

# LẬP TRÌNH WINDOWS FORM VỚI GIAO DIỆN MDI

Biên soạn: **ThS. Nguyễn Thị Anh Thư**

1

# NỘI DUNG

1. Khái niệm MDI Form
2. MenuStrip
3. Xây dựng ứng dụng MDI Form
4. ProgressBar
5. FontDialog
6. PictureBox
7. DateTimePicker
8. DataGridView
9. ListView

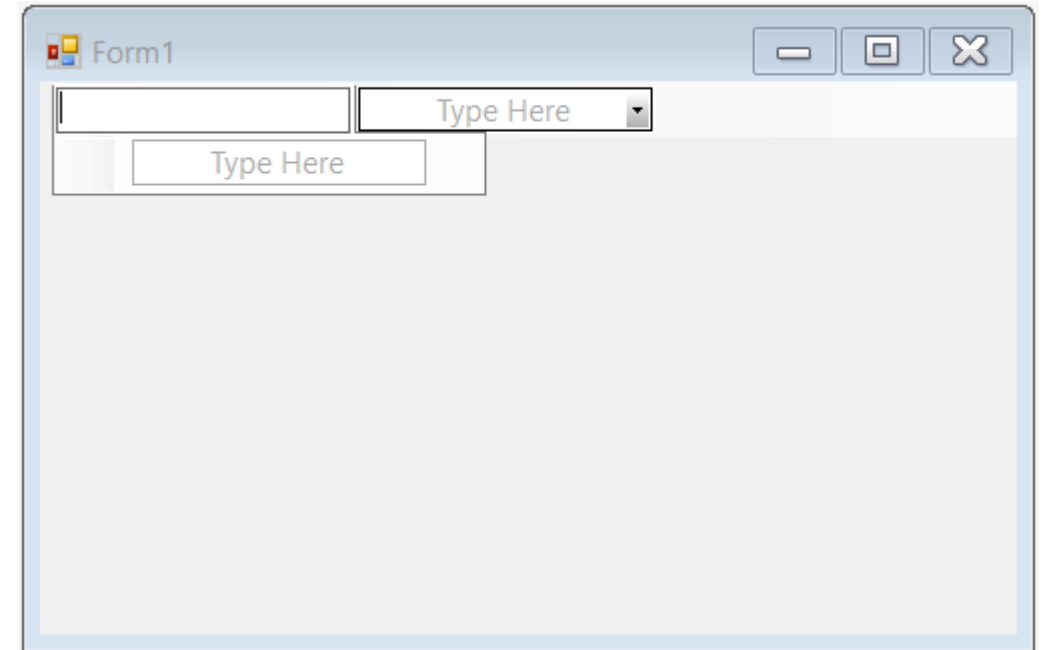
# 1. KHÁI NIỆM MDI FORM

- **MDI Form (Multiple Document Interface)** là 1 dạng Form chứa nhiều Form con.
- Một ứng dụng đa cửa sổ MDI ***có thể hiển thị nhiều cửa sổ con bên trong chúng***, điều này trái ngược với các ứng dụng có giao diện đơn lẻ (*SDI*) chỉ có thể tạo tác với một tài liệu cùng một lúc.
- Các ứng dụng đa cửa sổ MDI thường ***có một Menu chứa các Menu cấp 2 để chuyển đổi qua lại giữa các cửa sổ con***.

## 2. MENUSTRIP

Điều khiển Menu trên Windows Form rất hay được sử dụng trên các ứng dụng được thiết kế bằng C#.

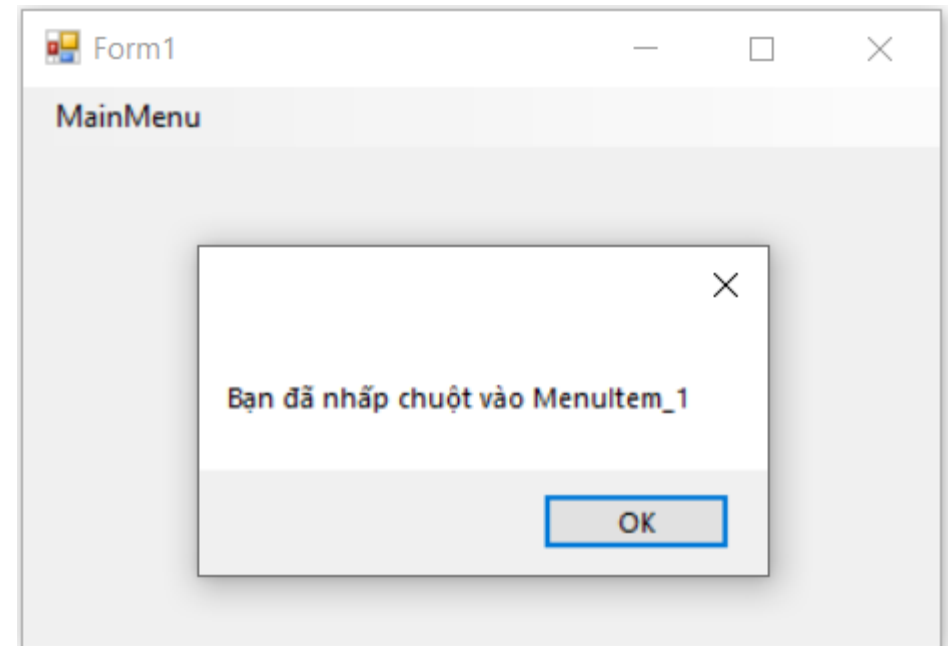
- Toolbox  $\Rightarrow$  kéo Menustrip vào Form  $\Rightarrow$  *tạo danh mục các Menu bằng cách nhấp đúp chuột vào Menu và viết tên của các MenuItem.*
- **Insert  $\Rightarrow$  Separator:** Tạo ra các vạch ngăn cách giữa các Items.
- **Nhấp đúp chuột vào mỗi Menu Item** và viết mã cho sự kiện **Click()** của Item đó.
- Tìm hiểu về **Properties** và **Events:** [Link](#).



