





Objects Oriented Programming

C# FOR BASE

GV: Bùi Quang Đăng

Contents

-  1 Inheritance
-  2 Polymorphism, Virtual & Abstract
-  3 Interface
-  4 Exercises

C#.NET for Base

Inheritance

C#.NET for Base

❖ Inheritance of Objects Class

Class Parent

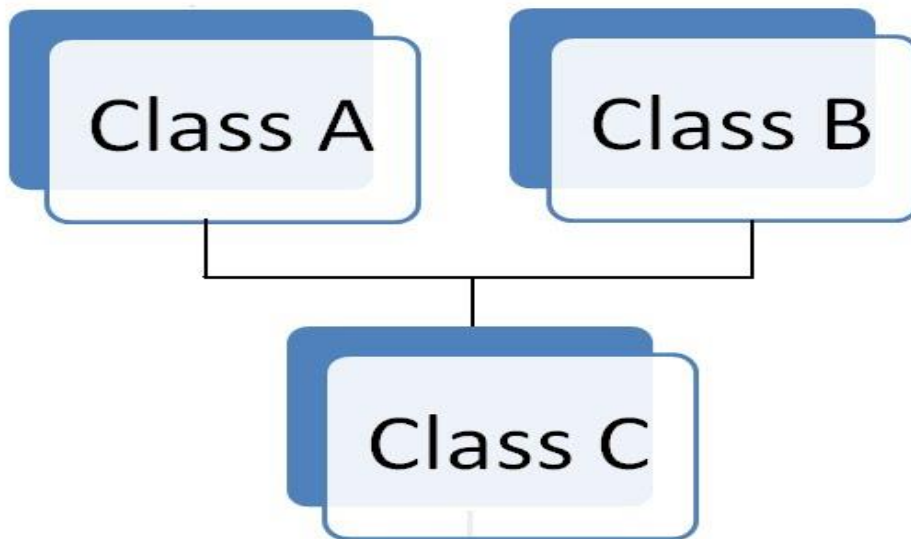
Inheritance

Class Child

Một lớp có thể thừa hưởng những thuộc tính và phương thức từ một lớp khác

C#.NET for Base

❖ Inheritance of Objects Class



```
class Person  
{  
    ...  
}
```

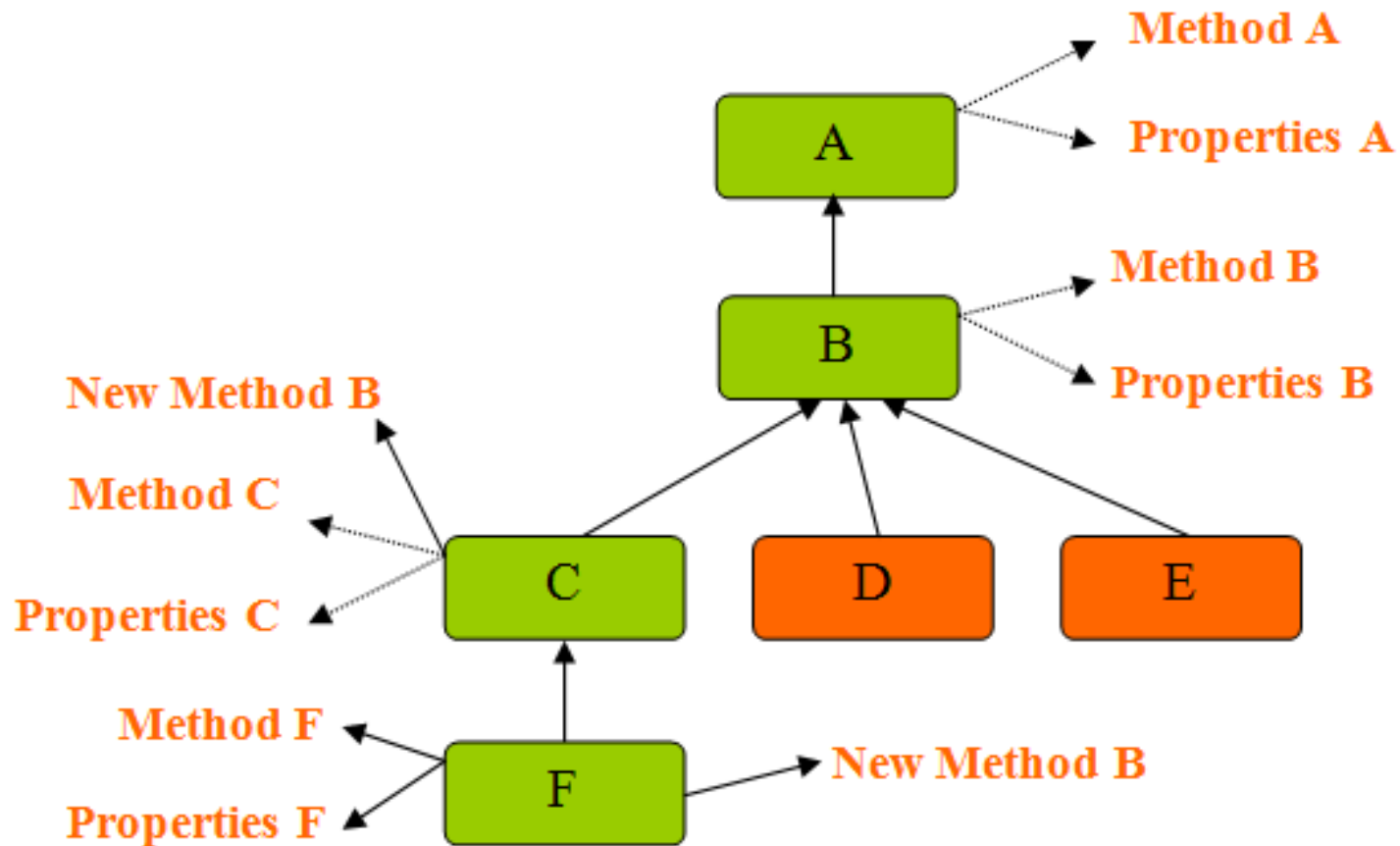
```
class Employee : Person  
{  
    ...  
}
```

