

LẬP TRÌNH WINDOWS

NỘI DUNG

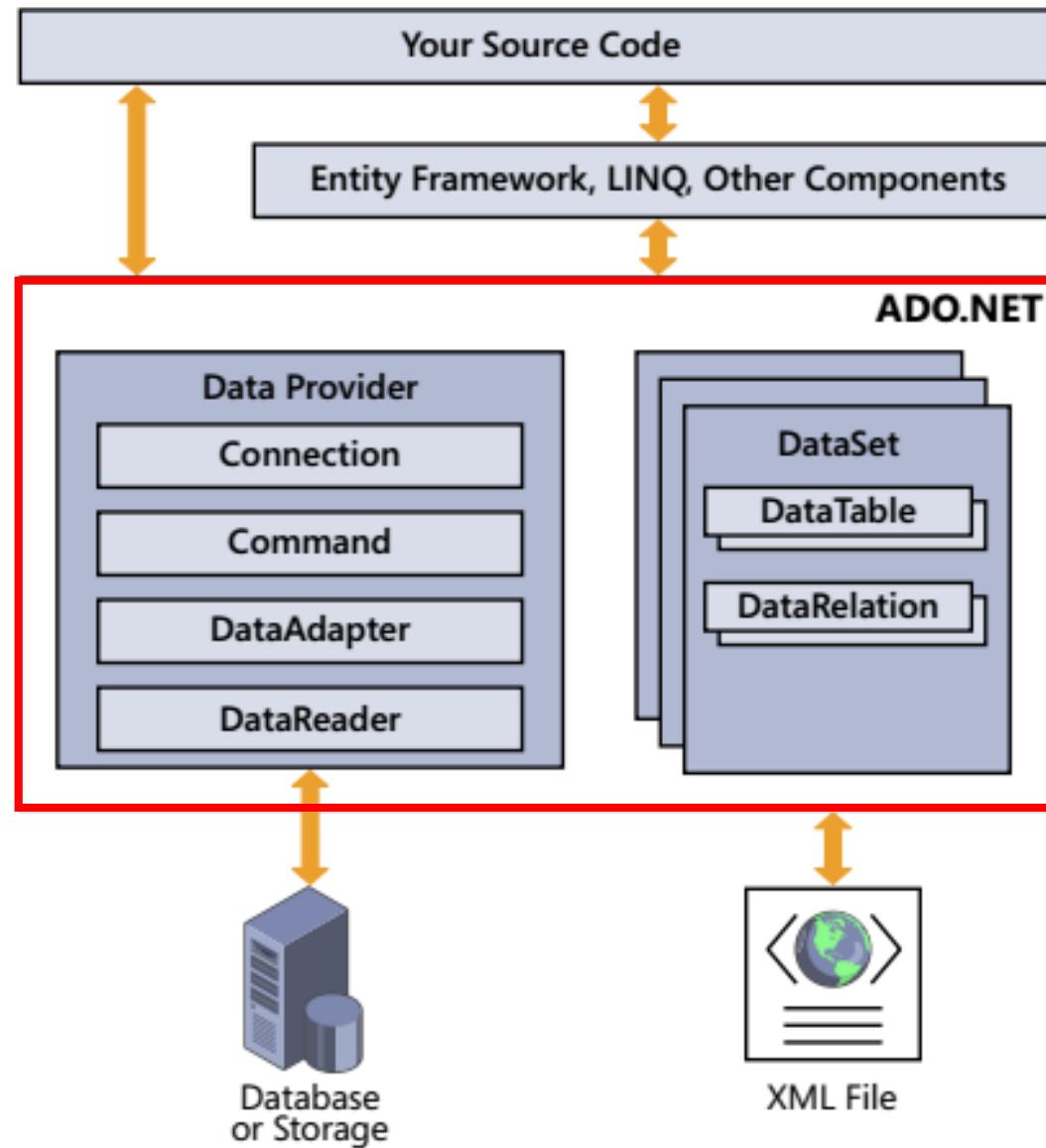
- Giới thiệu ADO.Net
- Các bước lập trình thao tác với CSDL
- .Net Framework Data Provider
- DataSet

GIỚI THIỆU ADO.Net

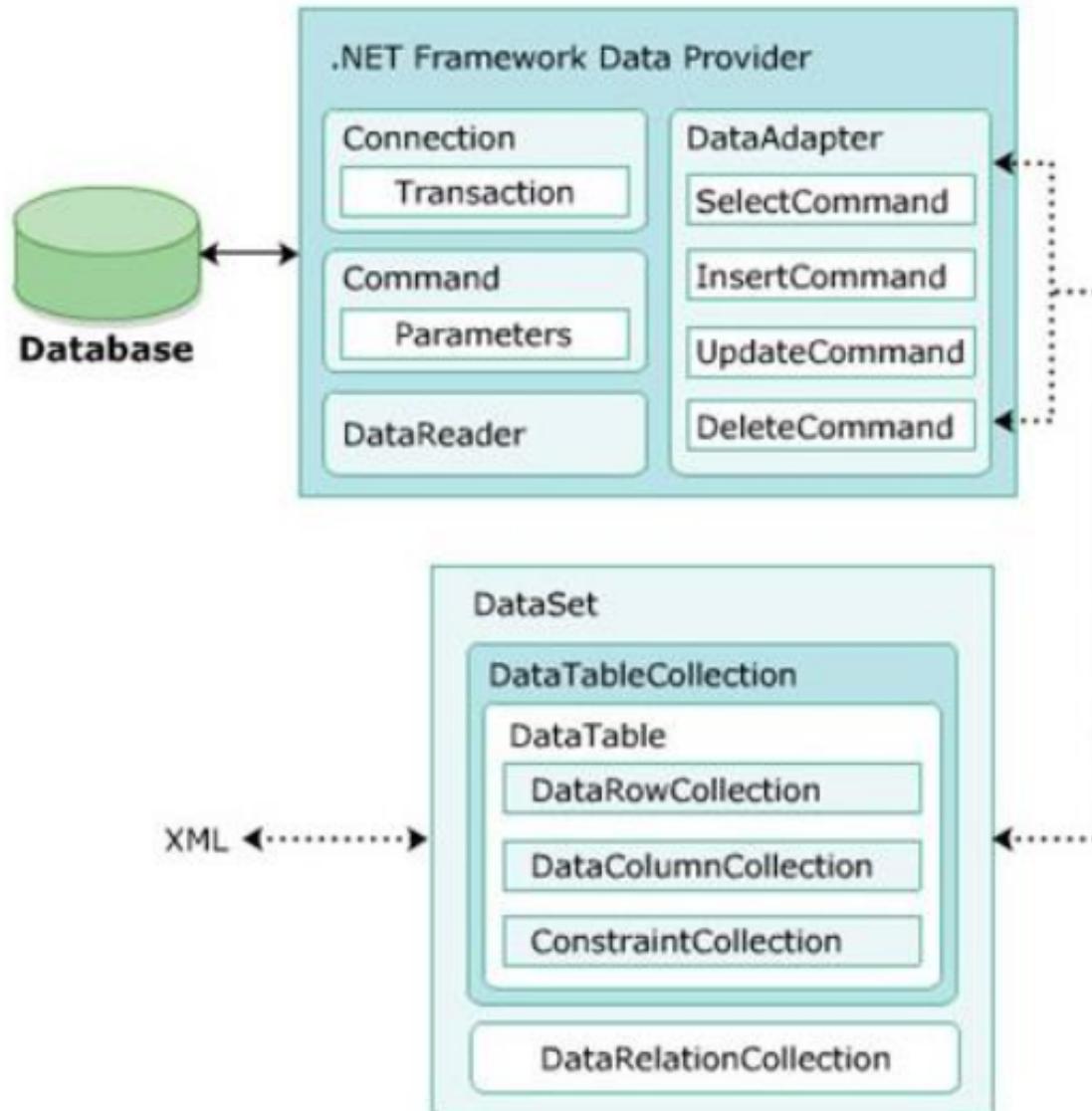
ADO.Net

- là một thành phần trong ***dotNet Framework***
- tập hợp các lớp đối tượng cho phép thao tác với nguồn dữ liệu:
 - *cơ sở dữ liệu lưu trữ trong các hệ quản trị cơ sở dữ liệu như MS Access, MS SQL Server, Oracle...*
 - *tập tin XML, Excel...*
- hỗ trợ mô hình ngắt kết nối (*disconnected model*)
- sử dụng XML để tương tác với cơ sở dữ liệu, nghĩa là dữ liệu trong CSDL được chuyển sang định dạng XML để thực hiện các thao tác truy vấn, cập nhật cơ sở dữ liệu.

GIỚI THIỆU ADO.Net



GIỚI THIỆU ADO.Net

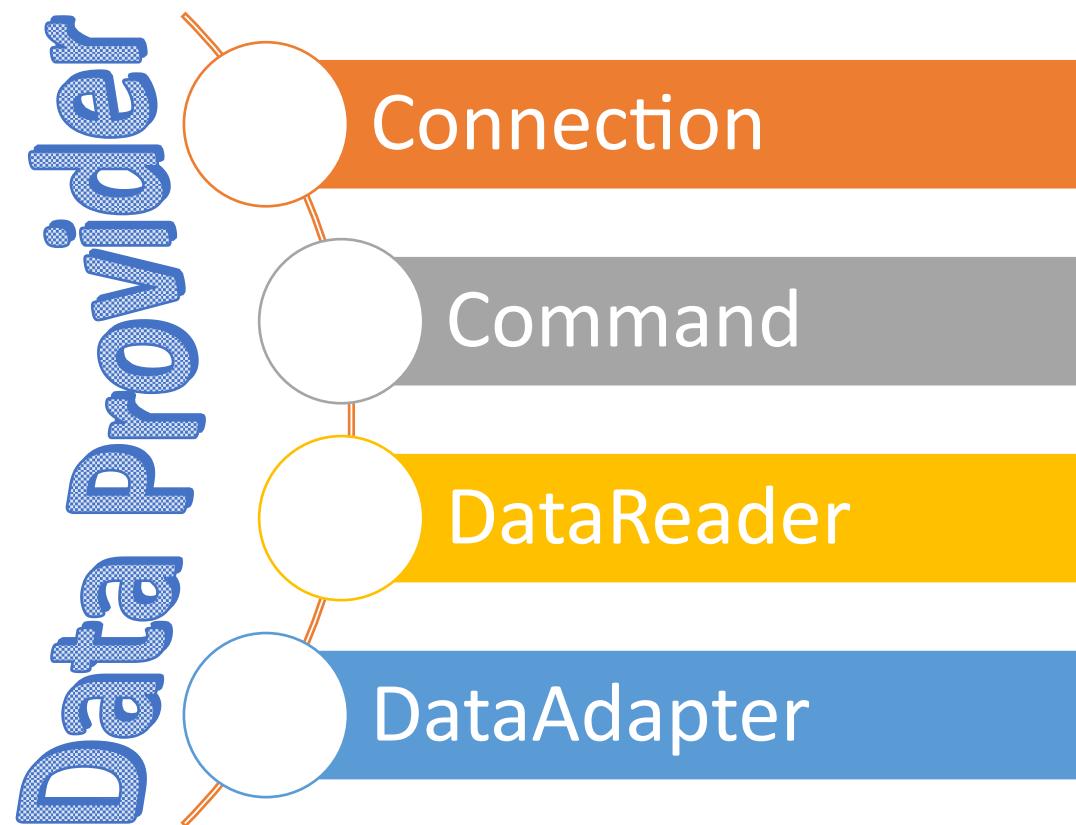


CÁC BƯỚC LẬP TRÌNH THAO TÁC VỚI CSDL

-
- 1 • Tạo và mở kết nối với CSDL
 - 2 • Tạo đối tượng truy vấn, thao tác với CSDL
 - 3 • Truy vấn, thao tác CSDL và xử lý kết quả trả về
(nếu có)
 - 4 • Đóng kết nối CSDL

.Net Framework DATA PROVIDER

- Thực hiện việc kết nối và duy trì kết nối đến nguồn dữ liệu.
- Các đối tượng trong .Net Framework **Data Provider**



.Net Framework DATA PROVIDER

- Các loại .Net Framework Data Provider

Loại CSDL	Tên Provider	Namespace
ODBC	.Net Framework Data Provider for ODBC	System.Data.Odbc
Access Excel MS SQL Server...	.Net Framework Data Provider for OleDb	System.Data.OleDb
MS SQL Server	.Net Framework Data Provider for SQL Server	System.Data.SqlClient
Oracle	.Net Framework Data Provider for Oracle	System.Data.OracleClient

CƠ SỞ DỮ LIỆU DemoDB

- Table **SINH_VIEN**

	Column Name	Data Type
PK	MSSV	nvarchar(10)
	HoTen	nvarchar(50)
	NgaySinh	date
	GioiTinh	bit
	DiemTB	float

```
CREATE TABLE SINH_VIEN(
    MSSV nvarchar(10) PRIMARY KEY,
    HoTen nvarchar(50),
    NgaySinh date,
    GioiTinh bit,
    DiemTB float
)
```

.Net Framework DATA PROVIDER

- **Đối tượng Connection**

- Tạo một kết nối giữa ứng dụng với cơ sở dữ liệu

Tên Provider	Class Connection	
.Net Framework Data Provider for ODBC	OdbcConnection	
.Net Framework Data Provider for OleDb	OleDbConnection	
.Net Framework Data Provider for SQL Server	SqlConnection	XYZConnection
.Net Framework Data Provider for Oracle	OracleConnection	

.Net Framework DATA PROVIDER

• Đối tượng Connection

- Khai báo:

- `XYZConnection <tên_bien> = new XYZConnection();`
- `XYZConnection <tên_bien> = new XYZConnection(<chuỗi_kết_nối>);`

- Chuỗi kết nối:

- Mỗi HQT CSDL khác nhau sẽ có chuỗi kết nối khác nhau

- Ví dụ:

- MS Access 2007:

`Provider=Microsoft.ACE.OLEDB.12.0; Data Source =
C:\myFolder\myAccessFile.accdb;`

- MS SQL Server:

`Data Source=myServerAddress;Initial Catalog=myDataBase;Integrated
Security=true; User ID=myUsername;Password=myPassword;`

- ...