## 2. MENUSTRIP

### Ví dụ:

- Tạo Form giao diện như hình bên.
- Sau khi tạo Menu trên Form: MainMenu  $\Rightarrow$  MenuItem\_1, MenuItem\_2.
- Sau khi nhấp chuột vào Item sẽ hiển thị hộp thoại Message.



# 3. XÂY DỰNG ỨNG DỤNG MDI FORM

Bất kỳ một Form nào của C# đều có thể trở thành MDI Form bằng cách gán thuộc tính: **IsMdiContainer = true;**

Các bước xây dựng phần mềm đa cửa sổ MDI:

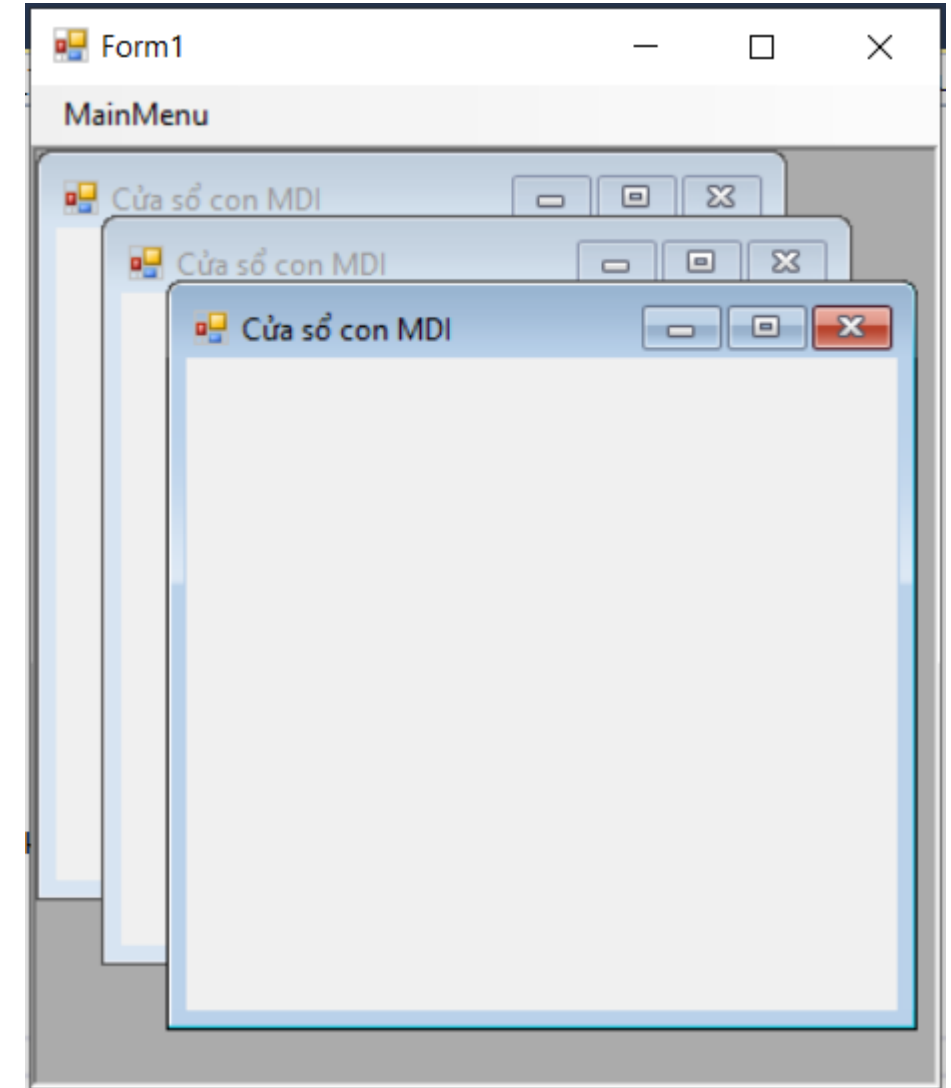
1. Tạo một Project Winform mới và đặt tên là HDMDIForm.
2. Thiết lập thuộc tính IsMdiContainer là true vào trong sự kiện Load() của Form chính: **this.IsMdiContainer = true;**
3. Kéo một menuStrip vào Form1 và tạo Menu: MainMenu  $\Rightarrow$  New Form.
4. Trong sự kiện Click của Menu:
  - **Form new\_mdi\_child = new Form();**
  - **new\_mdi\_child.Text = "Cửa sổ con MDI";**
  - **new\_mdi\_child.MdiParent = this;**
  - **new\_mdi\_child.Show();**

