sau khi đã xử lý ngoại lệ
}]
```

Xử lý ngoại lệ

- **Xử lý ngoại lệ trong C#:**

```
try
{
    // Khối xử lý chính
}
catch
{
    // Khối xử lý chung cho các loại ngoại lệ
}
[finally
{
    // Khối xử lý sau khi đã xử lý ngoại lệ
}]
```

Xử lý ngoại lệ

- **Tạo ngoại lệ trong C#:**

`throw new Exception(<thông tin ngoại lệ>);`

Multiple Document Interface

- Ứng dụng MDI có duy nhất một cửa sổ cha (**MDI Parent**) và có thể có nhiều cửa sổ con (**MDI Child**)
- Một số ưu điểm của ứng dụng MDI:
 - Các cửa sổ con được quản lý bởi một cửa sổ cha duy nhất
 - Hệ thống Menu (**MenuStrip**), Thanh công cụ (**ToolStrip**) được sử dụng chung cho nhiều cửa sổ con
 - Đóng cửa sổ cha thì các cửa sổ con cũng được đóng lại

Multiple Document Interface

- Thiết lập Form thành MDI:
IsMdiContainer → True
- Một số thuộc tính của MDI Parent:
 - **Form** ActiveMdiChild // trả về MDI Child Form đang được kích hoạt
 - **bool** IsMdiContainer // xác định form là MDI Parent hoặc không
 - **Form[]** MdiChildren // trả về mảng chứa các MDI Child Form

Multiple Document Interface

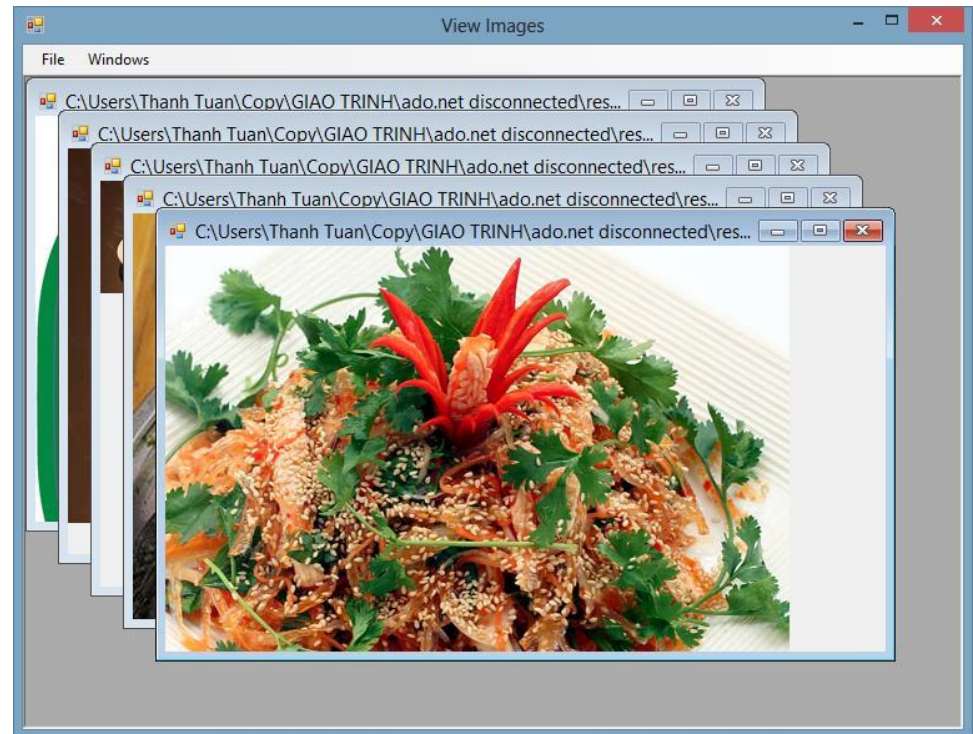
- Một số thuộc tính của MDI Child:
 - `Form MdiParent` // trả về MDI Parent Form
 - `bool IsMdiChild` // xác định form là MDI Child hoặc không