- Tham khảo chuỗi kết nối tại website: <https://www.connectionstrings.com>

.Net Framework DATA PROVIDER

• Đối tượng Connection

- Một số thuộc tính:
 - `string` `ConnectionString`: chuỗi kết nối
 - `ConnectionState` `State`: trạng thái của đối tượng Connection
 - Broken
 - **Closed**
 - Connecting
 - Executing
 - Fetching
 - Open
- Một số phương thức:
 - `void` `Open()`: mở kết nối đến CSDL
 - `void` `Close()`: đóng kết nối CSDL

.Net Framework DATA PROVIDER

- **Đối tượng Connection**

- Ví dụ: tạo kết nối đến CSDL DemoDB

```
// Chuỗi kết nối
string ChuoiKetNoi = "Data Source=THANHTUAN-PC;Initial Catalog=DemoDB;Integrated Security=true;";
// Tạo đối tượng kết nối
SqlConnection conn = new SqlConnection(ChuoiKetNoi);

// Mở kết nối
conn.Open();

// Thực hiện truy vấn / cập nhật
// ...

// Đóng kết nối
conn.Close();
```

.Net Framework DATA PROVIDER

- **Đối tượng Connection**

- Ví dụ: tạo kết nối đến CSDL DemoDB

```
// Tạo đối tượng kết nối
SqlConnection conn = new SqlConnection();
conn.ConnectionString = "Data Source=THANHTUAN-PC;Initial Catalog=DemoDB;Integrated Security=true;";

// Mở kết nối
conn.Open();

// Thực hiện truy vấn / cập nhật
// ...

// Đóng kết nối
conn.Close();
```

.Net Framework DATA PROVIDER

- **Đối tượng Command**

- Thực thi một câu lệnh SQL: SELECT, INSERT, UPDATE, DELETE
- Thực thi một Store Procedure

Tên Provider	Class Command	
.Net Framework Data Provider for ODBC	OdbcCommand	
.Net Framework Data Provider for OleDb	OleDbCommand	
.Net Framework Data Provider for SQL Server	SqlCommand	
.Net Framework Data Provider for Oracle	OracleCommand	

XYZCommand

.Net Framework DATA PROVIDER

- **Đối tượng Command**

- Khai báo:
 - `XYZCommand <tên_biến> = new XYZCommand();`
 - `XYZCommand <tên_biến> = new XYZCommand(<câu_lệnh_SQL>, <đối_tượng_Connection>);`
- Một số thuộc tính:
 - `string CommandText`: câu lệnh SQL hoặc Store Procedure
 - `XYZConnection Connection`: đối tượng Connection
 - `CommandType CommandType`: kiểu câu lệnh thực thi
 - Text
 - StoreProcedure
 - `XYZParameterCollection Parameters`: danh sách các tham số trong câu lệnh (`CommandText`)

.Net Framework DATA PROVIDER

- **Đối tượng Command**

- Một số phương thức:
 - `int ExecuteNonQuery()`
 - Thực thi các câu lệnh SQL: INSERT, UPDATE, DELETE.
 - Trả về số dòng được thực thi
 - `object ExecuteScalar()`
 - Thực thi câu lệnh SQL SELECT.
 - Trả về một giá trị duy nhất có kiểu object
 - `XYZDataReader ExecuteReader()`
 - Thực thi câu lệnh SQL SELECT.
 - Trả về đối tượng XYZDataReader chứa kết quả truy vấn SELECT

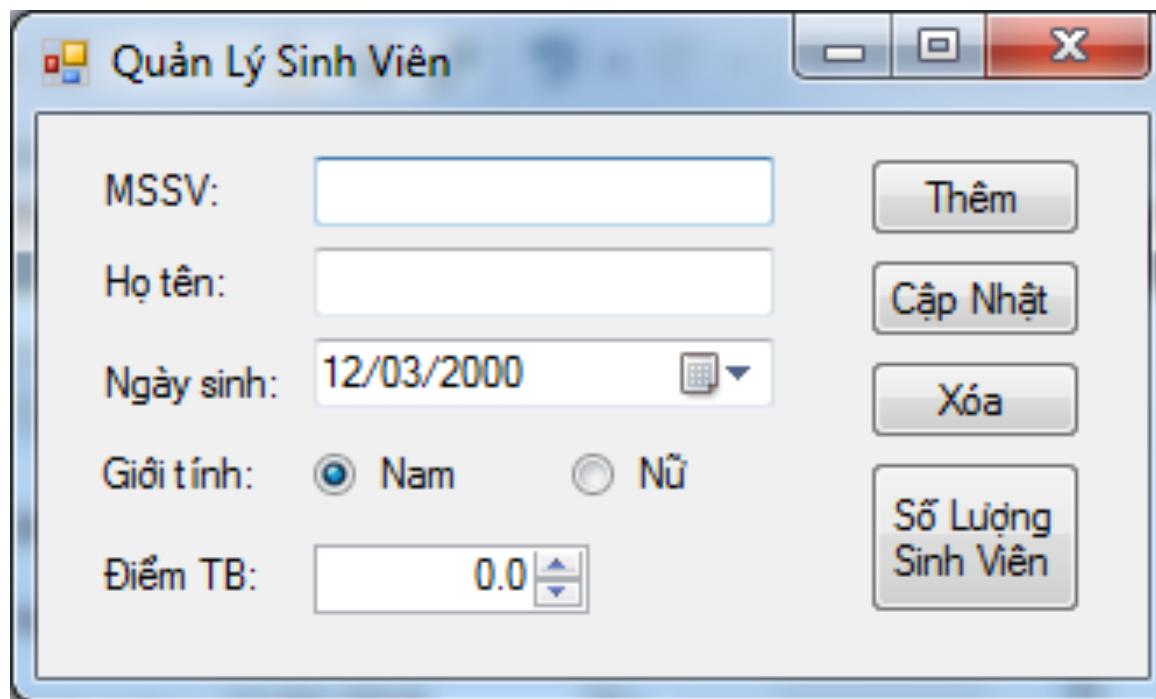
.Net Framework DATA PROVIDER

- **Đối tượng Command**

- Ôn tập lại các câu lệnh SQL
 - INSERT INTO <bảng>(<danh_sách_cột>) VALUES (<danh_sách_giá_trị>)
 - UPDATE <bảng> SET <cột_1> = <giá_trị_1>, <cột_2> = <giá_trị_2>, ... , <cột_N> = <giá_trị_N> WHERE <điều_kiện>
 - DELETE FROM <bảng> WHERE <điều_kiện>
 - SELECT <danh_sách_cột> FROM <bảng> WHERE <điều_kiện>
 - Truy vấn lồng, kết bảng
 - Các hàm thống kê: COUNT, MIN, MAX, SUM, AVERAGE...
 - Mệnh đề ORDER, HAVING
 - ...

.Net Framework DATA PROVIDER

- Ứng dụng Quản Lý Sinh Viên đơn giản



.Net Framework DATA PROVIDER

- **Đối tượng Command**

- Ví dụ: thêm sinh viên mới (bảng SINH_VIEN trong CSDL DemoDB)

```
// B1. Tạo đối tượng kết nối
SqlConnection conn = new SqlConnection("Data Source=THANHTUAN-PC;Initial Catalog=DemoDB;Integrated Security=true;");

// B2. Mở kết nối
conn.Open();

// B3. Thực hiện thêm mới
string CauTruyVan = string.Format("INSERT INTO SINH_VIEN(MSSV, HoTen, NgaySinh, GioiTinh, DiemTB) VALUES ('{0}', N'{1}', '{2}', {3}, {4})",
    txtMSSV.Text, txtHoTen.Text, dtpNgaySinh.Value.ToString("yyyy-MM-dd"), (radNam.Checked ? 1 : 0), nudDiemTB.Value);
SqlCommand com = new SqlCommand(CauTruyVan, conn);

int NumOfRows = com.ExecuteNonQuery();
if(NumOfRows > 0)
{
    // Xử lý sau khi thêm thành công
}
else
{
    // Xử lý sau khi thêm thất bại
}
// B4. Đóng kết nối
conn.Close();
```

.Net Framework DATA PROVIDER

- **Đối tượng Command**

- Ví dụ: thêm sinh viên mới (bảng SINH_VIEN trong CSDL DemoDB)

```
// B1. Tạo đối tượng kết nối
SqlConnection conn = new SqlConnection("Data Source=THANHTUAN-PC;Initial Catalog=DemoDB;Integrated Security=true");

// B2. Mở kết nối
conn.Open();

// B3. Thực hiện thêm mới
SqlCommand com = new SqlCommand();
com.CommandText = string.Format("INSERT INTO SINH_VIEN(MSSV, HoTen, NgaySinh, GioiTinh, DiemTB) VALUES ('{0}', N'{1}', '{2}', {3}, {4})", 
    txtMSSV.Text, txtHoTen.Text, dtpNgaySinh.Value.ToString("yyyy-MM-dd"), (radNam.Checked ? 1 : 0), nudDiemTB.Value);
com.Connection = conn;

int NumOfRows = com.ExecuteNonQuery();
if(NumOfRows > 0)
{
    // Xử lý sau khi thêm thành công
}
else
{
    // Xử lý sau khi thêm thất bại
}
// B4. Đóng kết nối
conn.Close();
```

.Net Framework DATA PROVIDER

- **Đối tượng Command**

- Ví dụ: thêm sinh viên mới (bảng SINH_VIEN trong CSDL DemoDB)

```
CREATE PROCEDURE ThemSinhVien
    @mssv      nvarchar(10),
    @ho_ten    nvarchar(50),
    @ngay_sinh date,
    @gioi_tinh bit,
    @diem_tb   float
AS
BEGIN
    SET NOCOUNT ON
    INSERT INTO SINH_VIEN
    (
        MSSV,
        HoTen,
        NgaySinh,
        GioiTinh,
        DiemTB
    )
    VALUES
    (
        @mssv,
        @ho_ten,
        @ngay_sinh,
        @gioi_tinh,
        @diem_tb
    )
END
```

.Net Framework DATA PROVIDER

• Đối tượng Command

- Ví dụ: thêm sinh viên mới (bảng SINH_VIEN trong CSDL DemoDB)

```
// B1. Tạo đối tượng kết nối
SqlConnection conn = new SqlConnection("Data Source=THANHTUAN-PC;Initial Catalog=DemoDB;Integrated Security=true;");

// B2. Mở kết nối
conn.Open();

// B3. Thực hiện thêm mới (sử dụng store procedure)
SqlCommand com = new SqlCommand("ThemSinhVien", conn);
com.CommandType = CommandType.StoredProcedure;
com.Parameters.AddWithValue("@mssv", txtMSSV.Text);
com.Parameters.AddWithValue("@ho_ten", txtHoTen.Text);
com.Parameters.AddWithValue("@ngay_sinh", dtpNgaySinh.Value);
com.Parameters.AddWithValue("@gioi_tinh", radNam.Checked);
com.Parameters.AddWithValue("@diem_tb", nudDiemTB.Value);

int NumOfRows = com.ExecuteNonQuery();
if(NumOfRows > 0)
{
    // Xử lý sau khi thêm thành công
}
else
{
    // Xử lý sau khi thêm thất bại
}
// B4. Đóng kết nối
conn.Close();
```

.Net Framework DATA PROVIDER

- **Đối tượng Command**

- Ví dụ: cho biết số lượng sinh viên (bảng SINH_VIEN trong CSDL DemoDB)

```
// B1. Tạo đối tượng kết nối
SqlConnection conn = new SqlConnection("Data Source=THANHTUAN-PC;Initial Catalog=DemoDB;Integrated Security=true;");

// B2. Mở kết nối
conn.Open();

// B3. Thực hiện truy vấn số lượng sinh viên
string CauTruyVan = "SELECT count(*) FROM SINH_VIEN";
SqlCommand com = new SqlCommand(CauTruyVan, conn);

int NumOfStudents = (int)com.ExecuteScalar();

// B4. Đóng kết nối
conn.Close();
```

.Net Framework DATA PROVIDER

- **Đối tượng Command**

- Ví dụ: cập nhật thông tin sinh viên (bảng SINH_VIEN trong CSDL DemoDB)

```
// B1. Tạo đối tượng kết nối
SqlConnection conn = new SqlConnection("Data Source=THANHTUAN-PC;Initial Catalog=DemoDB;Integrated Security=true");

// B2. Mở kết nối
conn.Open();

// B3. Thực hiện truy vấn số lượng sinh viên
string CauTruyVan = string.Format("UPDATE SINH_VIEN SET HoTen = N'{0}', NgaySinh = '{1}', GioiTinh = {2}, DiemTB = {3} WHERE MSSV = '{4}'",
    txtHoTen.Text, dtpNgaySinh.Value.ToString("yyyy-MM-dd"), (radNam.Checked ? 1 : 0), nudDiemTB.Value, txtMSSV.Text);
SqlCommand com = new SqlCommand(CauTruyVan, conn);

int NumOfRows = com.ExecuteNonQuery();
if (NumOfRows > 0)
{
    // Xử lý sau khi thêm thành công
}
else
{
    // Xử lý sau khi thêm thất bại
}

// B4. Đóng kết nối
conn.Close();
```

.Net Framework DATA PROVIDER

- **Đối tượng Command**

- Ví dụ: xóa sinh viên (bảng SINH_VIEN trong CSDL DemoDB)

```
// B1. Tạo đối tượng kết nối
SqlConnection conn = new SqlConnection("Data Source=THANHTUAN-PC;Initial Catalog=DemoDB;Integrated Security=true;");

// B2. Mở kết nối
conn.Open();

// B3. Thực hiện truy vấn số lượng sinh viên
string CauTruyVan = string.Format("DELETE FROM SINH_VIEN WHERE MSSV = '{0}'", txtMSSV.Text);
SqlCommand com = new SqlCommand(CauTruyVan, conn);

int NumOfRows = com.ExecuteNonQuery();
if (NumOfRows > 0)
{
    // Xử lý sau khi thêm thành công
}
else
{
    // Xử lý sau khi thêm thất bại
}

// B4. Đóng kết nối
conn.Close();
```

.Net Framework DATA PROVIDER

- **Đối tượng DataReader**

- Đọc các dòng dữ liệu (từ trong CSDL) một cách tuần tự từ đầu đến cuối (không theo chiều ngược lại)
- Chỉ đọc dữ liệu ra, không cập nhật ngược lại dữ liệu vào CSDL

Tên Provider	Class DataReader	
.Net Framework Data Provider for ODBC	OdbcDataReader	
.Net Framework Data Provider for OleDb	OleDbDataReader	
.Net Framework Data Provider for SQL Server	SqlDataReader	XYZDataReader
.Net Framework Data Provider for Oracle	OracleDataReader	

.Net Framework DATA PROVIDER

- **Đối tượng DataReader**

- Khai báo:

- ```
XYZDataReader <tên_biến> = <đối_tượng_Command>.ExecuteReader();
```

- Một số thuộc tính:

- **bool HasRow**

- **True**: đối tượng DataReader có chứa ít nhất một dòng dữ liệu

- **False**: đối tượng DataReader không chứa dòng dữ liệu nào

- **int FieldCount**: số cột trong dòng mà đối tượng DataReader đang đọc

- **bool IsClosed**

- **True**: đối tượng DataReader đã bị đóng lại

- **False**: đối tượng DataReader chưa bị đóng

- Lấy giá trị của ô dữ liệu (Cell) theo chỉ số cột hoặc tên cột

- **object <đối\_tượng\_DataReader>[<chỉ số cột>]**

- **object <đối\_tượng\_DataReader>[“tên\_cột”]**

# .Net Framework DATA PROVIDER

- **Đối tượng DataReader**

- Một số phương thức:

- `bool Read()`

- Đọc 1 dòng dữ liệu

- Trả về true nếu đối tượng DataReader đọc được một dòng dữ liệu. Ngược lại trả về false

- `bool IsDBNull(int index)`

- Kiểm tra giá trị tại ô có chỉ số index của dòng đang đọc có phải là NULL hay không?