# 3. XÂY DỰNG ỨNG DỤNG MDI FORM

3 references

```
public partial class Form1 : Form
{
    1 reference
    public Form1()
    {
        InitializeComponent();
        this.IsMdiContainer = true;
    }

    1 reference
    private void subMenuToolStripMenuItem_Click(object sender, EventArgs e)
    {
        Form new_mdi_child = new Form();
        new_mdi_child.Text = "Cửa sổ con MDI";
        new_mdi_child.MdiParent = this;
        new_mdi_child.Show();
    }
}
```



## 4. PROGRESSBAR

**ProgressBar** dùng để hiển thị quá trình theo phần trăm (*để cho biết chương trình xử lý đã được bao nhiêu phần trăm, sắp xong hay chưa*).

- Tìm hiểu về **Properties** và **Events**: [Link](#).
- **Properties:**
  - Minimum: Đặt giá trị nhỏ nhất cho biến Value của 1 tác vụ trên ProgressBar.
  - Maximum: Đặt giá trị lớn nhất cho biến Value của 1 tác vụ trên ProgressBar.
  - Value: Thuộc tính này là giá trị hiện tại của 1 tác vụ trên ProgressBar.
  - Mặc định, các thuộc tính **Minimum** và **Maximum** được đặt là **0** và **100**. Khi chạy ứng dụng và nhấp vào nút “Bắt đầu” tiến trình sẽ được thực hiện từ trái qua phải.



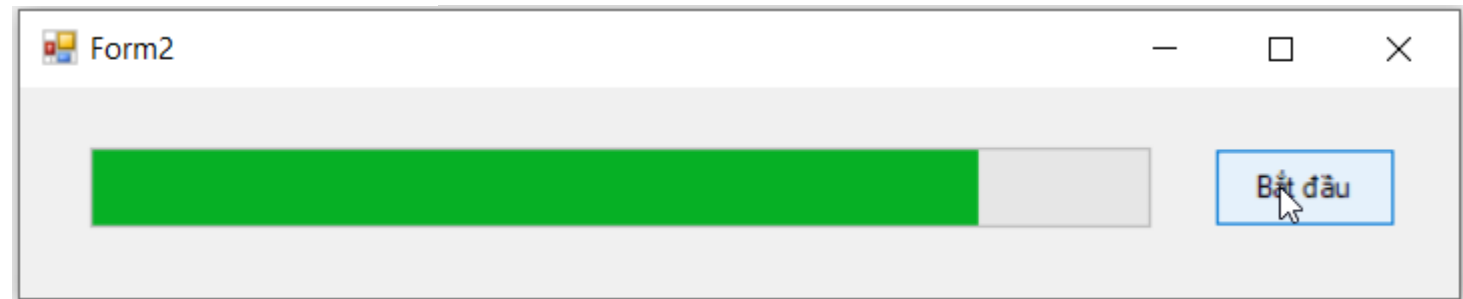
# 4. PROGRESSBAR

## Ví dụ:

- Tạo Form giao diện như hình sau.
- Trong sự kiện Click() của button, code đoạn mã sau đây:

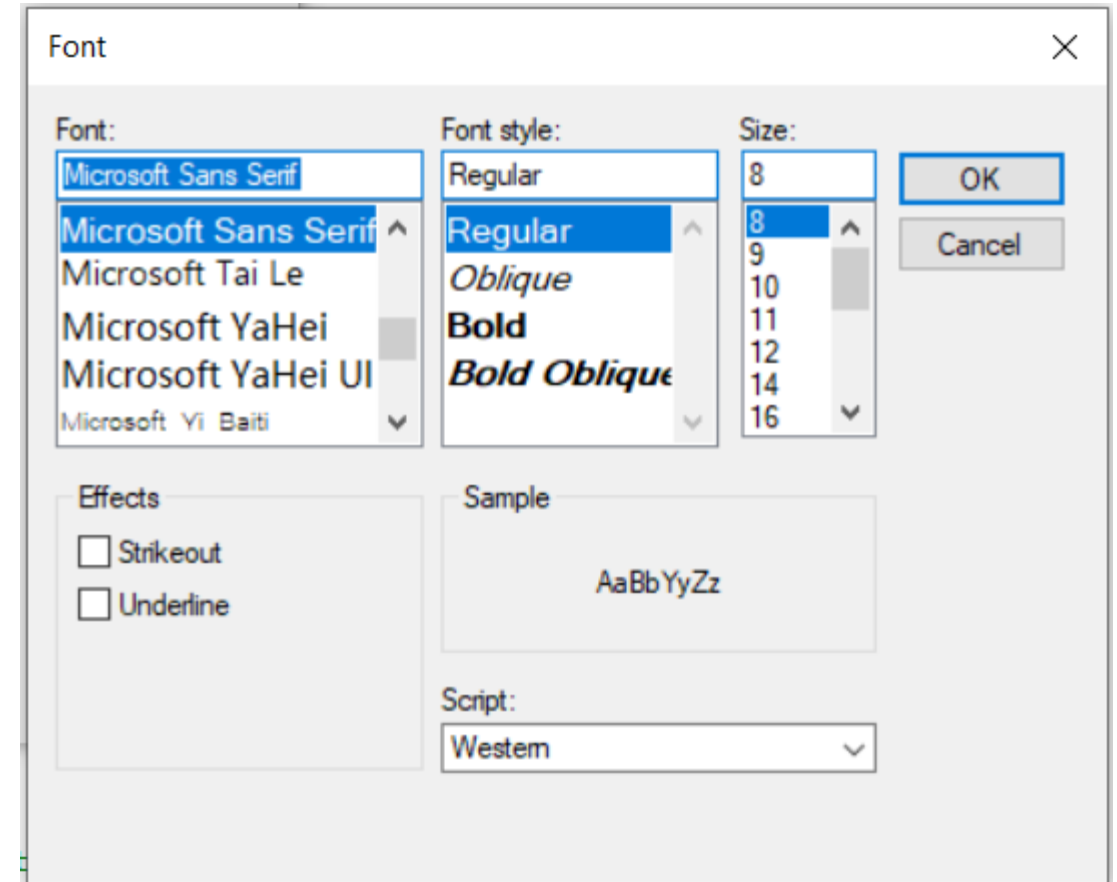
```
private void button1_Click(object sender, EventArgs e)
{
    int i; //Khởi tạo biến i

    progressBar1.Minimum = 0; //Đặt giá trị nhỏ nhất cho ProgressBar
    progressBar1.Maximum = 200; //Đặt giá trị lớn nhất cho ProgressBar
    for (i = 0; i <= 200; i++)
    {
        progressBar1.Value = i; //Gán giá trị cho ProgressBar
    }
}
```



# 5. FONTDIALOG

- FontDialogBox thường được sử dụng trên C# để xây dựng các phần mềm soạn thảo văn bản, khi thiết kế một trình duyệt web và nhiều ứng dụng khác.
- Hộp thoại FontDialogBox chứa danh sách các phông chữ được cài đặt trên hệ thống.
- Hộp thoại FontDialogBox cho phép người dùng chọn thuộc tính cho phông chữ**, ví dụ thay đổi tên phông, kiểu phông, kích thước phông, hiệu ứng và kích bản.
- Tìm hiểu về **Properties** và **Events**: [Link](#).



# 6. PICTUREBOX

Đối tượng chứa hình ảnh. Dùng để hiển thị các hình ảnh dưới định dạng Bitmap, GIF, icon, hoặc định dạng JPEG.

Tìm hiểu về **Properties** và **Events**: [Link](#).