inherits from
all ancestors

```
class Faculty : Employee  
{  
    ...  
}
```

C#.NET for Base

❖ Inheritance of Objects Class

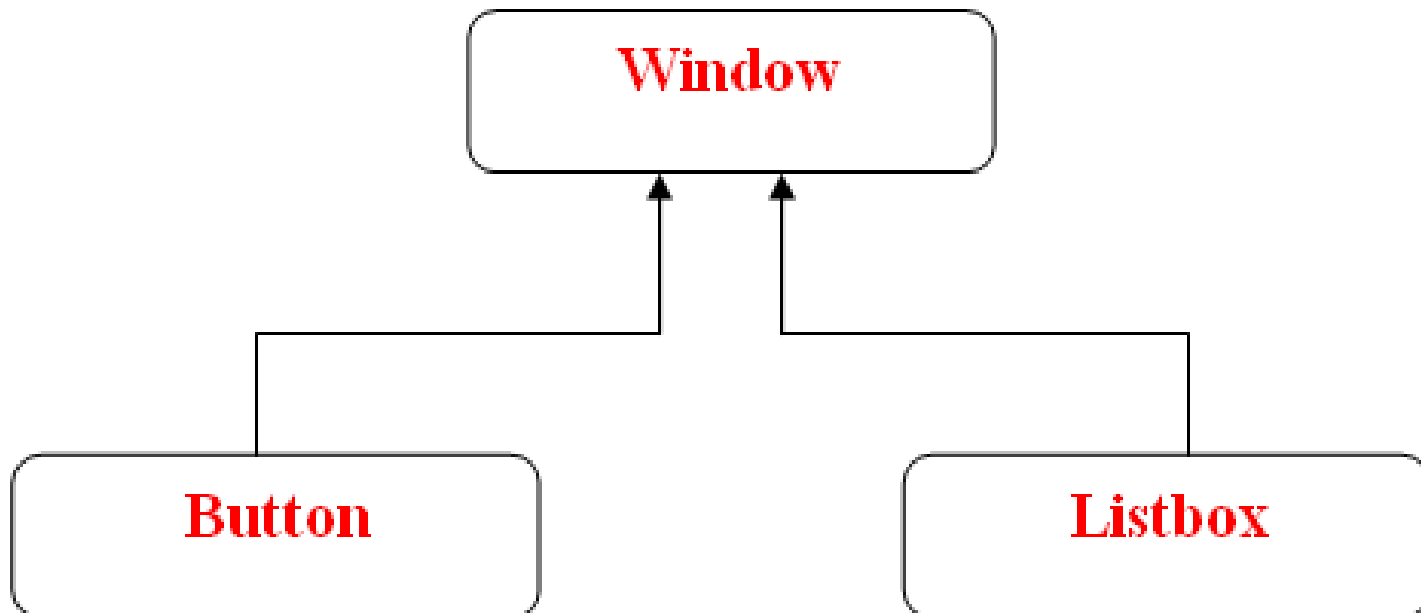


C#.NET for Base

❖ Inheritance of Objects Class

- 1. Thực thi kế thừa :

Cú pháp: <**Tên lớp dẫn xuất**> : <**Tên lớp cơ sở**>



C#.NET for Base

❖ Inheritance of Objects Class

- 2. Truyền tham số cho construction lớp cơ sở :

```
public ContructionBase (int A, int B, int C) : base (A, B)
{
    this.C = C;
}
```

- Gọi method của lớp cơ sở :

```
//call method of objects class
base.MethodA();
```


C#.NET for Base

❖ Inheritance of Objects Class

- Ví dụ:

```
class DataProvider
{
    private static DataAccess _provider = null;

    public DataProvider() ...

    private void CreateProvider() ...

    public static DataAccess Instance()
    {
        return _provider;
    }

    //Viết các hàm lấy dữ liệu của các bảng ở đây
}
```

C#.NET for Base

❖ Inheritance of Objects Class

- Ví dụ:

```
class SqlDataProvider : DataProvider
{
    #region SinhVien
    public DataTable GetListSinhVien()...

    public int SinhVienAdd(SinhVien objSinhVien)...

    public int SinhVienUpdate(SinhVien objSinhVien)...

    public bool SinhVienDelete(int SinhVienId)...

    public SinhVien GetSinhVienById(int SinhVienid)...

    #endregion SinhVien

    Chuyen khoa
}
```