- Giá trị NULL trả về true. Ngược lại trả về false

- `void Close()`

- Đóng đối tượng DataReader

# .Net Framework DATA PROVIDER

- **Đối tượng DataReader**

- Một số phương thức lấy giá trị tại ô có chỉ số index của dòng đang đọc
  - `bool GetBoolean(int index)`
  - `byte GetByte(int index)`
  - `char GetChar(int index)`
  - `DateTime GetDateTime(int index)`
  - `decimal GetDecimal(int index)`
  - `double GetDouble(int index)`
  - `float GetFloat(int index)`
  - `short GetInt16(int index)`
  - `int GetInt32(int index)`
  - `long GetInt64(int index)`
  - `string GetString(int index)`
  - `object GetValue(int index)`

# .Net Framework DATA PROVIDER

- **Đối tượng DataReader**

- Ví dụ: lấy danh sách sinh viên (bảng SINH\_VIEN trong CSDL DemoDB)

The screenshot shows a Windows application window titled "Quản Lý Sinh Viên". The main title bar has three buttons: a minus sign for minimize, a square for maximize/minimize, and a red X for close. Below the title bar is a header section with the text "DANH SÁCH SINH VIÊN" in bold. The main content area is a table displaying ten student records:

|            |                  |            |     |     |
|------------|------------------|------------|-----|-----|
| 0306171020 | Hồ Tiến Đạt      | 04/01/1999 | Nam | 7.4 |
| 0306171021 | Triệu Văn Định   | 04/04/1999 | Nam | 7.8 |
| 0306171023 | Phùng Hiếu Đông  | 18/08/1999 | Nam | 8.1 |
| 0306171024 | Bùi Văn Đồng     | 29/10/1999 | Nam | 6.7 |
| 0306171025 | Mai Thanh Hải    | 16/02/1999 | Nam | 7.5 |
| 0306171027 | Nguyễn Ngọc Hiền | 02/04/1999 | Nữ  | 7.3 |
| 0306171029 | Trần Đình Hiếu   | 09/09/1999 | Nam | 8.7 |
| 0306171031 | Phạm Huy Hoàng   | 22/01/1999 | Nam | 7.6 |
| 0306171035 | Trần Quốc Hưng   | 21/07/1999 | Nam | 5.9 |
| 0306171036 | Trần Công Khanh  | 10/12/1999 | Nam | 6.6 |

# .Net Framework DATA PROVIDER

- **Đối tượng DataReader**

- Ví dụ: lấy danh sách sinh viên (bảng SINH\_VIEN trong CSDL DemoDB)

```
// B1. Tạo đối tượng kết nối
SqlConnection conn = new SqlConnection("Data Source=THANHTUAN-PC;Initial Catalog=DemoDB;Integrated Security=true");

// B2. Mở kết nối
conn.Open();

// B3. Thực hiện truy vấn số lượng sinh viên
string CauTruyVan = "SELECT * FROM SINH_VIEN";
SqlCommand com = new SqlCommand(CauTruyVan, conn);

SqlDataReader dr = com.ExecuteReader();
// Đọc tất cả dữ liệu từ DataReader
while(dr.Read())
{
 string MSSV = "";
 if (!dr.IsDBNull(0))
 MSSV = (string)dr[0];

 string HoTen = "";
 if (!dr.IsDBNull(1))
 HoTen = (string)dr[1];

 DateTime NgaySinh = new DateTime();
 if (!dr.IsDBNull(2))
 NgaySinh = (DateTime)dr[2];

 bool GioiTinh = false;
 if (!dr.IsDBNull(3))
 GioiTinh = (bool)dr[3];

 double DiemTB = 0;
 if (!dr.IsDBNull(4))
 DiemTB = (double)dr[4];

 // Thêm sinh viên vào ListBox
 lsbSinhVien.Items.Add(MSSV + "\t" + HoTen + "\t" + NgaySinh.ToString("dd/MM/yyyy") + "\t" + (GioiTinh ? "Nam" : "Nữ") + "\t" + DiemTB);
}
dr.Close();
// B4. Đóng kết nối
conn.Close();
```

# .Net Framework DATA PROVIDER

- **Đối tượng DataReader**

- Ví dụ: lấy danh sách sinh viên (bảng SINH\_VIEN trong CSDL DemoDB)

```
// B1. Tạo đối tượng kết nối
SqlConnection conn = new SqlConnection("Data Source=THANHTUAN-PC;Initial Catalog=DemoDB;Integrated Security=true");

// B2. Mở kết nối
conn.Open();

// B3. Thực hiện truy vấn số lượng sinh viên
string CauTruyVan = "SELECT * FROM SINH_VIEN";
SqlCommand com = new SqlCommand(CauTruyVan, conn);

SqlDataReader dr = com.ExecuteReader();
// Đọc tất cả dữ liệu từ DataReader
while(dr.Read())
{
 string MSSV = "";
 if (!dr.IsDBNull(0))
 MSSV = (string)dr["MSSV"];

 string HoTen = "";
 if (!dr.IsDBNull(1))
 HoTen = (string)dr["HoTen"];

 DateTime NgaySinh = new DateTime();
 if (!dr.IsDBNull(2))
 NgaySinh = (DateTime)dr["NgaySinh"];

 bool GioiTinh = false;
 if (!dr.IsDBNull(3))
 GioiTinh = (bool)dr["GioiTinh"];

 double DiemTB = 0;
 if (!dr.IsDBNull(4))
 DiemTB = (double)dr["DiemTB"];

 // Thêm sinh viên vào ListBox
 lsbSinhVien.Items.Add(MSSV + "\t" + HoTen + "\t" + NgaySinh.ToString("dd/MM/yyyy") + "\t" + (GioiTinh ? "Nam" : "Nữ") + "\t" + DiemTB);
}

dr.Close();
// B4. Đóng kết nối
conn.Close();
```

# .Net Framework DATA PROVIDER

- **Đối tượng DataReader**

- Ví dụ: lấy danh sách sinh viên (bảng SINH\_VIEN trong CSDL DemoDB)

```
// B1. Tạo đối tượng kết nối
SqlConnection conn = new SqlConnection("Data Source=THANHTUAN-PC;Initial Catalog=DemoDB;Integrated Security=true;");

// B2. Mở kết nối
conn.Open();

// B3. Thực hiện truy vấn số lượng sinh viên
string CauTruyVan = "SELECT * FROM SINH_VIEN";
SqlCommand com = new SqlCommand(CauTruyVan, conn);

SqlDataReader dr = com.ExecuteReader();
// Đọc tất cả dữ liệu từ DataReader
while(dr.Read())
{
 string MSSV = "";
 if (!dr.IsDBNull(0))
 MSSV = dr.GetString(0);

 string HoTen = "";
 if (!dr.IsDBNull(1))
 HoTen = dr.GetString(1);

 DateTime NgaySinh = new DateTime();
 if (!dr.IsDBNull(2))
 NgaySinh = dr.GetDateTime(2);

 bool GioiTinh = false;
 if (!dr.IsDBNull(3))
 GioiTinh = dr.GetBoolean(3);

 double DiemTB = 0;
 if (!dr.IsDBNull(4))
 DiemTB = dr.GetDouble(4);

 // Thêm sinh viên vào ListBox
 lsbSinhVien.Items.Add(MSSV + "\t" + HoTen + "\t" + NgaySinh.ToString("dd/MM/yyyy") + "\t" + (GioiTinh ? "Nam" : "Nữ") + "\t" + DiemTB);
}

dr.Close();
// B4. Đóng kết nối
conn.Close();
```

# .Net Framework DATA PROVIDER

## BÀI TẬP:

- Viết Store Procedure
    - Cập nhật sinh viên
    - Xoá sinh viên
    - Lấy danh sách sinh viên
  - Viết lại các Event-Handler xử lý
    - Cập nhật sinh viên
    - Xoá sinh viên
    - Lấy danh sách sinh viên
- (Gọi thực thi Store Procedure thay cho Command Text)**

# ĐIỀU KHIỂN LISTVIEW

- **Listview** là control dùng để trình bày danh sách các mục bằng 1 trong các cách hiển thị (**View**) sau:
  - LargeIcon
  - Details
  - SmallIcon
  - List
  - Tile
- Các item trong ListView là một đối tượng **ListViewItem**

# ĐIỀU KHIỂN LISTVIEW

## Details

| Tên sản phẩm                                                                                    | Giá (vnđ) | Mô tả sản phẩm                                        |
|-------------------------------------------------------------------------------------------------|-----------|-------------------------------------------------------|
|  Cà phê sữa    | 15.000    | Hương vị cà phê thơm ngon của chúng tôi...            |
|  Nuoc trái cây | 25.000    | Nuoc trái cây ép nguyên chất đến từ trái cây tươi...  |
|  Món khai vị   | 50.000    | Ngoài ra, cửa hàng còn phục vụ các món ăn việt nam... |

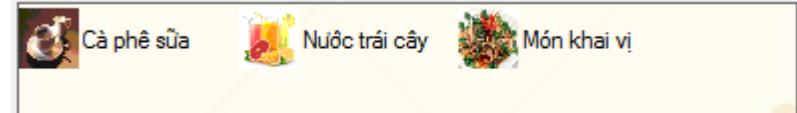
## Tile



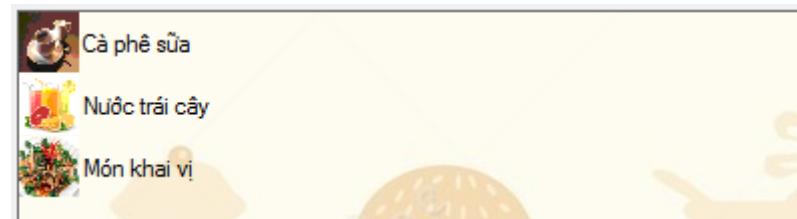
## LargelIcon



## SmallIcon



## List



# ĐIỀU KHIỂN LISTVIEW

- Một số thuộc tính
  - **Items**: danh sách các đối tượng ListViewItem được hiển thị trong ListView
  - **Columns**: danh sách các đối tượng ColumnHeader được dùng trong *Details View*
  - **SelectedItems**, **SelectedIndices**
  - **LargeImageList**
  - **SmallImageList**
  - **MultiSelect**
  - **View**: *LargeIcon*, *SmallIcon*, *List*, *Details*, *Tile*
  - **Sorting**
  - **FullRowSelect**
  - **Checkboxes**
  - **GridLines**

# ĐIỀU KHIỂN LISTVIEW

- Một số phương thức
  - `Clear()`
- Một số sự kiện
  - `SelectedIndexChanged`
  - `Click`
- Thêm một item đơn giản
  - `ListViewItem item = new ListViewItem(string);`
  - `listView.Items.Add(item);`
- Thêm item có ảnh
  - *Tạo imagelist*
  - *Gán imagelist cho LargeImageList/SmallImageList*
  - *item.ImageIndex = chỉ số hình trong imagelist*
- Thêm item có nhiều subitem
  - `string[] data = {string1, string2, ...};`
  - `ListViewItem item = new ListViewItem(data);`
  - `listView.Items.Add(item);`

# ĐIỀU KHIỂN LISTVIEW

- Thêm item có nhiều subitem
  - *ListViewItem item = new ListViewItem(string1);*
  - *item.SubItems.Add(string2);*
  - ...
  - *listView.Items.Add(item);*
- Nhận dữ liệu trên ListView
  - *ListViewItem item = lstSignature.SelectedItems[0];*
  - *string sub1 = item.SubItems[0].Text;*
  - *string sub2 = item.SubItems[1].Text;*
  - ...
- Tạo detail view
  - **View:** Detail
  - **GridLines:** true
  - **Columns:** Thêm các column

# ĐIỀU KHIỂN LISTVIEW

| Column0     | Column1     | Column2     | ... | ColumnN     |
|-------------|-------------|-------------|-----|-------------|
| SubItems[0] | SubItems[1] | SubItems[2] | ... | SubItems[N] |
|             |             |             |     |             |
|             |             |             |     |             |
|             |             |             |     |             |

ListViewItem

# ÁNH XẠ CLASS vs TABLE

- Table **SINH\_VIEN**

|    | Column Name | Data Type    |
|----|-------------|--------------|
| PK | MSSV        | nvarchar(10) |
|    | HoTen       | nvarchar(50) |
|    | NgaySinh    | date         |
|    | GioiTinh    | bit          |
|    | DiemTB      | float        |

```
public class SinhVien {
 // Properties
 public string MSSV { set; get; }
 public string Hoten { set; get; }
 public DateTime NgaySinh { set; get; }
 public bool GioiTinh { set; get; }
 public double DiemTB { set; get; }
}
```

# ÁNH XẠ CLASS vs TABLE

- Table SINH\_VIEN

| Column Name | Data Type    |
|-------------|--------------|
| MSSV        | nvarchar(10) |
| HoTen       | nvarchar(50) |
| NgaySinh    | date         |
| GioiTinh    | bit          |
| DiemTB      | float        |

```
public class SinhVien {

 // Properties
 public string MSSV { set; get; }
 public string Hoten { set; get; }
 public DateTime NgaySinh { set; get; }
 public bool GioiTinh { set; get; }
 public double DiemTB { set; get; }

}
```

# ÁNH XẠ CLASS vs TABLE

- Table SINH\_VIEN

|    | Column Name | Data Type    |
|----|-------------|--------------|
| PK | MSSV        | nvarchar(10) |
|    | HoTen       | nvarchar(50) |
|    | NgaySinh    | date         |
|    | GioiTinh    | bit          |
|    | DiemTB      | float        |

```
public class SinhVien {

 // Properties
 public string MSSV { set; get; }
 public string Hoten { set; get; }
 public DateTime NgaySinh { set; get; }
 public bool GioiTinh { set; get; }
 public double DiemTB { set; get; }

}
```

# ÁNH XẠ CLASS vs TABLE

- Table SINH\_VIEN

| Column Name | Data Type    |
|-------------|--------------|
| MSSV        | nvarchar(10) |
| HoTen       | nvarchar(50) |
| NgaySinh    | date         |
| GioiTinh    | bit          |
| DiemTB      | float        |

```
public class SinhVien {

 // Properties
 public string MSSV { set; get; }
 public string Hoten { set; get; }
 public DateTime NgaySinh { set; get; }
 public bool GioiTinh { set; get; }
 public double DiemTB { set; get; }

}
```

# ÁNH XẠ CLASS vs TABLE

- Table SINH\_VIEN

|    | Column Name | Data Type    |
|----|-------------|--------------|
| PK | MSSV        | nvarchar(10) |
|    | HoTen       | nvarchar(50) |
|    | NgaySinh    | date         |
|    | GioiTinh    | bit          |
|    | DiemTB      | float        |

```
public class SinhVien {

 // Properties
 public string MSSV { set; get; }
 public string Hoten { set; get; }
 public DateTime NgaySinh { set; get; }
 public bool GioiTinh { set; get; }
 public double DiemTB { set; get; }

}
```

# ÁNH XẠ CLASS vs TABLE

- Table SINH\_VIEN

|    | Column Name | Data Type    |
|----|-------------|--------------|
| PK | MSSV        | nvarchar(10) |
|    | HoTen       | nvarchar(50) |
|    | NgaySinh    | date         |
|    | GioiTinh    | bit          |
|    | DiemTB      | float        |

```
public class SinhVien {

 // Properties
 public string MSSV { set; get; }
 public string Hoten { set; get; }
 public DateTime NgaySinh { set; get; }
 public bool GioiTinh { set; get; }
 public double DiemTB { set; get; }

}
```

# ÁNH XẠ CLASS vs TABLE

- Table SINH\_VIEN

|    | Column Name | Data Type    |
|----|-------------|--------------|
| PK | MSSV        | nvarchar(10) |
|    | HoTen       | nvarchar(50) |
|    | NgaySinh    | date         |
|    | GioiTinh    | bit          |
|    | DiemTB      | float        |

```
public class SinhVien {

 // Properties
 public string MSSV { set; get; }
 public string Hoten { set; get; }
 public DateTime NgaySinh { set; get; }
 public bool GioiTinh { set; get; }
 public double DiemTB { set; get; }

}
```