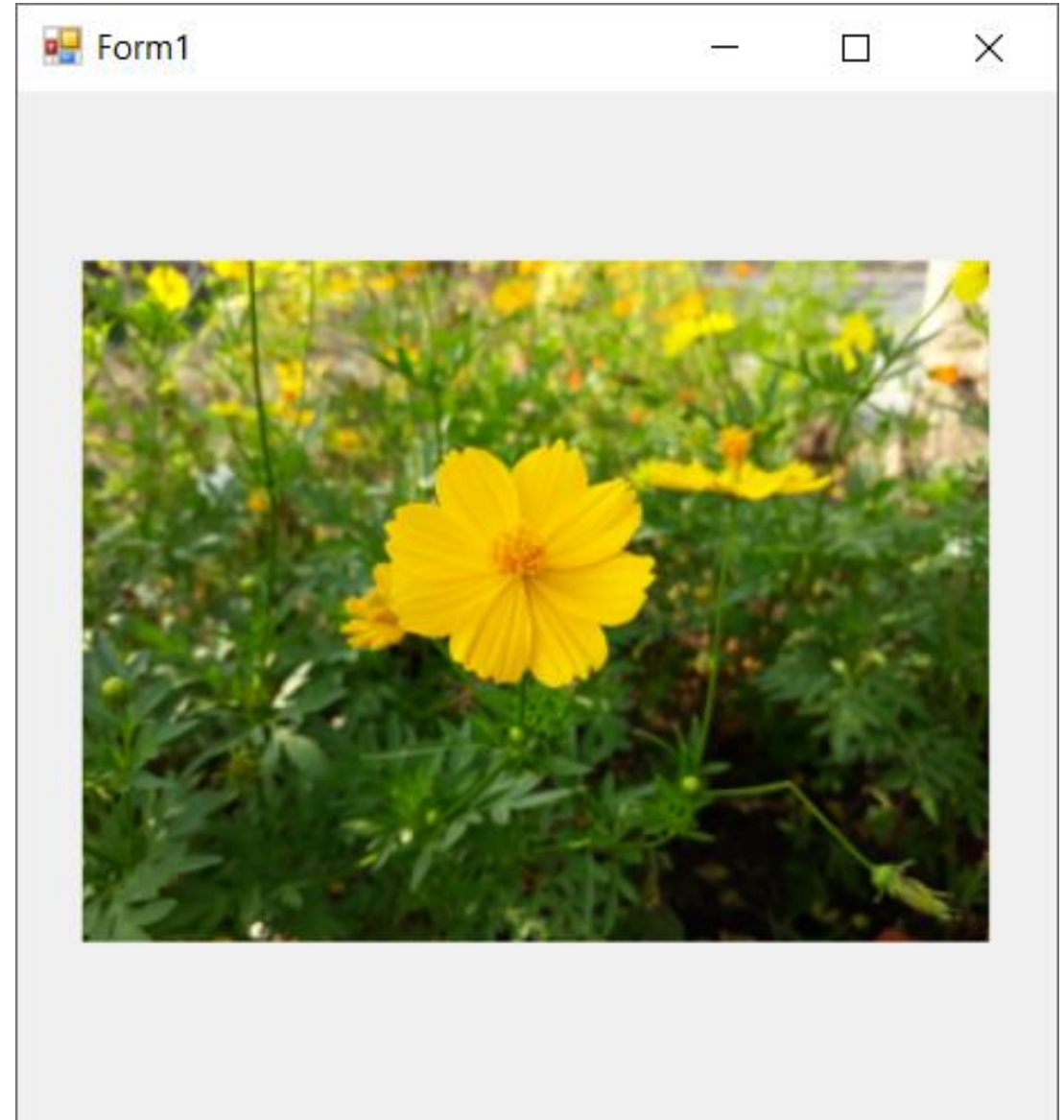
## ■ **Properties**:

- **Image**: Tải ảnh lên PictureBox.
- **SizeMode**: Dùng để thay đổi cách hiển thị ảnh trong PictureBox. Có tất cả 5 cách hiển thị hình ảnh trên PictureBox:
  - Normal: Đặt góc trái phía trên của ảnh vào vị trí góc trái phía trên của PictureBox.
  - StretchImage: Giãn nở kích thước theo PictureBox.
  - AutoSize: Tự động điều chỉnh kích cỡ ảnh.
  - CenterImage: Căn giữa ảnh.
  - Zoom: Kích thước của hình ảnh được tăng hoặc giảm mà vẫn duy trì tỷ lệ kích thước.

## 6. PICTUREBOX

### Ví dụ:

- Tạo một ứng dụng Windows Form có giao diện như hình bên.
- Hiệu chỉnh các properties đã học và quan sát sự thay đổi của PictureBox.
  - Trong cửa sổ Properties.
  - Code trong sự kiện form load.



# 7. DATETIMEPICKER

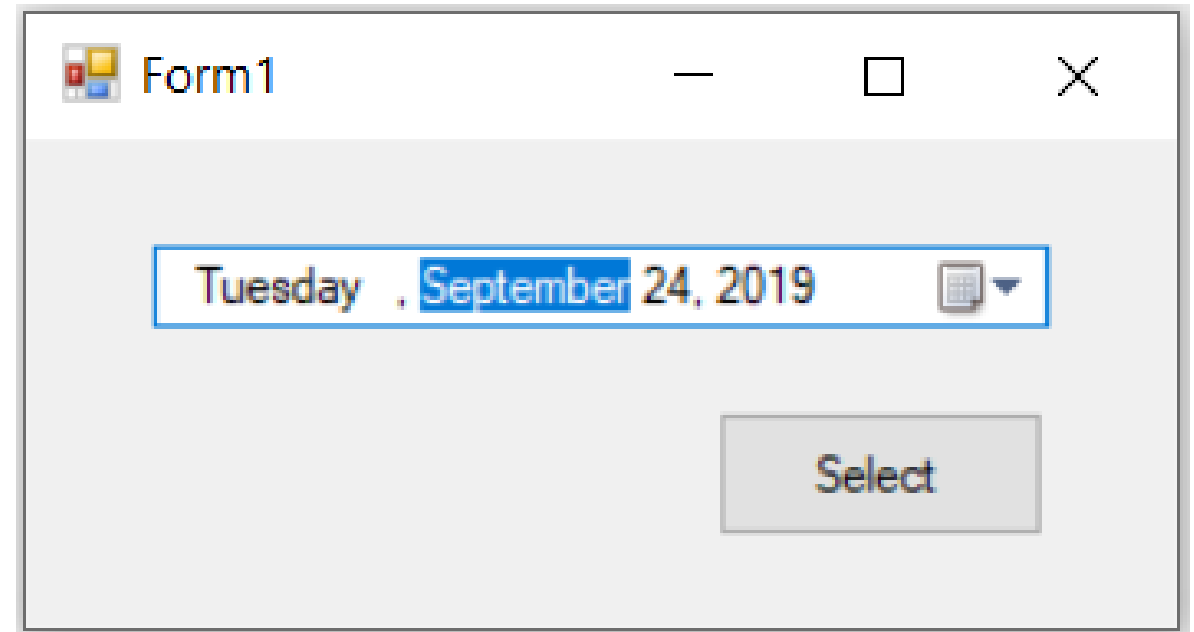
Control cho phép hiển thị và lựa chọn ngày và thời gian theo định dạng. DateTimePicker có 2 phần:

- **Phần 1**: Label để hiển thị ngày tháng đã lựa chọn.
- **Phần 2**: Bảng lịch dưới dạng Popup cho phép người dùng lựa chọn ngày tháng.
- Tìm hiểu về **Properties** và **Events**: [Link](#).
- **Properties**:
  - **Value**: Cho phép lưu trữ thông tin về ngày tháng đã được người sử dụng lựa chọn. Để lưu lại giá trị của thuộc tính Value của điều khiển DateTimePicker, cần sử dụng biến đối tượng ***DateTime***.
  - **Format**: Cho phép chọn kiểu hiển thị ngày tháng. Cụ thể, gồm có 4 định dạng hiển thị gồm các kiểu: ***Long, Short, Time, hoặc dạng Custom***.

# 7. DATETIMEPICKER

## Ví dụ:

- Tạo một ứng dụng Windows Form có giao diện như hình bên.
- Hiệu chỉnh các properties đã học và quan sát sự thay đổi của DateTimePicker.
- **Code chương trình:**
  - Click “Select” thì xuất hiện màn hình hiển thị giá trị ngày tháng mà người dùng đã chọn.



# 8. DATAGRIDVIEW

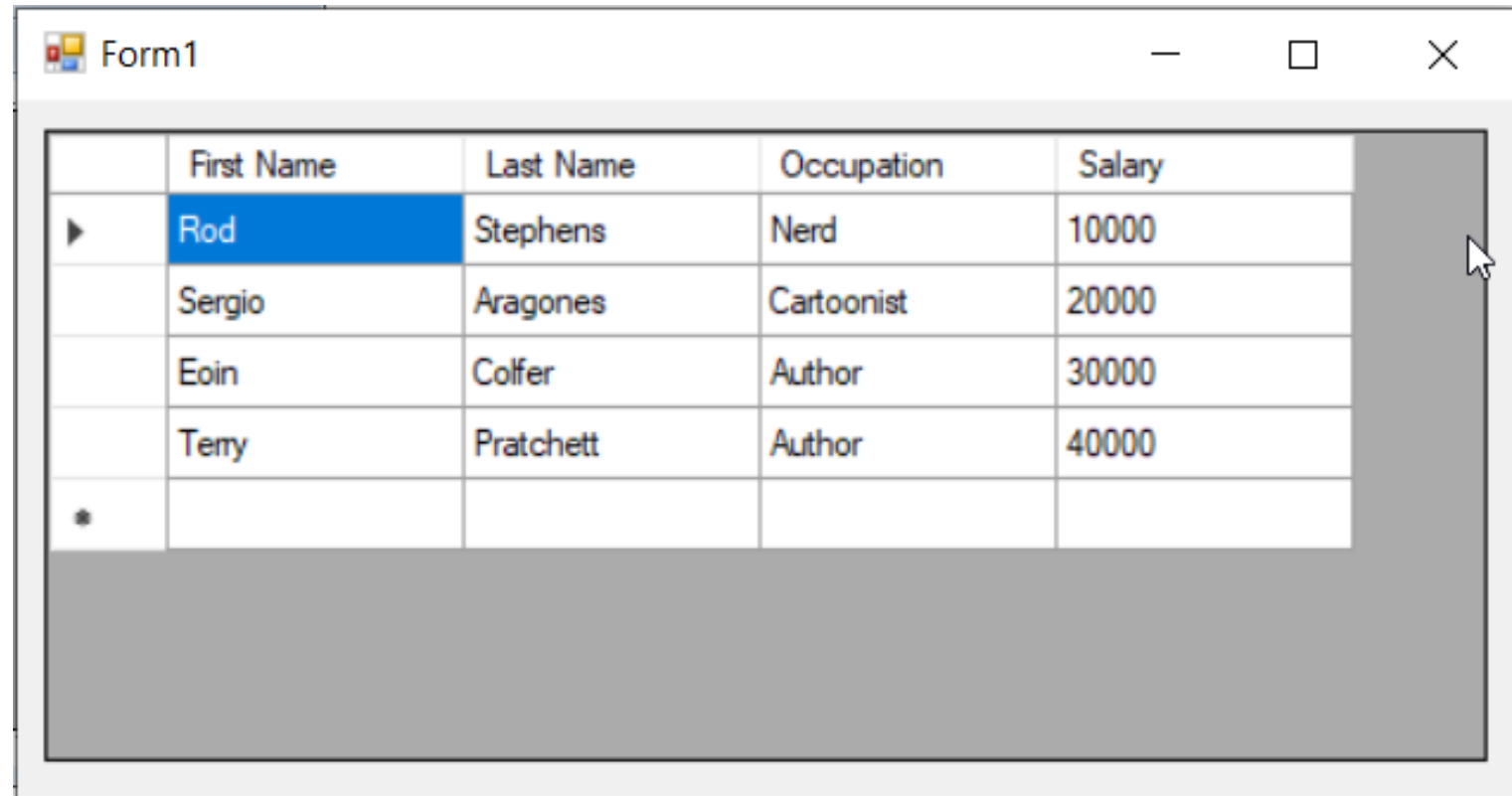
Control hiển thị dữ liệu trong một lưới tùy chỉnh.