- Phương thức:
 - `void LayoutMdi(MdiLayout value)` // xác định cách hiển thị các MDI Child Form

Multiple Document Interface

- Một số thuộc tính của MDI Child:
 - `Form MdiParent` // trả về MDI Parent Form
 - `bool IsMdiChild` // xác định form là MDI Child hoặc không
- Phương thức:
 - `void LayoutMdi(MdiLayout value)` // xác định cách hiển thị các MDI Child Form

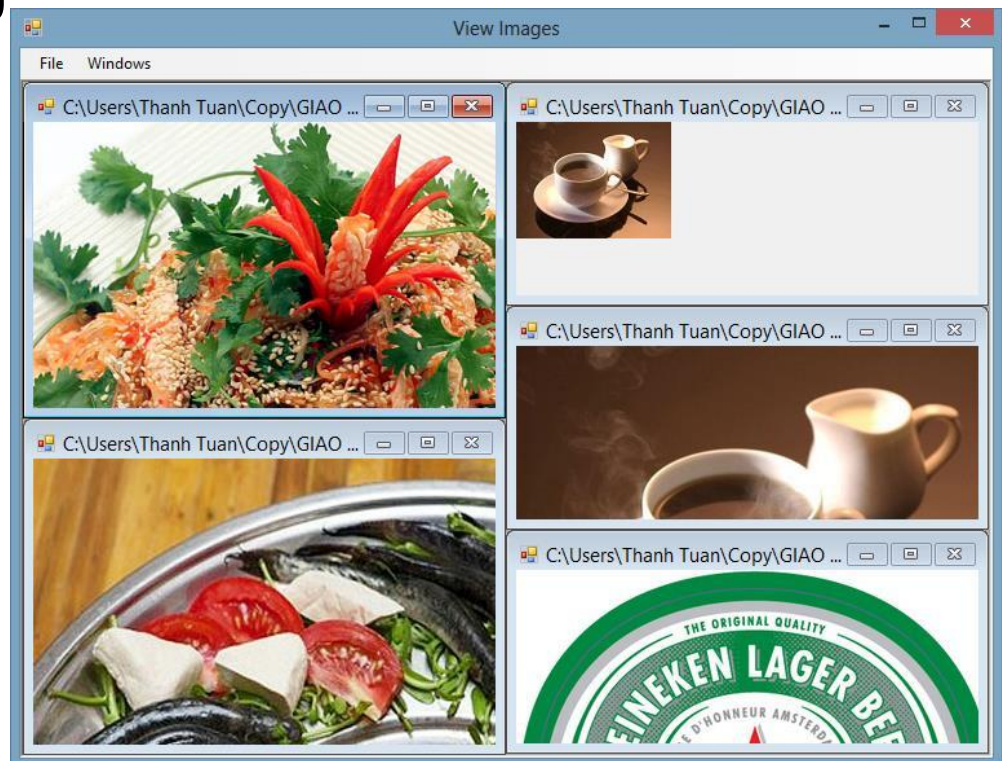
Multiple Document Interface

- Một số cách hiển thị Child Form:
 - **Cascade**: sắp xếp các cửa sổ Child Form theo dạng hình mái ngói



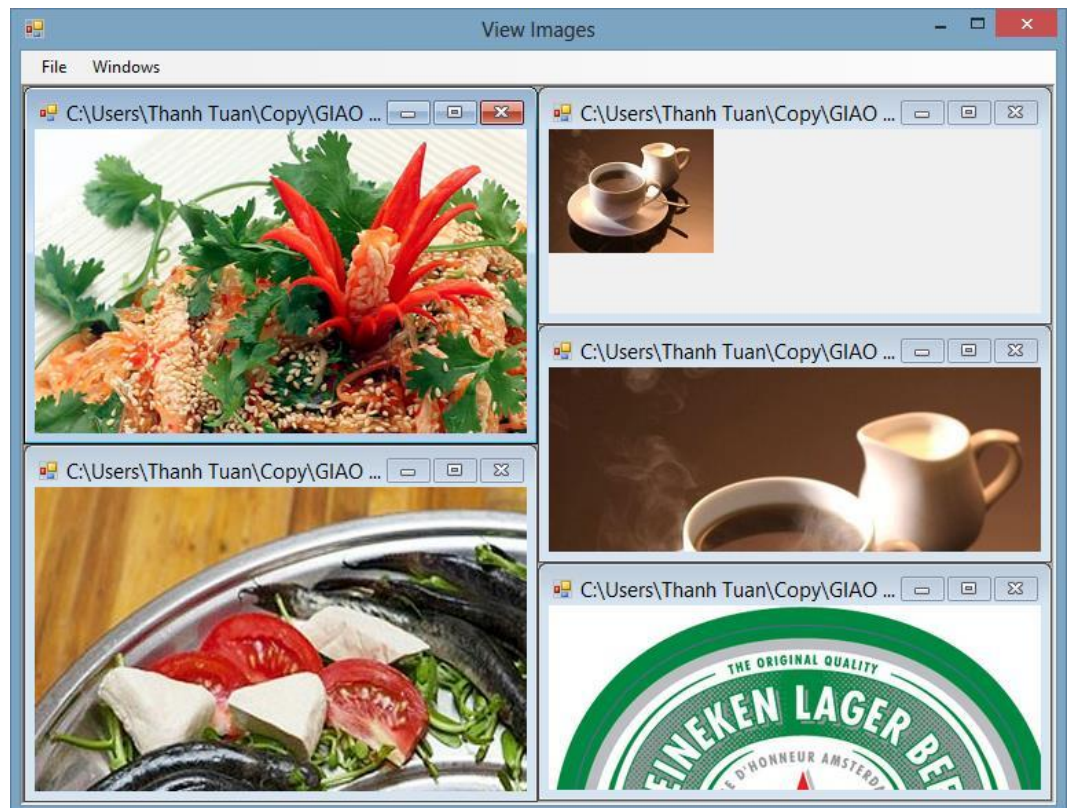
Multiple Document Interface

- Một số cách hiển thị Child Form:
 - **TileHorizontal**: sắp xếp các cửa sổ Child Form theo chiều ngang



Multiple Document Interface

- Một số cách hiển thị Child Form:
 - **TileVertical**: sắp xếp các cửa sổ Child Form theo chiều dọc



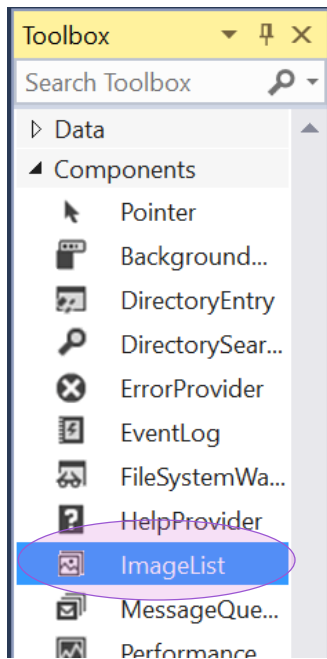
ImageList

- Quản lý danh sách các đối tượng **Image** (hình ảnh)

ImageList

Tạo đối tượng ImageList

Design:



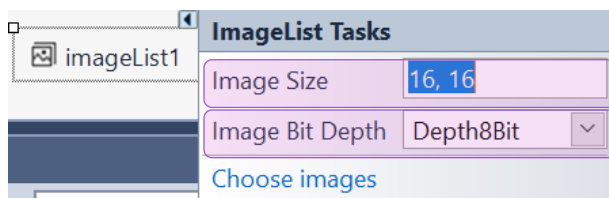
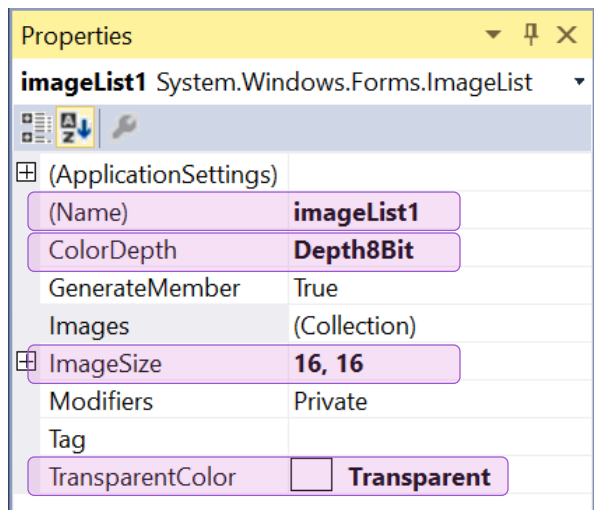
Code:

```
ImageList <tên_biến> = new ImageList();
```

ImageList

Thiết lập các thuộc tính

Design:



Code:

```
<tên_biến>.ColorDepth = ColorDepth.<giá_trị_Bit>;