# ÁNH XẠ CLASS vs TABLE

- Table SINH\_VIEN

|    | Column Name | Data Type    |
|----|-------------|--------------|
| PK | MSSV        | nvarchar(10) |
|    | HoTen       | nvarchar(50) |
|    | NgaySinh    | date         |
|    | GioiTinh    | bit          |
|    | DiemTB      | float        |

```
public class SinhVien {

 // Properties
 public string MSSV { set; get; }
 public string Hoten { set; get; }
 public DateTime NgaySinh { set; get; }
 public bool GioiTinh { set; get; }
 public double DiemTB { set; get; }

}
```

# ÁNH XẠ CLASS vs TABLE

- Table SINH\_VIEN

|    | Column Name | Data Type    |
|----|-------------|--------------|
| PK | MSSV        | nvarchar(10) |
|    | HoTen       | nvarchar(50) |
|    | NgaySinh    | date         |
|    | GioiTinh    | bit          |
|    | DiemTB      | float        |

```
public class SinhVien {

 // Properties
 public string MSSV { set; get; }
 public string Hoten { set; get; }
 public DateTime NgaySinh { set; get; }
 public bool GioiTinh { set; get; }
 public double DiemTB { set; get; }

}
```

# ÁNH XẠ CLASS vs TABLE

- Table SINH\_VIEN

|    | Column Name | Data Type    |
|----|-------------|--------------|
| PK | MSSV        | nvarchar(10) |
|    | HoTen       | nvarchar(50) |
|    | NgaySinh    | date         |
|    | GioiTinh    | bit          |
|    | DiemTB      | float        |

```
public class SinhVien {

 // Properties
 public string MSSV { set; get; }
 public string Hoten { set; get; }
 public DateTime NgaySinh { set; get; }
 public bool GioiTinh { set; get; }
 public double DiemTB { set; get; }

}
```

# ÁNH XẠ CLASS vs TABLE

- Table SINH\_VIEN

|    | Column Name | Data Type    |
|----|-------------|--------------|
| PK | MSSV        | nvarchar(10) |
|    | HoTen       | nvarchar(50) |
|    | NgaySinh    | date         |
|    | GioiTinh    | bit          |
|    | DiemTB      | float        |

```
public class SinhVien {

 // Properties
 public string MSSV { set; get; }
 public string Hoten { set; get; }
 public DateTime NgaySinh { set; get; }
 public bool GioiTinh { set; get; }
 public double DiemTB { set; get; }

}
```

# ÁNH XẠ CLASS vs TABLE

- Table SINH\_VIEN

|    | Column Name | Data Type    |
|----|-------------|--------------|
| PK | MSSV        | nvarchar(10) |
|    | HoTen       | nvarchar(50) |
|    | NgaySinh    | date         |
|    | GioiTinh    | bit          |
|    | DiemTB      | float        |

Custom property

```
public class SinhVien {

 // Properties
 public string MSSV { set; get; }
 public string Hoten { set; get; }
 public DateTime NgaySinh { set; get; }
 public bool GioiTinh { set; get; }
 public string GioiTinh_HienThi { set; get; }
 public double DiemTB { set; get; }

}
```

# ÁNH XẠ CLASS vs TABLE

- Danh sách các **Record trong Table** → **List đối tượng**

```
public List<SinhVien> LayDanhSach()
{
 string ChuoiKetNoi = "...";
 SqlConnection conn = new SqlConnection(ChuoiKetNoi);
 conn.Open();

 string CauTruyVan = "SELECT * FROM SINH_VIEN";
 SqlCommand com = new SqlCommand(CauTruyVan, conn);

 SqlDataReader dr = com.ExecuteReader();

 List<SinhVien> DanhSachSinhVien = new List<SinhVien>();
 while(dr.Read())
 {
 // Tạo đối tượng SinhVien
 SinhVien SV = new SinhVien();

 if(!dr.IsDBNull(0))
 SV.MSSV = dr.GetString(0);

 if(!dr.IsDBNull(1))
 SV.Hoten = dr.GetString(1);

 if(!dr.IsDBNull(2))
 SV.NgaySinh = dr.GetDateTime(2);

 if(!dr.IsDBNull(3))
 SV.GioiTinh = dr.GetBoolean(3);

 if(!dr.IsDBNull(4))
 SV.DiemTB = dr.GetDouble(4);

 // Thêm đối tượng SinhVien vào danh sách
 DanhSachSinhVien.Add(SV);
 }
 dr.Close();
 conn.Close();
}

return DanhSachSinhVien;
}
```

# ÁNH XẠ CLASS vs TABLE

- Danh sách các **Record trong Table → List đối tượng**

```
public List<SinhVien> LayDanhSach()
{
 string ChuoiKetNoi = "...";
 SqlConnection conn = new SqlConnection(ChuoiKetNoi);
 conn.Open();

 string CauTruyVan = "SELECT * FROM SINH_VIEN";
 SqlCommand com = new SqlCommand(CauTruyVan, conn);

 SqlDataReader dr = com.ExecuteReader();

 List<SinhVien> DanhSachSinhVien = new List<SinhVien>();
 while(dr.Read())
 {
 // Tạo đối tượng SinhVien
 SinhVien SV = new SinhVien();

 if(!dr.IsDBNull(0))
 SV.MSSV = dr.GetString(0);

 if(!dr.IsDBNull(1))
 SV.Hoten = dr.GetString(1);

 if(!dr.IsDBNull(2))
 SV.NgaySinh = dr.GetDateTime(2);

 if(!dr.IsDBNull(3))
 SV.GioiTinh = dr.GetBoolean(3);

 if(!dr.IsDBNull(4))
 SV.DiemTB = dr.GetDouble(4);

 // Thêm đối tượng SinhVien vào danh sách
 DanhSachSinhVien.Add(SV);
 }
 dr.Close();
 conn.Close();
}

return DanhSachSinhVien;
}
```



```
if(!dr.IsDBNull(3))
{
 SV.GioiTinh = dr.GetBoolean(3);
 SV.GioiTinh_HienThi = (SV.GioiTinh ? "Nam" : "Nữ");
}
```

# ÁNH XẠ CLASS vs TABLE

- Kiểu dữ liệu MS SQL SERVER vs Visual C#

| MS SQL SERVER                                  | C#       |
|------------------------------------------------|----------|
| char, nchar, varchar,<br>nvarchar, text, ntext | string   |
| int                                            | int      |
| bigint                                         | long     |
| bit                                            | bool     |
| date, datetime                                 | DateTime |
| float                                          | double   |

# ĐIỀU KHIỂN DATAGRIDVIEW

Quản Lý Sinh Viên

Danh sách sinh viên:

|   | MSSV       | Họ Tên           | NgaySinh   | GioiTinh                            | DiemTB | MaLop |  |
|---|------------|------------------|------------|-------------------------------------|--------|-------|--|
| ▶ | 0306171020 | Hồ Tiễn Đạt      | 1/4/1999   | <input checked="" type="checkbox"/> | 7.4    | 1     |  |
|   | 0306171021 | Triệu Văn Định   | 4/4/1999   | <input checked="" type="checkbox"/> | 7.8    | 2     |  |
|   | 0306171023 | Phùng Hiếu Đông  | 8/18/1999  | <input checked="" type="checkbox"/> | 8.1    | 1     |  |
|   | 0306171024 | Bùi Văn Đồng     | 10/29/1999 | <input checked="" type="checkbox"/> | 6.7    | 1     |  |
|   | 0306171025 | Mai Thanh Hải    | 2/16/1999  | <input checked="" type="checkbox"/> | 7.5    | 6     |  |
|   | 0306171027 | Nguyễn Ngọc Hiền | 4/2/1999   | <input type="checkbox"/>            | 7.4    | 6     |  |

# ĐIỀU KHIỂN DATAGRIDVIEW

- **DataGridView** là điều khiển chuyên dùng cho việc hiển thị dữ liệu dưới dạng bảng
- Một số thuộc tính
  - *AllowUserToAddRows, AllowUserToDeleteRows, AllowUserToOrderColumns, AllowUserToResizeColumns, AllowUserToResizeRows*
  - *Columns, Rows*: *Danh sách cột, dòng trong Datagridview*
  - *CurrentCell, CurrentRow*
  - *DataSource*: *Nguồn dữ liệu hiển thị trên Datagridview*
  - *MultiSelect*
  - *SelectedRows, SelectedCells*

# ĐIỀU KHIỂN DATAGRIDVIEW

- Một số phương thức
  - *ClearSelection()*
  - *SelectAll()*
- Một số sự kiện
  - *Click, CellClick, CellContentClick*
  - *UserAddedRow, UserDeletedRow*

# ĐIỀU KHIỂN DATAGRIDVIEW

- Gán nguồn dữ liệu vào DataGridView
  - `<datagridview>.DataSource = <danh_sách_đối_tượng>;`
  - `<datagridview>.DataSource = <đối_tượng_BindingSource>;`
  - `<datagridview>.DataSource = <đối_tượng_DataTable>;`