Tìm hiểu về **Properties** và **Events**: [Link](#).

- **Properties**:

- **DataSource**: Nguồn dữ liệu.

Ví dụ: Sử dụng DataGridView để hiển thị nội dung chứa trong DataTable.



	First Name	Last Name	Occupation	Salary
▶	Rod	Stephens	Nerd	10000
	Sergio	Aragones	Cartoonist	20000
	Eoin	Colfer	Author	30000
	Terry	Pratchett	Author	40000
•				

# 8. DATAGRIDVIEW

Ví dụ: Sử dụng DataGridView để hiển thị nội dung chứa trong DataTable.

- Tạo đối tượng DataTable.
- Thêm cột vào DataTable.
- Thêm các mục vào bảng.
- Tạo DataGridView sử dụng DataTable làm nguồn dữ liệu của nó.

```
private void Form1_Load(object sender, EventArgs e)
{
    // Make the DataTable object.
    DataTable dt = new DataTable("People");

    // Add columns to the DataTable.
    dt.Columns.Add("First Name",
        System.Type.GetType("System.String"));
    dt.Columns.Add("Last Name",
        System.Type.GetType("System.String"));
    dt.Columns.Add("Occupation",
        System.Type.GetType("System.String"));
    dt.Columns.Add("Salary",
        System.Type.GetType("System.Int32"));

    // Add items to the table.
    dt.Rows.Add(new object[]
        { "Rod", "Stephens", "Nerd", 10000 });
    dt.Rows.Add(new object[]
        { "Sergio", "Aragones", "Cartoonist", 20000 });
    dt.Rows.Add(new object[]
        { "Eoin", "Colfer", "Author", 30000 });
    dt.Rows.Add(new object[]
        { "Terry", "Pratchett", "Author", 40000 });

    // Make the DataGridView use the DataTable as its data source.
    dgvPeople.DataSource = dt;
}
```



# 9. LISTVIEW

Sử dụng ListView để hiển thị thông tin dạng bảng.

Tìm hiểu về **Properties** và **Events**: [Link](#).

## ▪ **Properties:**

- **CheckBoxes**: Giá trị “*true*” để thêm Checkbox vào ListView.
- **GridLines**: Kẻ đường lưới trong ListView.
- **FullRowSelect**: Cho phép chọn 1 dòng dữ liệu trong ListView.

## ▪ **Thêm cột vào ListView:**

- `myListView.Columns.Add(Tiêu đề, Chiều rộng);`

## ▪ **Thêm Item vào Listview:**

- Sử dụng **ListviewItem** đại diện cho một Item trong Listview Control.