C#.NET for Base

Polymorphism

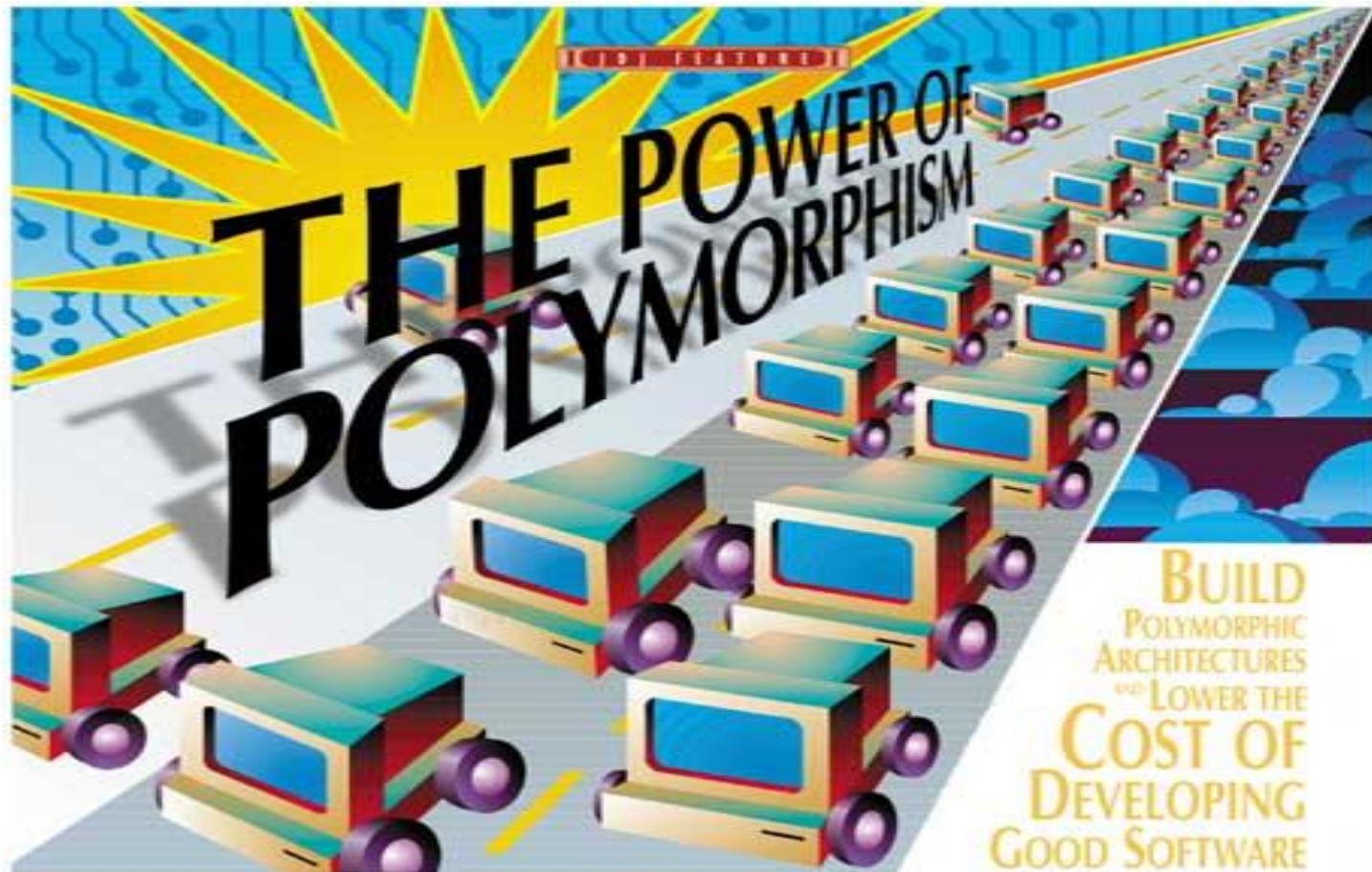
C#.NET for Base

❖ Polymorphism



C#.NET for Base

❖ Polymorphism



C#.NET for Base

❖ Polymorphism

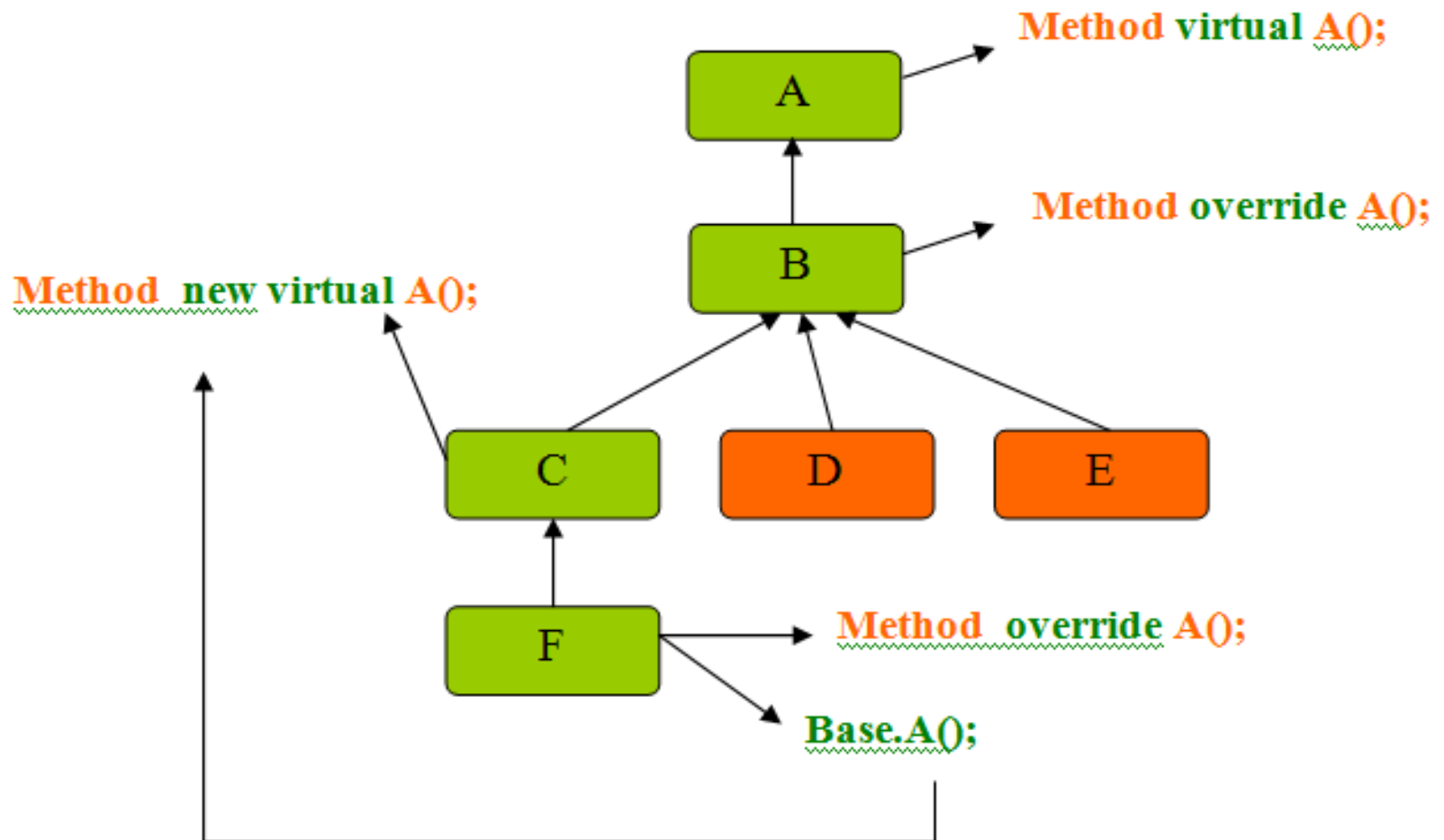
- Là khả năng cho phép một hành vi (**method**) có thể được thể hiện dưới nhiều hình thức khác nhau mà không quan tâm tới chi tiết.
- Để tạo một method hỗ trợ tính đa hình khai báo với từ khoá Virtual trong phương thức của lớp cơ sở.