# DATASET

- 

Database

Dataset

Table

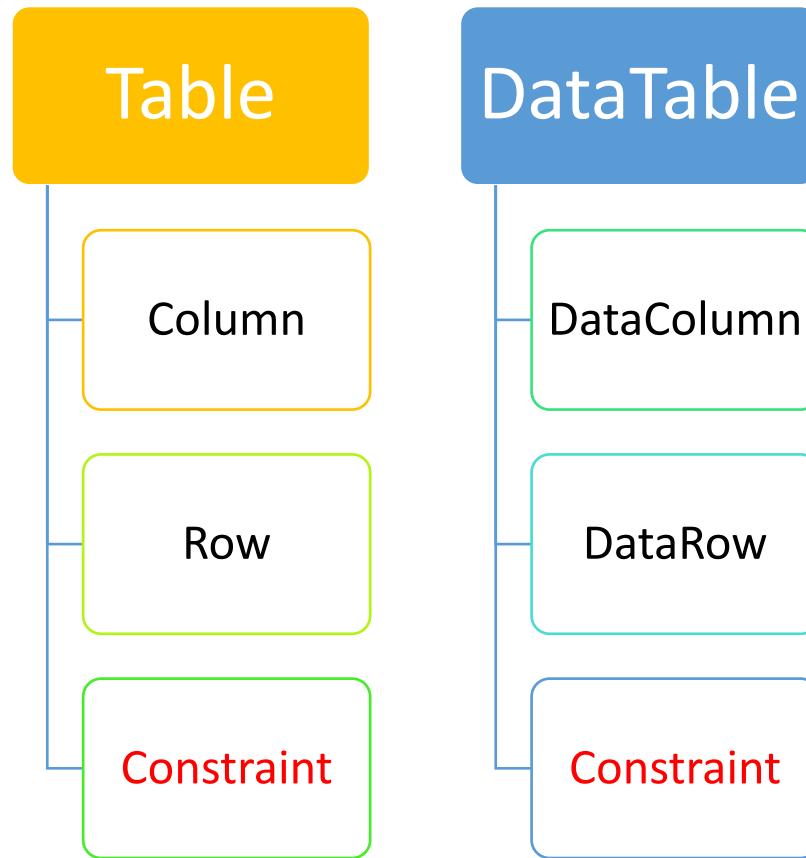
DataTable

Relation

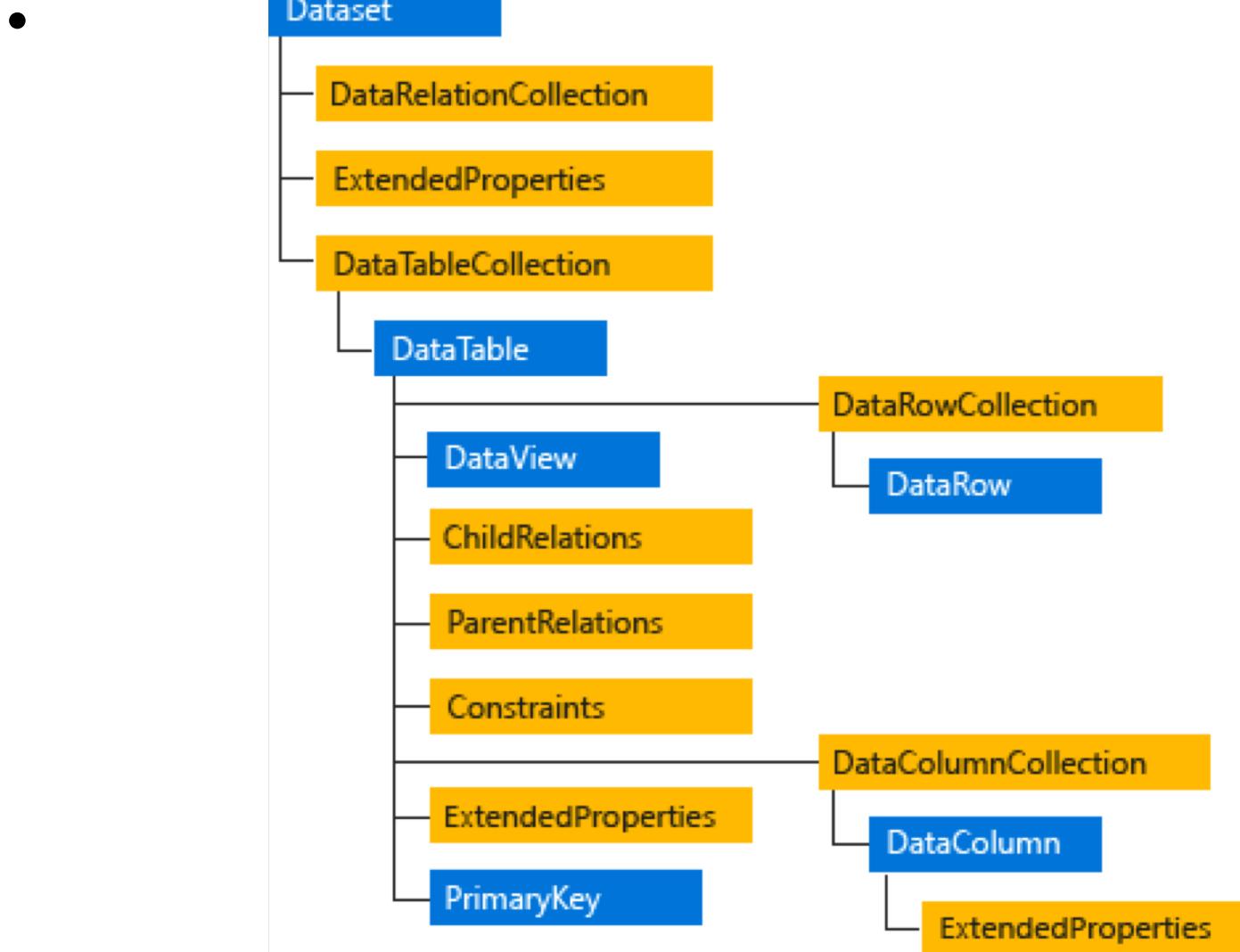
DataRelation

# DATASET

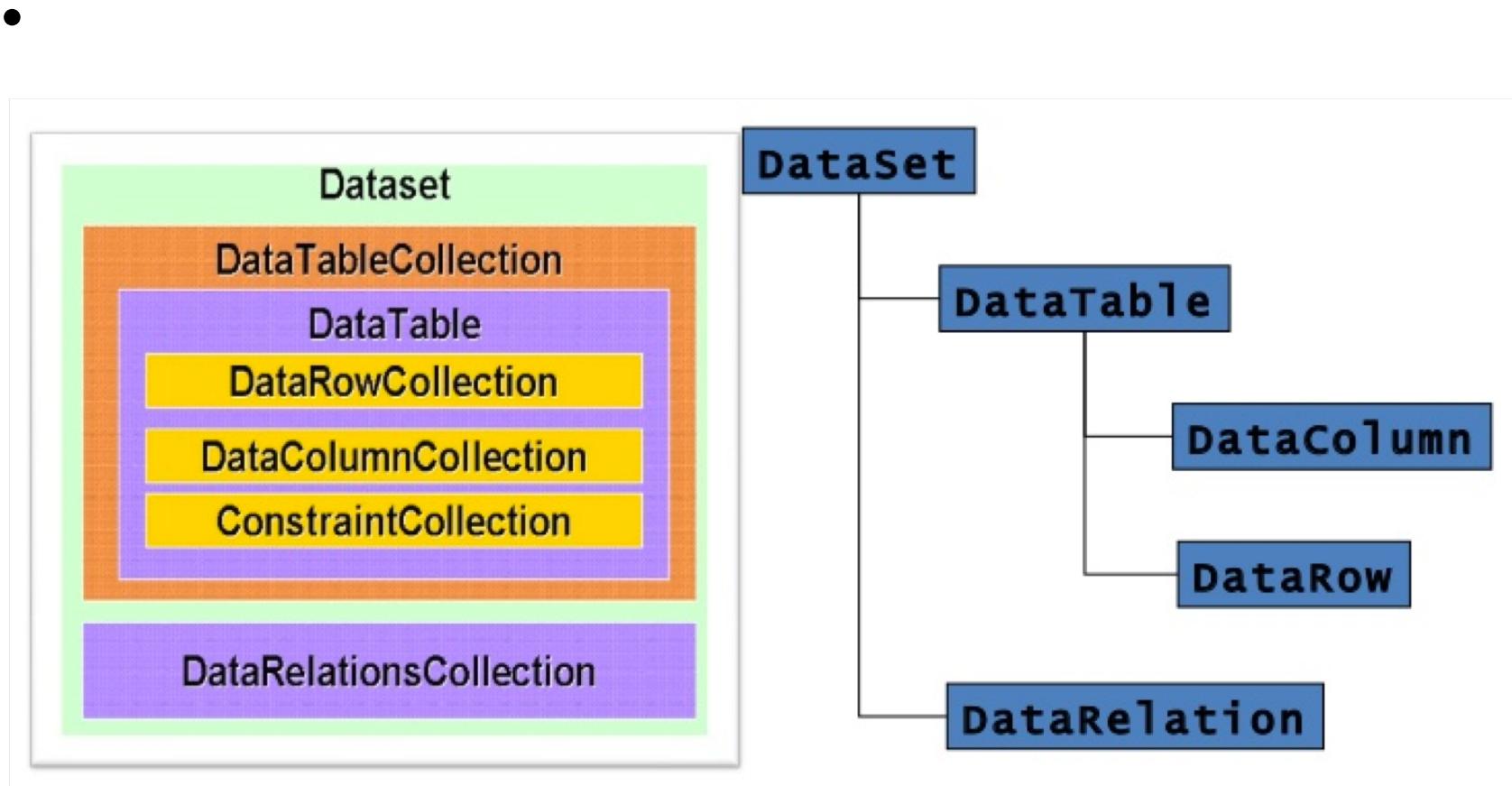
- 



# DATASET



# DATASET



# DATASET

- **DataTable**

- Namespace: *System.Data.DataTable*
- Định nghĩa một bảng dữ liệu trong đó:
  - Các cột được định nghĩa với **tên** và **kiểu dữ liệu**
  - Các dòng dữ liệu xuất hiện khi thực thi chương trình
- DataTable là kho chứa dữ liệu dựa trên bộ nhớ chính

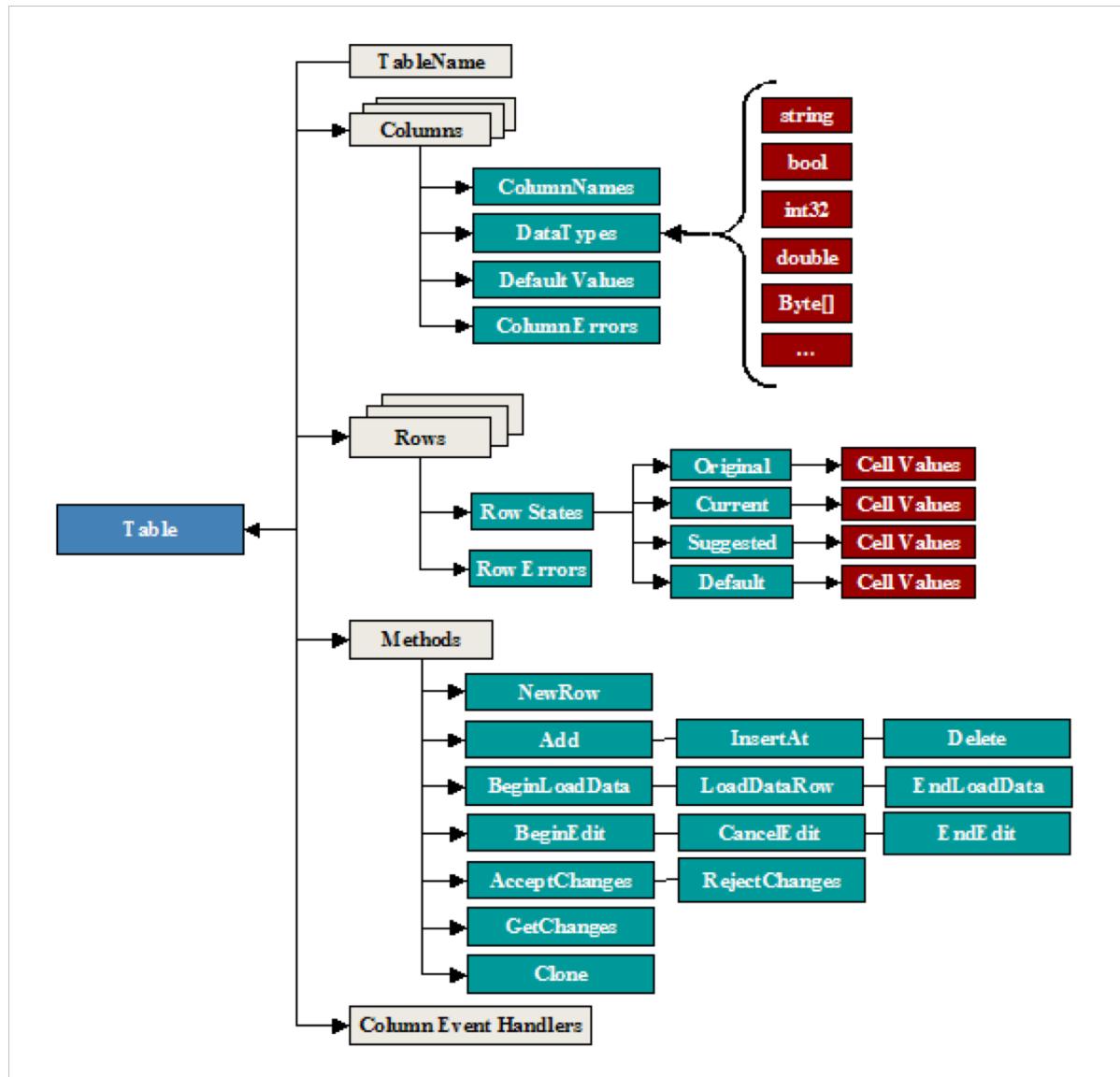
# DATASET

- **DataTable**

- Một số thuộc tính của lớp DataTable
  - DataSet
  - DefaultView
  - Columns, Rows
  - PrimaryKey
  - TableName
- Một số phương thức của lớp DataTable
  - AcceptChanges(), GetChanges(), RejectChanges()
  - NewRow()
  - Merge()
  - Clear()
  - Copy(), Clone()
  - ReadXML(), WriteXML()

# DATASET

- DataTable



# DATASET

- **DataTable**

- Tạo DataTable

```
DataTable customer = new DataTable("Customer");
```

// Hoặc

```
DataTable customer = new DataTable();
```

```
customer.TableName = "Customer";
```

- Thêm cột (*DataColumn*) vào DataTable

```
DataColumn keyField = new DataColumn("ID", typeof(long));
customer.Columns.Add(keyField);
```

# DATASET

- **DataTable**

- Một số kiểu dữ liệu của thể hiện *DataColumn*:

*Boolean*



*bool*

*Byte*



*byte*

*Char*



*char*

*DateTime*



*DateTime*

*Decimal*



*decimal*

*Double*



*double*

*Int16*



*short*

*Int32*



*int*

*Int64*



*long*

*Single*



*float*

*String*



*string*

*UInt16*



*ushort*

*UInt32*



*uint*

*UInt64*



*ulong*

Arrays of *Byte* (có một số hạn chế)

# DATASET

- **DataTable**

- Một số thuộc tính của lớp DataColumn
  - *AllowDBNull*
  - *AutoIncrement, AutoIncrementSeed, AutoIncrementStep*
  - *Caption*
  - *ColumnName*
  - *DataType*
  - *DateTimeMode*
  - *DefaultValue*
  - *MaxLength*
  - *ReadOnly*
  - *Unique*

# DATASET

- **DataTable**

- Thiết lập khóa chính

```
customer.PrimaryKey = new DataColumn[] {customer.Columns["ID"]};
```

- Tạo khoá ngoại

```
<datatable_child>.Constraints.Add("<tên_khoá_ngoại>"
 <datatable_parent>.Columns("<tên_trường>"),
 <datatable_child>.Columns("<tên_trường>"));
```

// Hoặc

```
ForeignKeyConstraint foreignKeyConstraint = new
ForeignKeyConstraint("<tên_khoá_ngoại>",
 <datatable_parent>.Columns("<tên_trường>"),
 <datatable_child>.Columns("<tên_trường>"));

//Setting Rule of constraint
foreignKeyConstraint.DeleteRule = Rule.SetNull;
foreignKeyConstraint.UpdateRule = Rule.Cascade;
<datatable_child>.Constraints.Add(foreignKeyConstraint);
```

# DATASET

- **DataTable**

- Thêm dòng dữ liệu (DataRow) vào DataTable (theo 3 bước sau)

- **Tạo một đối tượng DataRow**

```
DataRow oneRow = someTable.NewRow();
```

- **Lưu dữ liệu vào đối tượng DataRow đã tạo ở trên**

```
oneRow.Item["ID"] = 123; // by column name
oneRow.Item[0] = 123; // by column position
```

```
 DataColumn whichColumn = someTable.Columns[0];
oneRow.Item[whichColumn] = 123; // by column instance
oneRow["Comments"] = System.DBNull.Value; // NULL value
```

- **Thêm đối tượng DataRow vào DataTable**

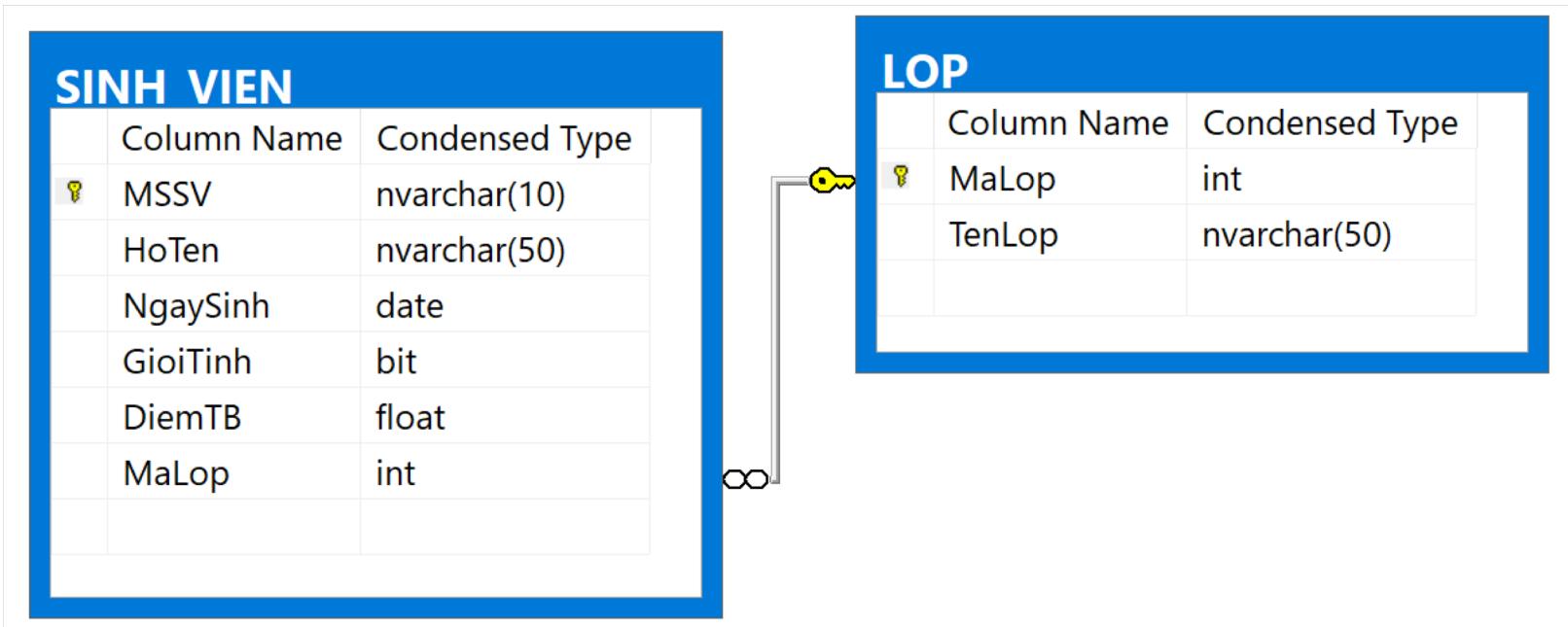
```
someTable.Rows.Add(oneRow);
```

# DATASET

- **DataTable**

- Bài tập 1:

- *Tạo DataTable SinhVien và Lop có cấu trúc tương tự như bảng SINH\_VIEN và LOP*



# DATASET

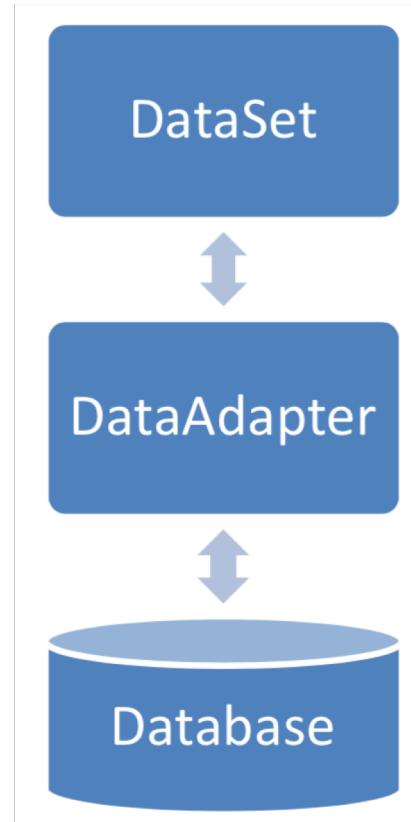
- **DataTable**

- Bài tập:

- *Thiết kế giao diện chương trình để người dùng có thể:*
  - *Xem danh sách sinh viên*
  - *Thêm sinh viên mới*
  - *Cập nhật sinh viên*
  - *Xoá sinh viên*

# .Net Framework DATA PROVIDER

- **Đối tượng DataAdapter:** thực hiện 2 chức năng
  - Đọc dữ liệu từ cơ sở dữ liệu vào DataSet / DataTable
  - Cập nhật dữ từ DataSet / DataTable ra cơ sở dữ liệu



# .Net Framework DATA PROVIDER

- Đối tượng DataAdapter

| Tên Provider                                       | Class Command     |                       |
|----------------------------------------------------|-------------------|-----------------------|
| .Net Framework Data Provider for <b>ODBC</b>       | OdbcDataAdapter   |                       |
| .Net Framework Data Provider for <b>OleDb</b>      | OleDbDataAdapter  |                       |
| .Net Framework Data Provider for <b>SQL Server</b> | SqlDataAdapter    |                       |
| .Net Framework Data Provider for <b>Oracle</b>     | OracleDataAdapter | <b>XYZDataAdapter</b> |

# .Net Framework DATA PROVIDER

- **Đối tượng DataAdapter**

- Khởi tạo đối tượng:

- `XYZDataAdapter <tên_bien> = new XYZDataAdapter();`
    - `XYZDataAdapter <tên_bien> = new XYZDataAdapter(<câu_lệnh_SQL_SELECT>, <chuỗi_kết_nối>);`
    - `XYZDataAdapter <tên_bien> = new XYZDataAdapter(<câu_lệnh_SQL_SELECT>, <đối_tượng_Connection>);`

# .Net Framework DATA PROVIDER

- **Đối tượng DataAdapter**

- Một số thuộc tính:
  - XYZCommand SelectCommand
  - XYZCommand InsertCommand
  - XYZCommand UpdateCommand
  - XYZCommand DeleteCommand
- Một số phương thức:
  - int Fill(DataSet dataSet)
  - int Fill(DataSet dataSet, string srcTable)
  - int Fill(DataSet dataSet, int startRecord, int maxRecords, string srcTable)
  - int Update(DataSet dataSet)
  - int Update(DataTable dataTable)
  - int Update(DataSet dataSet, string srcTable)

# .Net Framework DATA PROVIDER

- **Đối tượng DataAdapter**

- Khai báo đối tượng Command sử dụng Parameter

```
<command>.Parameters.Add(<tên_parameter>,
<KDL_parameter>, <độ_dài_paramter>,
<tên_cột_tương_ứng_trong_CSDL>);
```

## Ví dụ:

```
SqlCommand updateCom = new SqlCommand("UPDATE LOP SET TenLop = @ten WHERE MALOP = @ma", conn);
updateCom.Parameters.Add("@ten", SqlDbType.NVarChar, 50, "TenLop");
updateCom.Parameters.Add("@ma", SqlDbType.Int, 4, "MaLop");
```

# DATASET

- **DataSet**

- Namespace: **System.Data**
- Tập hợp các DataTable
- Khởi tạo đối tượng DataSet:
  - **DataSet <đặt tên biến> = new DataSet();** // DataSet được tạo sẽ có tên là **NewDataSet**
  - **DataSet <đặt tên biến> = new DataSet(<tên\_DataSet>);**

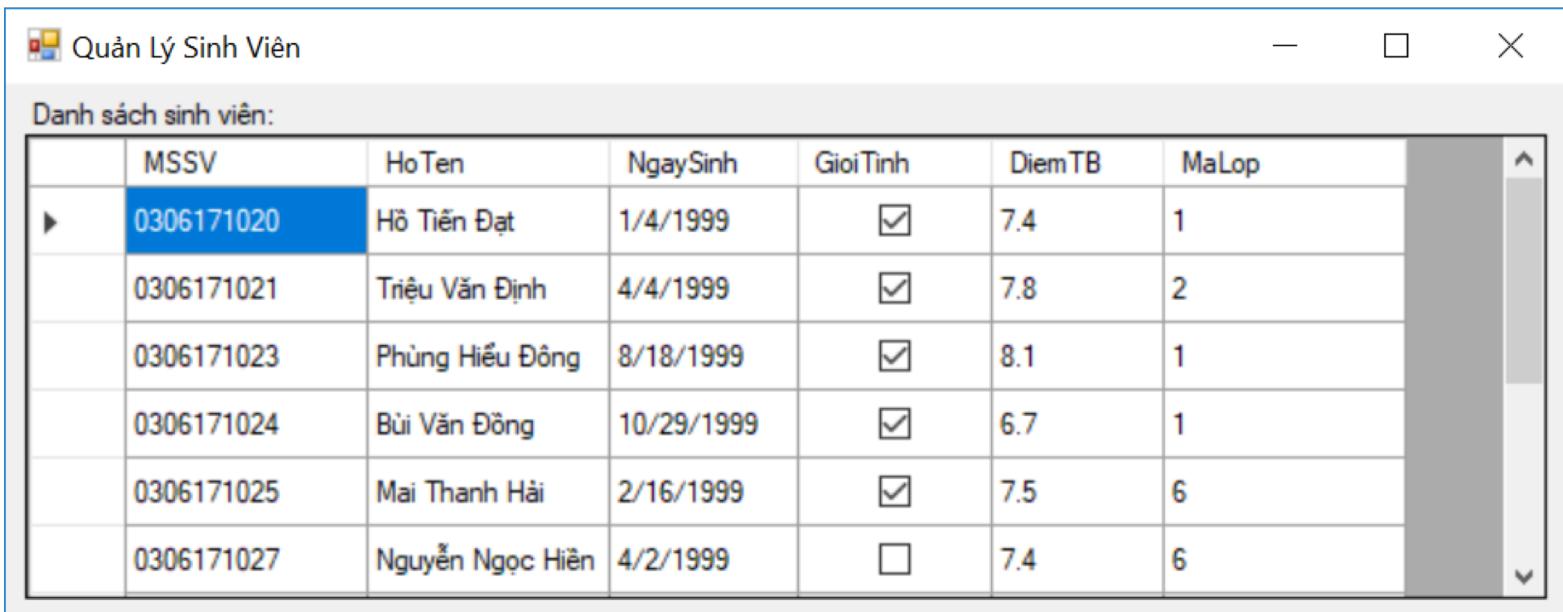
# DATASET

- **DataSet**

- Một số thuộc tính:
  - `string DataSetName`
  - `DataTableCollection Tables`
  - `DataRelationCollection Relations`
- Một số phương thức:
  - `void AcceptChanges()`
  - `void RejectChanges()`
  - `void Clear()`
  - `XMLReadMode ReadXML(string filename)`
  - `void WriteXML(string filename)`

# DATASET

- **DataSet**
  - Bài tập 2: Hiển thị danh sách sinh viên (Sử dụng DataSet)



The screenshot shows a Windows application window titled "Quản Lý Sinh Viên". The main title bar has standard window controls (minimize, maximize, close). Below the title bar, there is a header section labeled "Danh sách sinh viên:". A data grid table is displayed, showing the following student information:

|   | MSSV       | Họ Tên           | NgaySinh   | GioiTinh                            | Diem TB | MaLop |
|---|------------|------------------|------------|-------------------------------------|---------|-------|
| ▶ | 0306171020 | Hồ Tiên Đạt      | 1/4/1999   | <input checked="" type="checkbox"/> | 7.4     | 1     |
|   | 0306171021 | Triệu Văn Định   | 4/4/1999   | <input checked="" type="checkbox"/> | 7.8     | 2     |
|   | 0306171023 | Phùng Hiểu Đông  | 8/18/1999  | <input checked="" type="checkbox"/> | 8.1     | 1     |
|   | 0306171024 | Bùi Văn Đồng     | 10/29/1999 | <input checked="" type="checkbox"/> | 6.7     | 1     |
|   | 0306171025 | Mai Thanh Hải    | 2/16/1999  | <input checked="" type="checkbox"/> | 7.5     | 6     |
|   | 0306171027 | Nguyễn Ngọc Hiền | 4/2/1999   | <input type="checkbox"/>            | 7.4     | 6     |

# DATASET

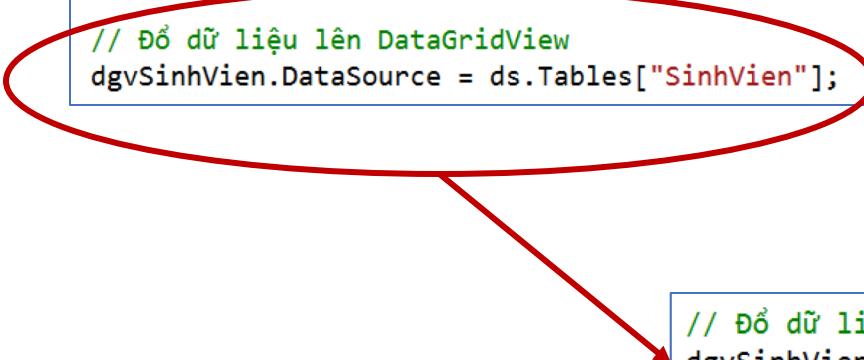
- **DataSet**

- Bài tập 2: Hiển thị danh sách sinh viên (Sử dụng DataSet)

```
DataSet ds = new DataSet();
SqlDataAdapter daSinhVien = new SqlDataAdapter("SELECT * FROM SINH_VIEN", Program.ChuoiKetNoi);

// Đổ dữ liệu vào DataTable SinhVien trong DataSet ds
daSinhVien.Fill(ds, "SinhVien");

// Đổ dữ liệu lên DataGridView
dgvSinhVien.DataSource = ds.Tables["SinhVien"];
```



```
// Đổ dữ liệu lên DataGridView
dgvSinhVien.DataSource = ds;
dgvSinhVien.DataMember = "SinhVien";
```

# DATASET

- **DataView**

- Dùng cho việc sắp xếp, lọc, tìm kiếm và thay đổi các dòng dữ liệu của một bảng
- Namespace: **System.Data**
- Khởi tạo đối tượng:
  - `DataView <tên_biến> = new DataView(<đối_tượng_DataTable>);`
  - `DataView <tên_biến> = <đối_tượng_DataTable>.DefaultView;`

# DATASET

- **DataView**

- Một số thuộc tính:
  - `int Count`
  - `string RowFilter`
  - `string Sort`
  - `DataTable Table`
- Một số phương thức:
  - `void Delete(int position)`
- **Các thay đổi trên DataView cũng thay đổi trên DataTable tạo ra DataView**

# DATASET

- **DataView**
  - Bài tập 3: Tìm kiếm (Lọc) Sinh viên theo Họ tên

The screenshot shows a Windows application window titled "Quản Lý Sinh Viên". At the top, there is a search bar labeled "Tìm theo tên:" containing the text "văn" and a "Tìm" button. Below the search bar, the text "Danh sách sinh viên:" is displayed. A table lists three students:

|   | MSSV       | Họ Tên         | NgaySinh   | GioiTinh                            | DiemTB | MaLop |
|---|------------|----------------|------------|-------------------------------------|--------|-------|
| ▶ | 0306171021 | Triệu Văn Định | 4/4/1999   | <input checked="" type="checkbox"/> | 7.8    | 2     |
|   | 0306171024 | Bùi Văn Đồng   | 10/29/1999 | <input checked="" type="checkbox"/> | 6.7    | 1     |
| * |            |                |            | <input type="checkbox"/>            |        |       |

# DATASET

- **DataView**

- Bài tập 3: Tìm kiếm (Lọc) Sinh viên theo Họ tên

```
// Tạo DataView từ Datatable
DataView dvSinhVien = ds.Tables["SinhVien"].DefaultView;

// Lọc dữ liệu trên DataView
dvSinhVien.RowFilter = string.Format("HoTen LIKE '{0}%", txtTen.Text);

// Đổ dữ liệu lên DataGridView
dgvSinhVien.DataSource = dvSinhVien;
```

`new DataView(ds.Tables["SinhVien"]);`

# DATASET

- **DataView**

- Bài tập 4: Tìm kiếm (Lọc) Sinh viên theo Lớp

The screenshot shows a Windows application window titled "Quản Lý Sinh Viên". At the top, there is a search bar labeled "Tim theo lop:" containing the text "CĐTH 17A" and a "Tim" button. Below the search bar, the text "Danh sách sinh viên:" is displayed. A data grid table follows, listing five students:

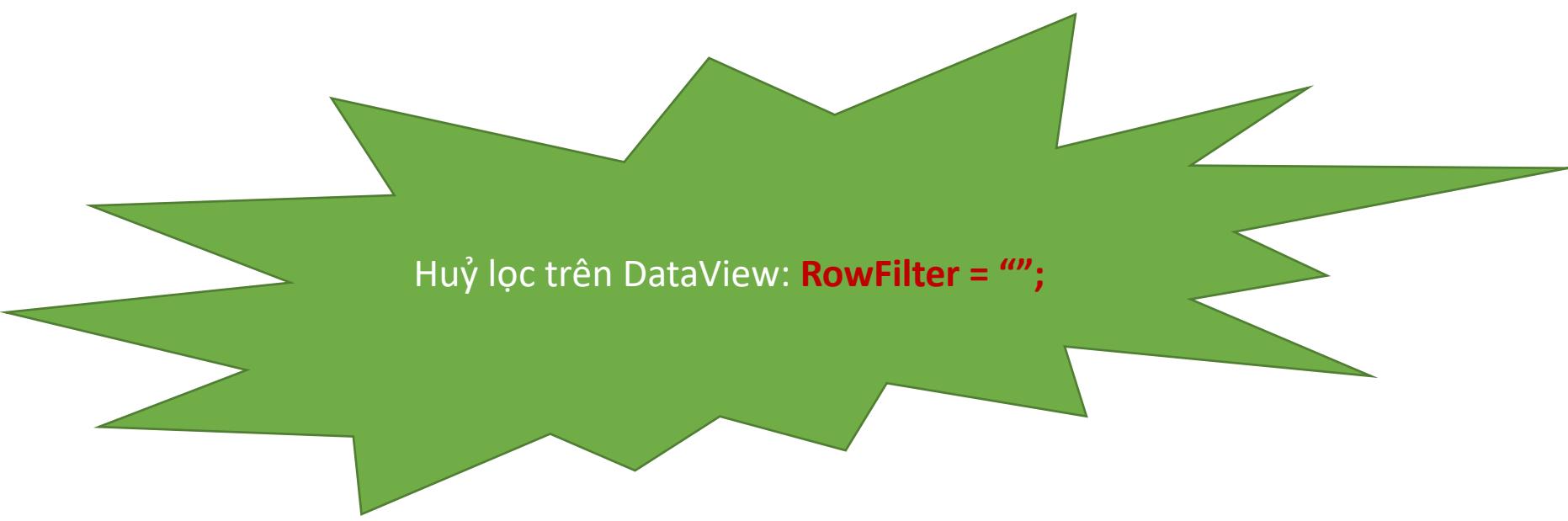
|   | MSSV       | Họ Tên          | NgaySinh   | GioiTinh                            | Diem TB | MaLop |
|---|------------|-----------------|------------|-------------------------------------|---------|-------|
| ▶ | 0306171020 | Hồ Tiên Đạt     | 1/4/1999   | <input checked="" type="checkbox"/> | 7.4     | 1     |
|   | 0306171023 | Phùng Hiếu Đông | 8/18/1999  | <input checked="" type="checkbox"/> | 8.1     | 1     |
|   | 0306171024 | Bùi Văn Đông    | 10/29/1999 | <input checked="" type="checkbox"/> | 6.7     | 1     |
| * | 0306171036 | Trần Công Khanh | 12/10/1999 | <input checked="" type="checkbox"/> | 6.6     | 1     |
|   |            |                 |            | <input type="checkbox"/>            |         |       |