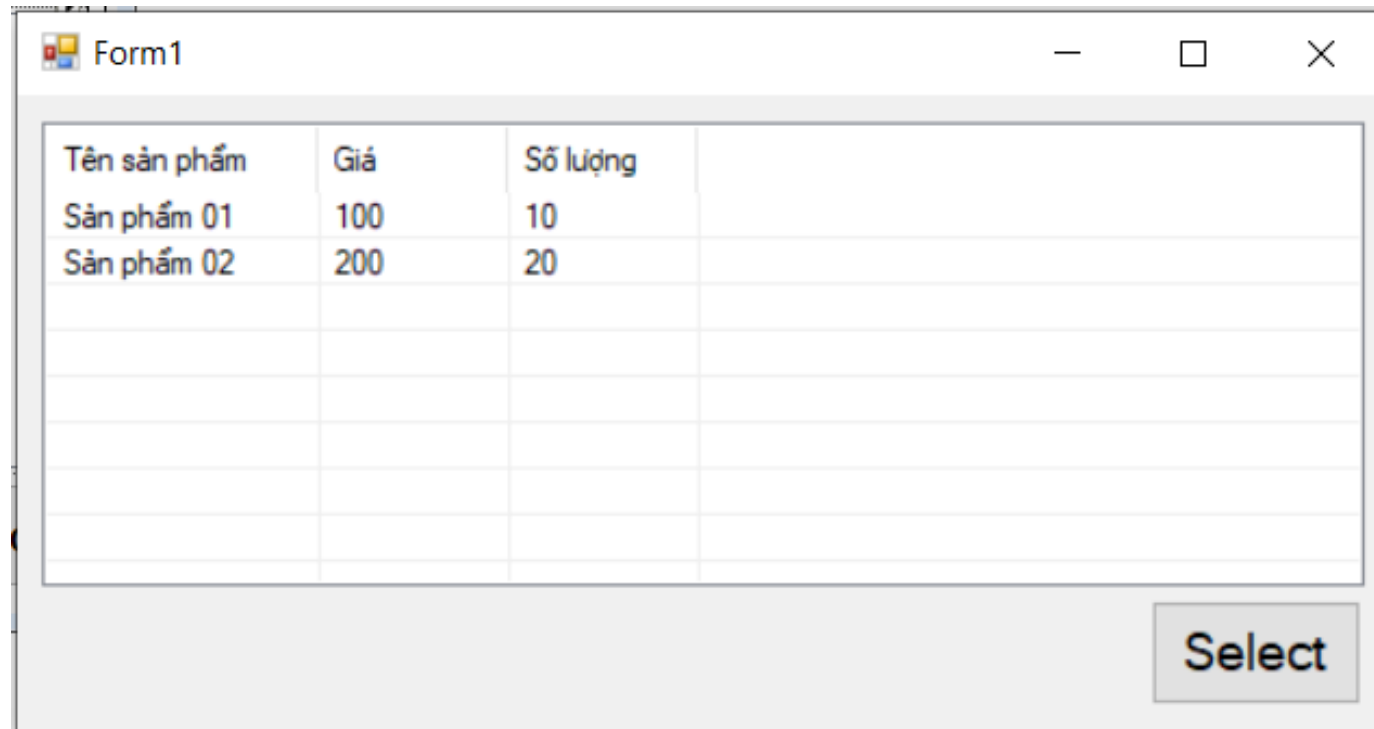
## ▪ **Lấy giá trị trả lại của Listview khi Item được chọn:**

- `productName = listView1.SelectedItems[0].SubItems[0].Text;`

## 9. LISTVIEW

### Ví dụ:

- Tạo một ứng dụng Windows Form có giao diện như hình bên.
- Thêm dữ liệu vào ListView.
- Click “Select” thì xuất hiện màn hình hiển thị dòng được chọn.



The screenshot shows a Windows Form titled "Form1" with a standard Windows title bar (minimize, maximize, close buttons). Inside the form, there is a ListView control displaying a table with three columns: "Tên sản phẩm", "Giá", and "Số lượng". The table contains two rows of data: "Sản phẩm 01" with a price of 100 and a quantity of 10, and "Sản phẩm 02" with a price of 200 and a quantity of 20. Below the table, there is a "Select" button.

Tên sản phẩm	Giá	Số lượng
Sản phẩm 01	100	10
Sản phẩm 02	200	20

Select

# 9. LISTVIEW

## Ví dụ:

```
private void button1_Click(object sender, EventArgs e)
{
    string productName = null;
    string price = null;
    string quantity = null;

    if (listView1.SelectedItems.Count > 0)
    {
        productName = listView1.SelectedItems[0].SubItems[0].Text;
        price = listView1.SelectedItems[0].SubItems[1].Text;
        quantity = listView1.SelectedItems[0].SubItems[2].Text;

        MessageBox.Show(productName + " , " + price + " , " + quantity);
    }
    else
    {
        MessageBox.Show("Hãy chọn 1 sản phẩm");
    }
}
```

```
private void Form1_Load(object sender, EventArgs e)
{
    listView1.View = View.Details;
    listView1.GridLines = true;
    listView1.FullRowSelect = true;

    //Thêm tiêu đề cho cột
    listView1.Columns.Add("Tên sản phẩm", 100);
    listView1.Columns.Add("Giá", 70);
    listView1.Columns.Add("Số lượng", 70);

    //Thêm Item vào listview
    string[] arr = new string[4];
    ListViewItem itm;

    //Thêm Item đầu tiên
    arr[0] = "Sản phẩm 01";
    arr[1] = "100";
    arr[2] = "10";
    itm = new ListViewItem(arr);
    listView1.Items.Add(itm);

    //Thêm Item thứ 2
    arr[0] = "Sản phẩm 02";
    arr[1] = "200";
    arr[2] = "20";
    itm = new ListViewItem(arr);
    listView1.Items.Add(itm);
}
```

# BÀI TẬP

- Sinh viên làm **3 bài tập** sau theo file hướng dẫn **Lab05.pdf** và tư duy hiệu chỉnh lại chức năng phù hợp:
  - **Bài tập 1 & 2:** Xây dựng chương trình Order của một quán thức ăn nhanh.
  - **Bài tập 3:** Xây dựng 1 form quản lý nhân viên.



# Q & A

18/07/2021