```

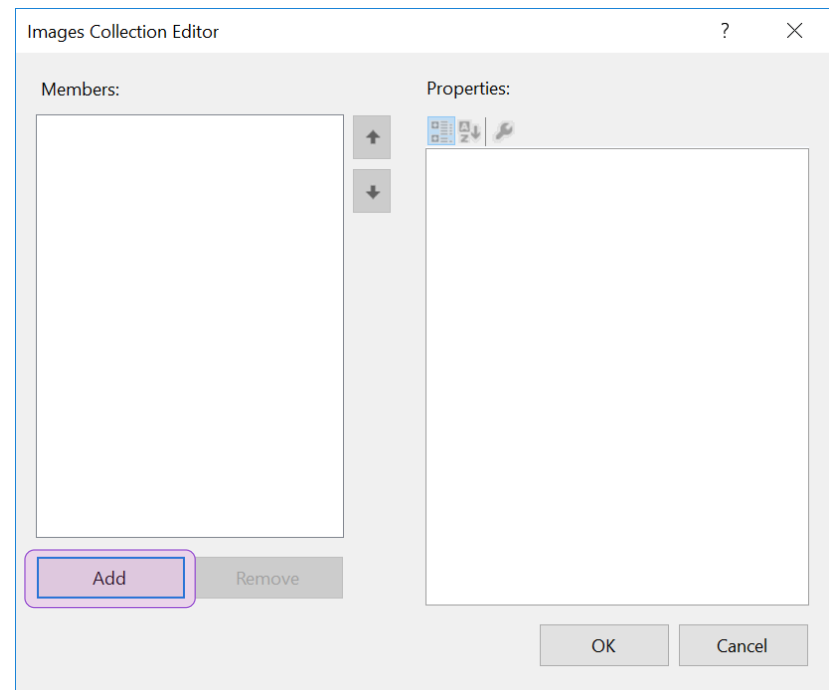
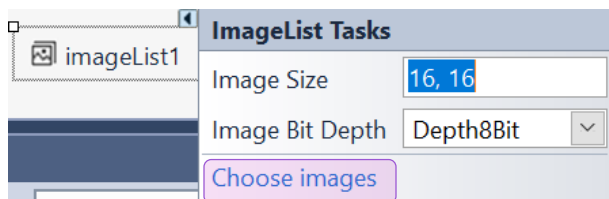
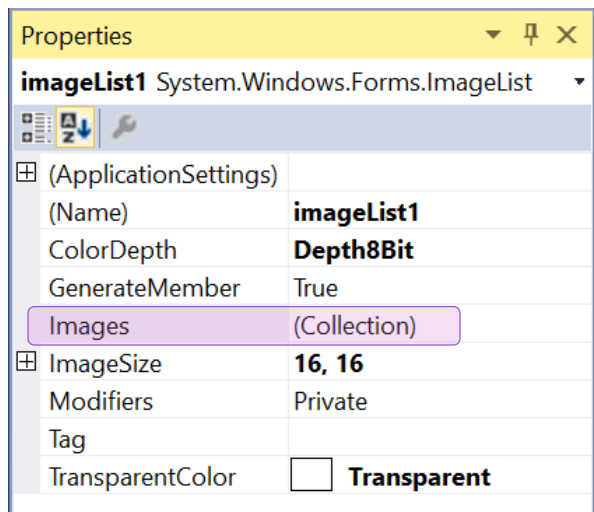
```
<tên_biến>.ImageSize = new Size(<width>, <height>);
```

```
<tên_biến>.TransparentColor = Color.<màu_sắc>;
```

ImageList

Thêm hình ảnh vào ImageList

Design:



ImageList

Thêm hình ảnh vào ImageList

Code:

```
<tên_biến>.Images.Add(Image.FromFile(<đường_dẫn>));
```

ImageList

- Lấy số lượng hình ảnh
 - `<tên_biến>.Count`
- Lấy hình ảnh
 - `<tên_biến>.Images[<index>]`
- Xóa hình ảnh
 - `<tên_biến>.RemoveAt(<index>)`
 - `<tên_biến>.Clear()`

TreeView

- TreeView là điều khiển cho phép trình bày danh sách phần tử ở dạng cây
- Mỗi phần tử của cây được gọi là một node (là đối tượng thuộc lớp **TreeNode**)

TreeView

- **Một số thuộc tính:**

- `bool` CheckBoxes
- `bool` FullRowSelect
- `ImageList` ImageList
- `int` ImageIndex
- `bool` LabelEdit
- `TreeNodeCollection` Nodes // danh sách các Node
- `bool` Scrollable
- `TreeNode` SelectedNode // Node được chọn
- `bool` ShowLines
- `bool` ShowPlusMinus
- `bool` ShowRootLines

TreeView

- **Một số phương thức:**

- void CollapseAll()
- void ExpandAll()
- int GetNodeCount(bool includeSubTrees)
 - includeSubTrees = true: bao gồm các node con

- **Sự kiện:**