# DATASET

- **DataView**

- Bài tập 4: Tìm kiếm (Lọc) Sinh viên theo Lớp

```
// Lọc dữ liệu trên DataView
dvSinhVien.RowFilter = string.Format("MaLop = {0}", cmbLop.SelectedValue);
```



Huỷ lọc trên DataView: **RowFilter = “”;**

# DATASET

- **DataView**

- Bài tập 5: Sắp xếp danh sách *Họ tên* Sinh viên theo thứ tự tăng (ASC) / giảm (DESC) dần
  - Gợi ý:
    - <đối\_tượng\_DataView>.Sort = “HoTen **ASC**”;
    - <đối\_tượng\_DataView>.Sort = “HoTen **DESC**”;
- Bài tập 6: Sắp xếp danh sách Sinh viên theo thứ tự **Lớp tăng dần**, nếu cùng lớp thì sắp xếp theo **Điểm TB giảm dần**
  - <đối\_tượng\_DataView>.Sort = “MaLop **ASC**, DiemTB **DESC**”;

# DATA BINDING

- **Data Binding:**

- Là một giải pháp cho vấn đề liên kết giữa tập dữ liệu với các controls (*Giúp hiển thị dữ liệu trong Data Source lên control*)
- Phân loại: Có 2 loại Data Binding
  - Simple Data Binding
  - Complex Data Binding

# DATA BINDING

- **Simple Data Binding:**

- Kết nối 1 property của control (*loại property chỉ lưu 1 giá trị tại 1 thời điểm*) với 1 cột (*hay property*) của Data Source
- Ví dụ:
  - `string tenControl.Text`
  - `Image pictureBox.Image`
  - `object tenControl.Tag`
  - `bool checkBox.Checked`
  - `bool radioButton.Checked`
  - `int trackBar.Value`
  - ...

# DATA BINDING

- Simple Data Binding:

- *Binding binding = new Binding("propertyName",  
dataSource, "dataMember", [true]);  
<control>.DataBindings.Add(binding);*

*Hoặc:*

- *<control>.DataBindings.Add(new Binding("propertyName",  
dataSource, "dataMember", [true]));*

# DATA BINDING

- Simple Data Binding:

- Bài tập 7: Hiển thị các thông tin của sinh viên lên các control tương ứng trong GroupBox Thông tin sinh viên  
*(Không xử lý sự kiện Click, ContentClick của DataGridView)*

The screenshot shows a Windows application window titled "Quản Lý Sinh Viên". Inside, there is a GroupBox labeled "Thông tin sinh viên" containing several controls: a text box for "MSSV" (value: 0306171023), a date picker for "Ngày sinh" (value: 18/08/1999), a numeric up-down control for "Điểm TB" (value: 8.1), a text box for "Họ tên" (value: Phùng Hiếu Đông), a radio button group for "Giới tính" (selected: Nam), and a dropdown menu for "Lớp" (value: CĐTH 17A). Below this, another GroupBox is labeled "Danh sách sinh viên:" and contains a DataGridView displaying student data:

|   | MSSV       | Họ Tên           | Ngay Sinh  | Gioi Tinh                           | Diem TB | Ma Lop |
|---|------------|------------------|------------|-------------------------------------|---------|--------|
|   | 0306171020 | Hồ Tiến Đạt      | 1/4/1999   | <input checked="" type="checkbox"/> | 7.4     | 1      |
|   | 0306171021 | Triệu Văn Định   | 4/4/1999   | <input checked="" type="checkbox"/> | 7.8     | 2      |
| ▶ | 0306171023 | Phùng Hiếu Đông  | 8/18/1999  | <input checked="" type="checkbox"/> | 8.1     | 1      |
|   | 0306171024 | Bùi Văn Đồng     | 10/29/1999 | <input checked="" type="checkbox"/> | 6.7     | 1      |
|   | 0306171025 | Mai Thanh Hải    | 2/16/1999  | <input checked="" type="checkbox"/> | 7.5     | 6      |
|   | 0306171027 | Nguyễn Ngọc Hiền | 4/2/1999   | <input type="checkbox"/>            | 7.4     | 6      |

# DATA BINDING

- Simple Data Binding:

- Bài tập 7: Hiển thị các thông tin của sinh viên lên các control tương ứng trong GroupBox Thông tin sinh viên  
*(Không xử lý sự kiện Click, ContentClick của DataGridView)*

```
// Đổ dữ liệu lên DataGridView
dgvSinhVien.DataSource = ds;
dgvSinhVien.DataMember = "SinhVien";

// Simple Data Binding
Binding binding = new Binding("Text", ds, "SinhVien.MSSV");
txtMSSV.DataBindings.Add(binding);

txtHoTen.DataBindings.Add(new Binding("Text", ds, "SinhVien.HoTen"));

dtpNgaySinh.DataBindings.Add(new Binding("Value", ds, "SinhVien.NgaySinh"));

radNam.DataBindings.Add(new Binding("Checked", ds, "SinhVien.GioiTinh"));

nudDiem.DataBindings.Add(new Binding("Value", ds, "SinhVien.DiemTB"));

cmbLop.DataBindings.Add(new Binding("SelectedValue", ds, "SinhVien.MaLop"));
```

# DATA BINDING

- Simple Data Binding:

- Bài tập 7: Hiển thị các thông tin của sinh viên lên các control tương ứng trong GroupBox Thông tin sinh viên  
*(Không xử lý sự kiện Click, ContentClick của DataGridView)*

```
// Đổ dữ liệu lên DataGridView
dgvSinhVien.DataSource = ds.Tables["SinhVien"];

// Simple Data Binding
Binding binding = new Binding("Text", ds.Tables["SinhVien"], "MSSV");
txtMSSV.DataBindings.Add(binding);

txtHoTen.DataBindings.Add(new Binding("Text", ds.Tables["SinhVien"], "HoTen"));

dtpNgaySinh.DataBindings.Add(new Binding("Value", ds.Tables["SinhVien"], "NgaySinh"));

radNam.DataBindings.Add(new Binding("Checked", ds.Tables["SinhVien"], "GioiTinh"));

nudDiem.DataBindings.Add(new Binding("Value", ds.Tables["SinhVien"], "DiemTB"));

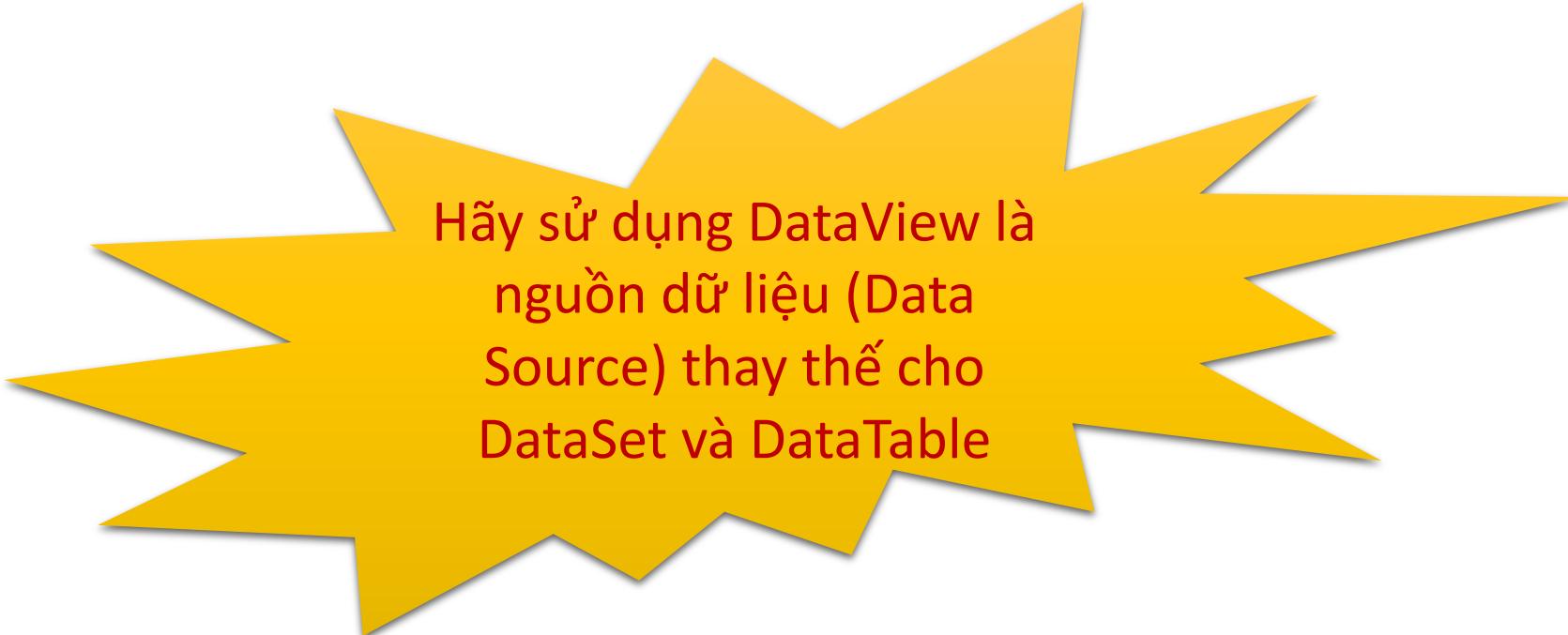
cmbLop.DataBindings.Add(new Binding("SelectedValue", ds.Tables["SinhVien"], "MaLop"));
```

Hoặc

# DATA BINDING

- **Simple Data Binding:**

- Bài tập 7: Hiển thị các thông tin của sinh viên lên các control tương ứng trong GroupBox Thông tin sinh viên  
*(Không xử lý sự kiện Click, ContentClick của DataGridView)*



Hãy sử dụng DataView là  
nguồn dữ liệu (Data  
Source) thay thế cho  
DataSet và DataTable

# DATA BINDING

- **Complex Data Binding:**

- Kết nối 1 control có khả năng hiển thị nhiều giá trị tại 1 thời điểm với 1 cột hay tất cả các cột trong data source

- Ví dụ:

- DataGridView
- ComboBox
- ListBox
- ...

# DATA BINDING

- Complex Data Binding:

- *<control>.DataSource = <đối\_tượng\_DataTable>;*

*Hoặc:*

- *<control>.DataSource = <đối\_tượng\_DataSet>;  
<control>.DataMember = “tên DataTable”;*

```
DataSet ds = new DataSet();
SqlDataAdapter daSinhVien = new SqlDataAdapter("SELECT * FROM SINH_VIEN", Program.ChuoiKetNoi);

// Đổ dữ liệu vào DataTable SinhVien trong DataSet ds
daSinhVien.Fill(ds, "SinhVien");

// Đổ dữ liệu lên DataGridView
dgvSinhVien.DataSource = ds.Tables["SinhVien"];
```

```
// Đổ dữ liệu lên DataGridView
dgvSinhVien.DataSource = ds;
dgvSinhVien.DataMember = "SinhVien";
```

# MASTER - DETAIL

- Dạng quan hệ **1-N** trong cơ sở dữ liệu
- Các bước thực hiện:

- **Bước 1:** fill dữ liệu

Fill 2 bảng vào dataset (*bao gồm khóa chính và khóa ngoại*)

- **Bước 2:** tạo quan hệ

```
DataTable colMaster = ds.Tables["tenMaster"].Columns["cộtPK"];
DataTable colDetail = ds.Tables["tenDetail"].Columns["cộtFK"];
DataRelation relation = new DataRelation("tenquanhe", colMaster,
colDetail);
ds.Relations.Add(relation);
```

# MASTER - DETAIL

- Các bước thực hiện:

- **Bước 3:** (Cách 1)

```
dataGridViewMaster.DataSource = ds;
```

```
dataGridViewMaster.DataMember = "tenMaster";
```

```
dataGridViewDetail.DataSource = ds;
```

```
dataGridViewDetail.DataMember = "tenMaster.tenquanhe";
```

# MASTER - DETAIL

- Các bước thực hiện:

- **Bước 3:** (Cách 2)

```
dataGridViewMaster.DataSource = ds.Tables["tenMaster"];
```

```
dataGridViewDetail.DataSource = ds.Tables["tenMaster"];
```

```
dataGridViewDetailDataMember = "tenquanhe";
```

# MASTER - DETAIL

- Bài tập 8: hiển thị danh sách Sinh viên theo Lớp

The screenshot shows a Windows application window titled "Quản Lý Sinh Viên". The window contains two tables:

**Danh sách lớp:**

|   | MaLop | TenLop   |
|---|-------|----------|
| ▶ | 1     | CĐTH 17A |
|   | 2     | CĐTH 17B |
|   | 6     | CĐTH 17C |

**Danh sách sinh viên:**

|   | MaLop | MSSV       | HoTen           | NgaySinh   | GioiTinh                            | Diem TB |
|---|-------|------------|-----------------|------------|-------------------------------------|---------|
| ▶ | 1     | 0306171020 | Hồ Tiên Đạt     | 1/4/1999   | <input checked="" type="checkbox"/> | 7.4     |
|   | 1     | 0306171023 | Phùng Hiếu Đông | 8/18/1999  | <input checked="" type="checkbox"/> | 8.1     |
|   | 1     | 0306171024 | Bùi Văn Đồng    | 10/29/1999 | <input checked="" type="checkbox"/> | 6.7     |
|   | 1     | 0306171036 | Trần Công Khanh | 12/10/1999 | <input checked="" type="checkbox"/> | 6.6     |
| * |       |            |                 |            | <input type="checkbox"/>            |         |