- AfterSelect: xảy ra khi 1 node được chọn

TreeView

- **Lớp TreeNode:** lớp đối tượng thể hiện cho một node trong TreeView (*mỗi node trong TreeView có kiểu dữ liệu là TreeNode*)
 - **Một số thuộc tính:**
 - bool Checked
 - TreeNodeCollection Nodes
 - TreeNode FirstNode
 - TreeNode LastNode
 - TreeNode NextNode
 - TreeNode PrevNode
 - TreeNode Parent

TreeView

- **Lớp TreeNode**

- **Một số thuộc tính:**

- `string` FullPath
 - `int` ImageIndex
 - `int` SelectedImageIndex
 - `int` Index // Chỉ số của node trong cây con
 - `int` Level // Độ sâu của node trong cây. Node gốc có Level = 0
 - `string` Text

TreeView

- Thao tác Thêm, Xóa, Chọn TreeNode
 - Tham khảo ví dụ trong file [ThamKhaoTreeView.pdf](#)

TreeView

- **Bài tập 1:** chọn vào lớp trên TreeView → hiển thị danh sách sinh viên của lớp được chọn (sử dụng *CSDL DemoDB*)

The screenshot shows a Windows application window titled "Quản Lý Sinh Viên". On the left, there is a TreeView control under the heading "Danh sách lớp". It contains a folder icon labeled "Bộ môn Tin Học" which is expanded to show three sub-items: "CĐTH 17A", "CĐTH 17B", and "CĐTH 17C". "CĐTH 17A" is currently selected. On the right, under the heading "Danh sách sinh viên", there is a table displaying a list of students. The table has six columns: MSSV, Họ Tên, Ngày Sinh, Giới Tính, and Điểm TB. The first row is highlighted in blue, corresponding to the selected class. Below the table, there is a grey rectangular area.

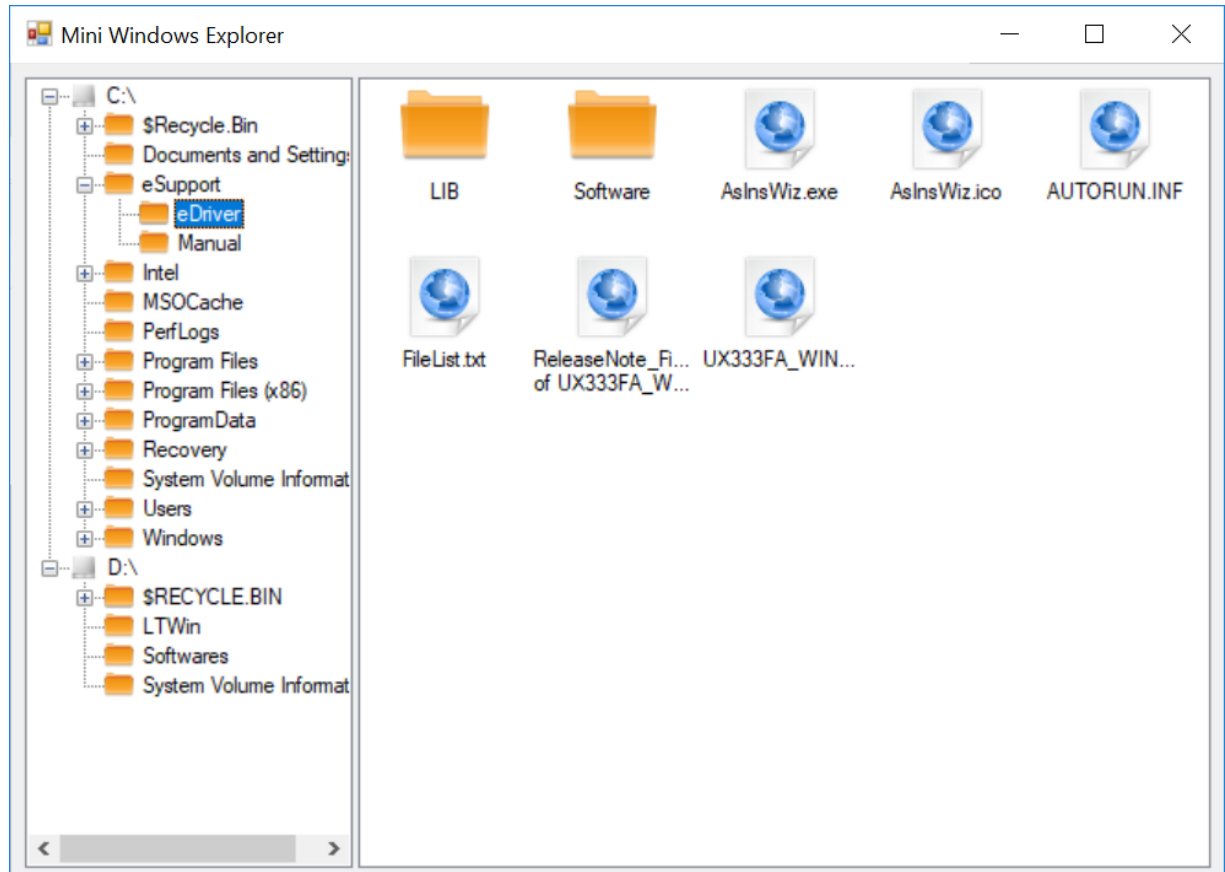
	MSSV	Họ Tên	Ngày Sinh	Giới Tính	Điểm TB
▶	0306171020	Hồ Tiến Đạt	1/4/1999	<input checked="" type="checkbox"/>	7.4
	0306171023	Phùng Hiếu Đông	8/18/1999	<input checked="" type="checkbox"/>	8.1
	0306171024	Bùi Văn Đồng	10/29/1999	<input checked="" type="checkbox"/>	6.7
	0306171036	Trần Công Khanh	12/10/1999	<input checked="" type="checkbox"/>	6.6
*				<input type="checkbox"/>	

TreeView

- **Bài tập 2:**

Load cấu trúc thư mục của máy tính (Ổ đĩa → Thư mục cấp 1 → Thư mục cấp 2)

Chọn vào thư mục trên TreeView → hiển thị danh sách thư mục con và tập tin trong thư mục được chọn



TreeView

- **Bài tập 2:** (gợi ý)

- Lấy danh sách các ổ đĩa trên máy tính

```
// Lấy tất cả ổ đĩa trên máy tính
DriveInfo[] allDrives = DriveInfo.GetDrives();

foreach (DriveInfo d in allDrives)
{
    Console.WriteLine("Tên ổ đĩa: {0}", d.Name);
    Console.WriteLine("Loại ổ đĩa: {0}", d.DriveType);
    if (d.IsReady == true) // Ổ đĩa đã sẵn sàng
    {
        Console.WriteLine("Nhãn: {0}", d.VolumeLabel);