# MASTER - DETAIL

- Bài tập 8: hiển thị danh sách Sinh viên theo Lớp

```
// Bước 1: Đổ dữ liệu vào DataTable SinhVien trong DataSet ds
ds = new DataSet();
daSinhVien = new SqlDataAdapter("SELECT * FROM SINH_VIEN", Program.ChuoiKetNoi);
daLop = new SqlDataAdapter("SELECT * FROM LOP", Program.ChuoiKetNoi);

daSinhVien.Fill(ds, "SinhVien");
daLop.Fill(ds, "Lop");

// Bước 2: tạo quan hệ
ds.Relations.Add(new DataRelation("SinhVien_Lop", ds.Tables["Lop"].Columns["MaLop"],
 ds.Tables["SinhVien"].Columns["MaLop"]));

// Bước 3: Đổ dữ liệu lên DataGridView
dgvLop.DataSource = ds.Tables["Lop"];
dgvSinhVien.DataSource = ds.Tables["Lop"];
dgvSinhVien.DataMember = "SinhVien_Lop";
```

```
// Bước 3: Đổ dữ liệu lên DataGridView
dgvLop.DataSource = ds;
dgvLop.DataMember = "Lop";

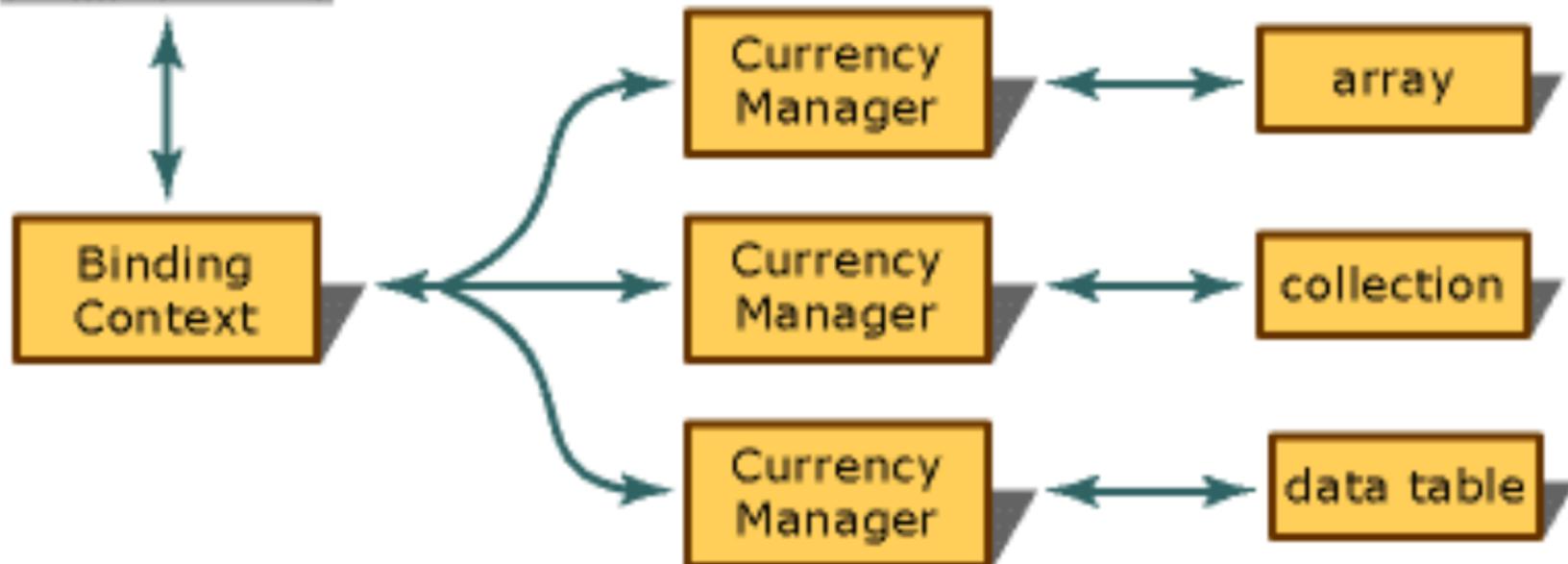
dgvSinhVien.DataSource = ds;
dgvSinhVien.DataMember = "Lop.SinhVien_Lop";
```

# CURRENCYMANAGER

- **Đồng bộ dữ liệu** là khả năng cập nhật tức thì dữ liệu giữa các control hay giữa các control và Data source.
- Khi kết nối các control với một Data source, chương trình tự động tạo ra một đối tượng thuộc lớp **CurrencyManager** để theo dõi vị trí của record trong Data source
- Mỗi data source tương ứng có 1 đối tượng CurrencyManager
- Windows Form lưu tất cả đối tượng CurrencyManager vào trong mảng **BindingContext**

# CURRENCYMANAGER

## Windows Form



# CURRENCYMANAGER

- Đối tượng CurrencyManager

*CurrencyManager cm =*

*(CurrencyManager)this.BindingContext[tenDataSet, "Bảng"];*

*Hoặc*

*CurrencyManager cm = (CurrencyManager)this.BindingContext[tenBang];*

- Một số thuộc tính:

- *int Position*
  - *int Count*

- Một số phương thức:

- *void AddNew()*
  - *void EndCurrentEdit()*
  - *void CancelCurrentEdit()*
  - *void RemoveAt(int index)*

# CURRENCYMANAGER

- Chú ý quan trọng:
  - Data Source và Data Member của các control phải đồng nhất với nhau
    - *Hoặc cùng Dataset*
    - *Hoặc cùng DataTable*
  - **Tốt nhất: nên dùng DataTable**

# CURRENCYMANAGER

Thông tin sinh viên

|                              |                                                                                            |                                                                             |
|------------------------------|--------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| MSSV: <input type="text"/>   | Ngày sinh: <input type="date" value="07/04/2019"/> <input type="button" value="Calendar"/> | Điểm TB: <input type="text" value="0.0"/> <input type="button" value="Up"/> |
| Họ tên: <input type="text"/> | Giới tính: <input checked="" type="radio"/> Nam <input type="radio"/> Nữ                   | Lớp: <input type="text"/>                                                   |

Next  Previous  First  Last  Insert  Update  Delete  Cancel

```
if (cm.Position > 0)
 cm.Position--;
```

```
if (cm.Position < cm.Count - 1)
 cm.Position++;
```

```
cm.Position = 0;
```

```
cm.Position = cm.Count - 1;
```

```
cm.AddNew();
```

```
cm.RemoveAt(cm.Position);
```

```
cm.CancelCurrentEdit();
```

```
cm.EndCurrentEdit();
daSinhVien.Update(ds.Tables["SinhVien"]);
```

# BINDING SOURCE

- **Binding Source** là đối tượng trung gian được sử dụng để kết nối giữa Control và Data Source.
- Ưu điểm khi sử dụng Binding Source
  - *Dễ dàng thay đổi Data Source của các Control*
  - *Cung cấp nhiều sự kiện và các phương thức*

# BINDING SOURCE

Quản Lý Sinh Viên

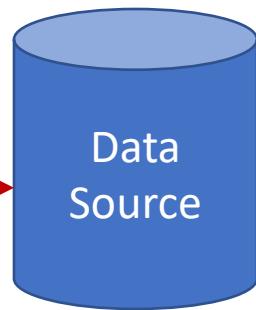
Thông tin sinh viên

|         |                 |            |                                                               |          |          |
|---------|-----------------|------------|---------------------------------------------------------------|----------|----------|
| MSSV:   | 0306171023      | Ngày sinh: | 18/08/1999                                                    | Điểm TB: | 8.1      |
| Họ tên: | Phùng Hiếu Đông | Giới tính: | <input checked="" type="radio"/> Nam <input type="radio"/> Nữ | Lớp:     | CĐTH 17A |

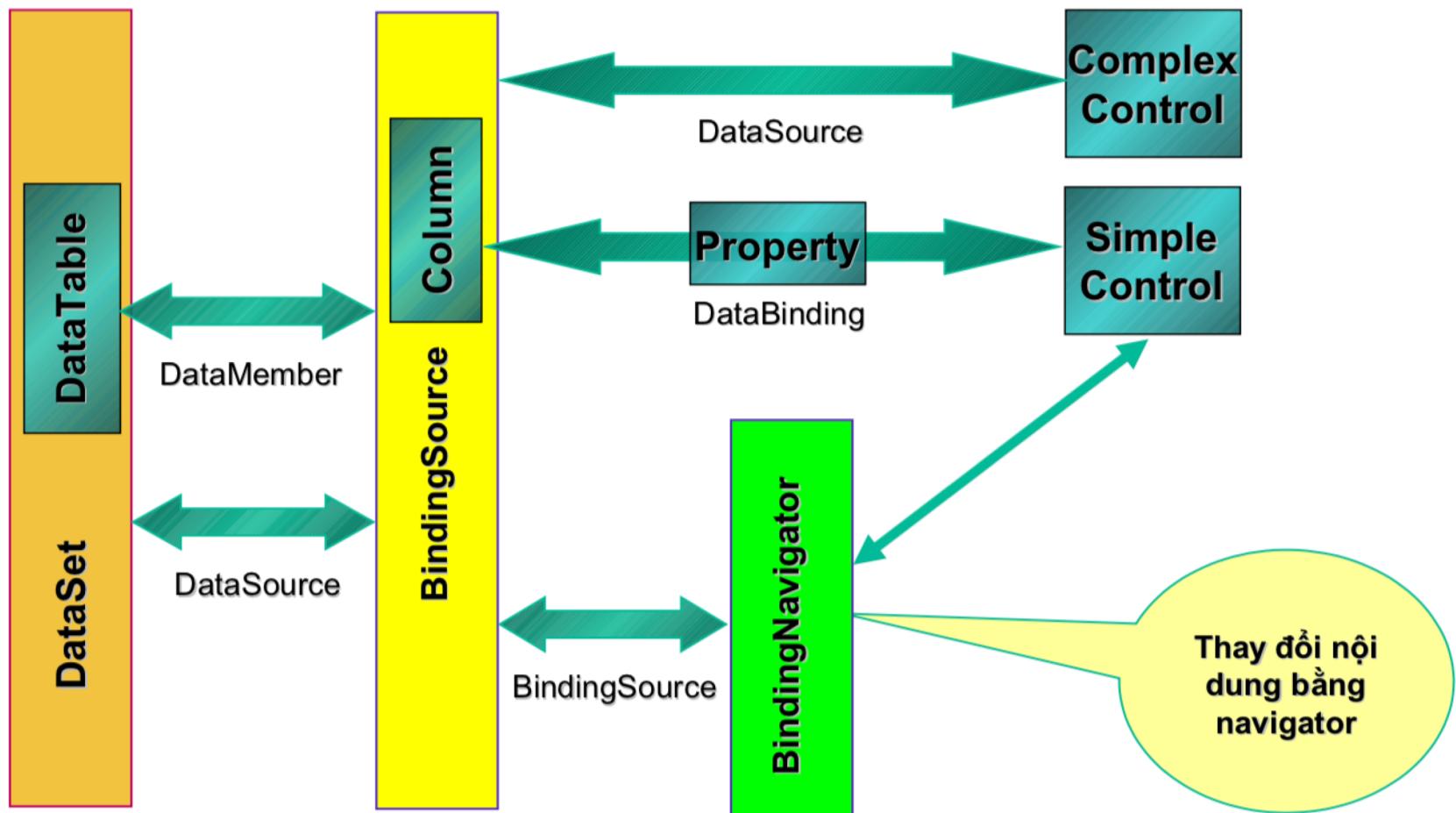
Danh sách sinh viên:

| MSSV       | HoTen            | NgaySinh   | GioiTinh                            | DiemTB | MaLop |
|------------|------------------|------------|-------------------------------------|--------|-------|
| 0306171020 | Hồ Tiên Đạt      | 1/4/1999   | <input checked="" type="checkbox"/> | 7.4    | 1     |
| 0306171021 | Triệu Văn Định   | 4/4/1999   | <input checked="" type="checkbox"/> | 7.8    | 2     |
| 0306171023 | Phùng Hiếu Đông  | 8/18/1999  | <input checked="" type="checkbox"/> | 8.1    | 1     |
| 0306171024 | Bùi Văn Đồng     | 10/29/1999 | <input checked="" type="checkbox"/> | 6.7    | 1     |
| 0306171025 | Mai Thanh Hải    | 2/16/1999  | <input checked="" type="checkbox"/> | 7.5    | 6     |
| 0306171027 | Nguyễn Ngọc Hiển | 4/2/1999   | <input type="checkbox"/>            | 7.4    | 6     |

Binding  
Source



# BINDING SOURCE



# BINDING SOURCE

- Tạo đối tượng Binding Source:

*BindingSource* bs = new *BindingSource*();

...

*bs.DataSource* = dataSource;  
*bs.DataMember* = "dataMember";

Hoặc

*BindingSource* bs = new *BindingSource*(dataSource,  
"dataMember");

# BINDING SOURCE

- Một số thuộc tính:
  - `int bs.Position`
  - `int bs.Count`
  - `CurrencyManager bs.CurrencyManager`
  - `string bs.Filter`
  - `string bs.Sort`
- Một số phương thức:
  - `void bs.ResetBinding(bool metaDataChanged);`
  - `void bs.ResetCurrentItem();`
  - `void bs.ResetItem(int index);`

# BINDING SOURCE

- Kết nối control với Binding Source
  - Simple Data Binding
    - *tenControl.DataBindings.Add("propertyName", bs, "dataMember", [true]);*
  - Complex Data Binding
    - *tenControl.DataSource = bs;*
    - *tenControl.DataMember = "tenBang";*

# BINDING SOURCE

Thông tin sinh viên

|                      |                                                               |                                                            |
|----------------------|---------------------------------------------------------------|------------------------------------------------------------|
| MSSV:                | Ngày sinh:                                                    | Điểm TB:                                                   |
| <input type="text"/> | 07/04/2019 <input type="button" value="Calendar"/>            | <input type="text"/> 0.0 <input type="button" value="Up"/> |
| Họ tên:              | Giới tính:                                                    | Lớp:                                                       |
| <input type="text"/> | <input checked="" type="radio"/> Nam <input type="radio"/> Nữ | <input type="text"/>                                       |

Next  Previous  First  Last  Insert  Update  Delete  Cancel

Red arrows point from each button to its corresponding method call:

- Next → `bs.MoveNext();`
- Previous → `bs.MovePrevious();`
- First → `bs.MoveFirst();`
- Last → `bs.MoveLast();`
- Insert → `bs.AddNew();` (highlighted with a red oval)
- Delete → `bs.RemoveCurrent();`
- Cancel → `bs.CancelEdit();`

```
bs = new BindingSource(ds, "SinhVien");

// Simple Data Binding
Binding binding = new Binding("Text", bs, "MSSV");
txtMSSV.DataBindings.Add(binding);

txtHoTen.DataBindings.Add(new Binding("Text", bs, "HoTen"));

dtpNgaySinh.DataBindings.Add(new Binding("Value", bs, "NgaySinh"));

radNam.DataBindings.Add(new Binding("Checked", bs, "GioiTinh"));

nudDiem.DataBindings.Add(new Binding("Value", bs, "DiemTB"));

cmbLop.DataBindings.Add(new Binding("SelectedValue", bs, "MaLop"));
```

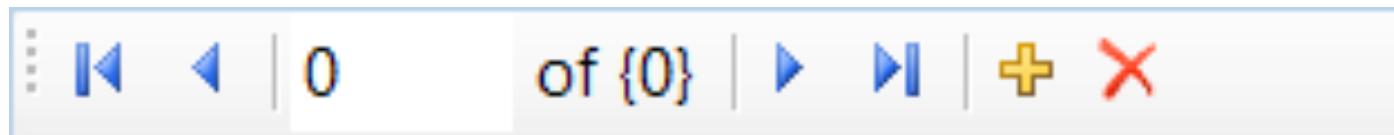
Red box highlights the code block for simple data binding:

```
bs = new BindingSource(ds, "SinhVien");

// Simple Data Binding
Binding binding = new Binding("Text", bs, "MSSV");
txtMSSV.DataBindings.Add(binding);
```

# BINDING NAVIGATOR

- **Binding Navigator** bao gồm các chức năng:
  - MoveNext
  - MovePrevious
  - MoveFirst
  - MoveLast
  - AddNew
  - RemoveCurrent
  - ...



# BINDING NAVIGATOR

- Các bước tạo Binding Navigator
  - Bước 1: Kéo Binding Navigator vào Form
  - Bước 2: Thiết lập BinddingSource  
`tenNavigator.BindingSource=tenBindingSource;`