        Console.WriteLine("File System: {0}", d.DriveFormat);
        Console.WriteLine("Dung lượng trống: {0} bytes", d.TotalFreeSpace);
        Console.WriteLine("Tổng dung lượng: {0} bytes ", d.TotalSize);
    }
}
```

TreeView

- **Bài tập 2:** (gợi ý)
 - Lấy danh sách các thư mục con của thư mục **parent**
 - `string[] dirs = Directory.GetDirectories(parent);`
 - Lấy danh sách tập tin của thư mục **parent**
 - `string[] files = Directory.GetFiles(parent);`

TreeView

- **Bài tập 2:** (gợi ý)

- Lấy thông tin của thư mục có đường dẫn **path**

- *DirectoryInfo* *dInfo* = new *DirectoryInfo*(**path**);
 - *dInfo.Name* // Tên thư mục
 - *dInfo.FullName* // Đường dẫn tuyệt đối của thư mục (**path**)
 - *dInfo.Root* // Thư mục gốc
 - *dInfo.Parent* // Thư mục cha
 - *dInfo.GetDirectories()* // Lấy danh sách thư mục con
 - *dInfo.GetFiles()* // Lấy danh sách tập tin
 - ...

TreeView

- **Bài tập 2:** (gợi ý)

- Lấy thông tin của tập tin có đường dẫn **path**

- *FileInfo flInfo = new FileInfo(path);*
- *flInfo.Name // Tên tập tin*
- *flInfo.Directory // Đường dẫn tuyệt đối của thư mục cha*
flInfo.DirectoryName // Tên thư mục cha
- *flInfo.Length // Kích thước tập tin (Bytes)*
- *flInfo.Create() // Tạo tập tin*
- *flInfo.Delete() // Xóa tập tin*
- ...