Ví dụ: `public virtual void UsePolymorphizm();`

- Các lớp dẫn xuất được tự do thực thi theo cách của riêng mình bằng cách sử dụng từ khoá Override để chồng lên phương thức ảo của lớp cơ sở.

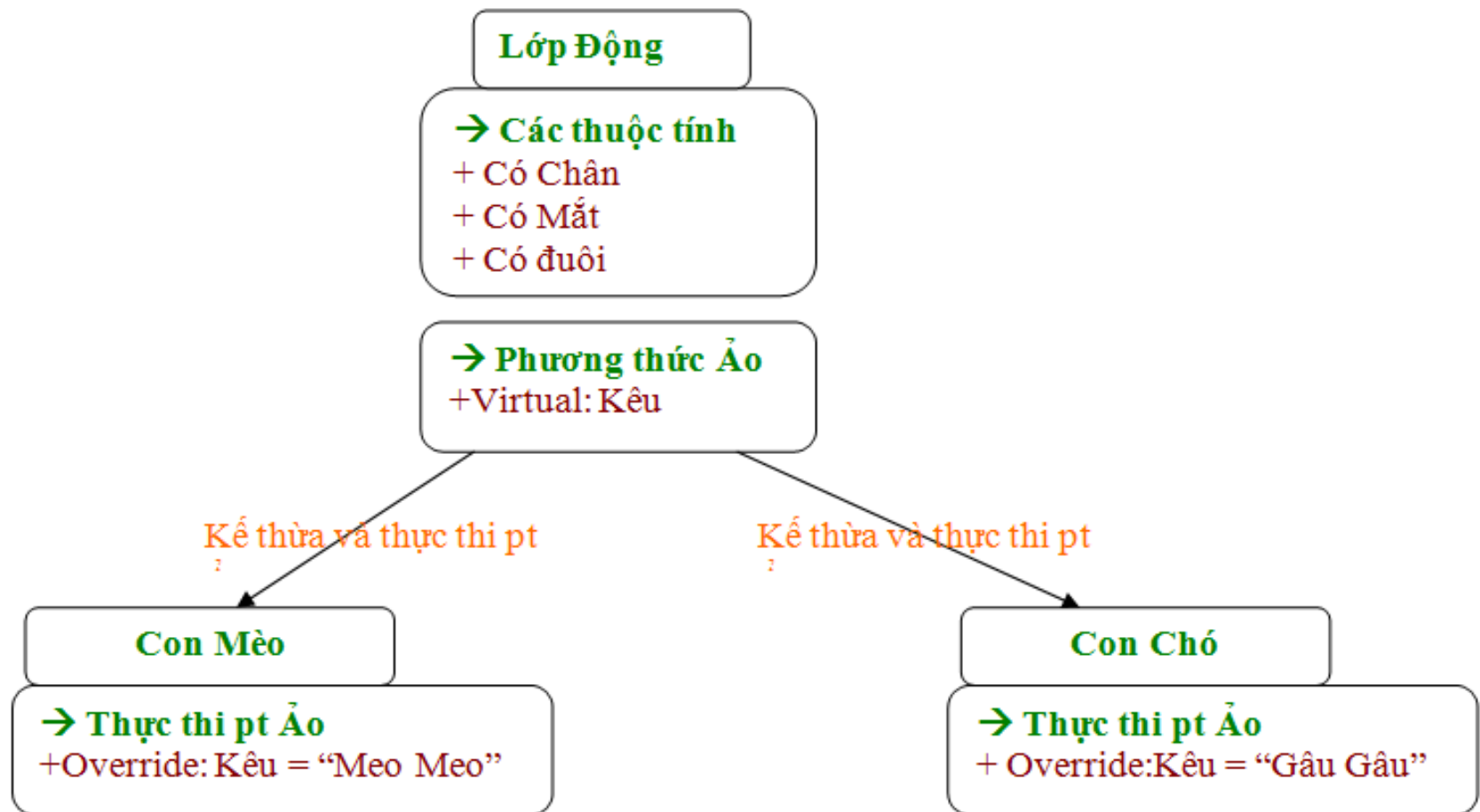
C#.NET for Base

❖ Polymorphism



C#.NET for Base

❖ Polymorphism of method



C#.NET for Base

❖ Polymorphism of method

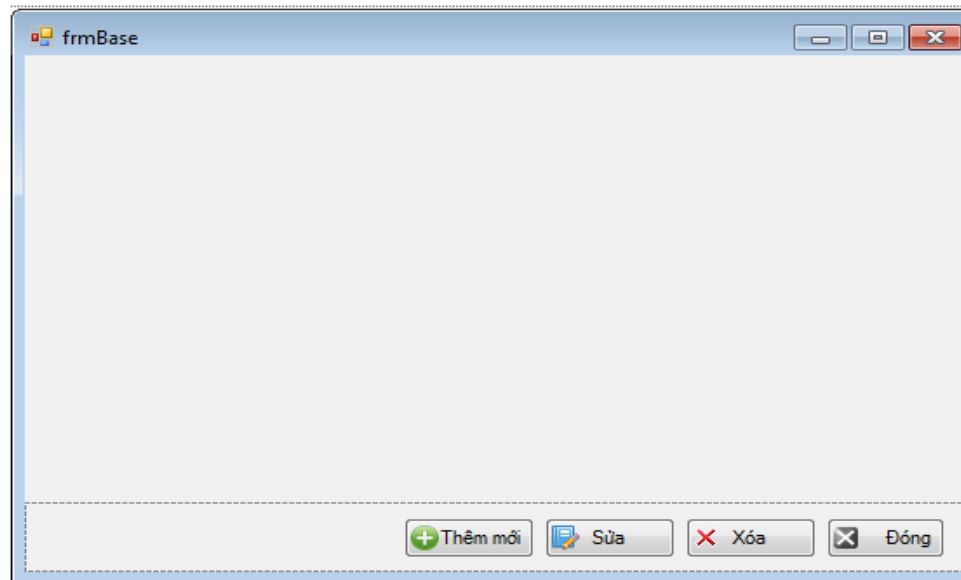
■ Ví dụ:

- Khi phát triển một ứng dụng thực tế người ta có thể sử dụng kế thừa và đa hình bằng việc tạo một lớp dạng form cơ sở có các tính năng như thêm mới, sửa, xóa,...
- Trong đó khai báo các phương thức ảo tương ứng với nó. Sau đó thực hiện các form chức năng khác kế thừa từ form base này và sử dụng đa hình cho các phương thức ảo đó thông qua từ khóa virtual, abstract và override

C#.NET for Base

❖ Polymorphism of method

- Ví dụ:



```
public virtual void ThemMoiThongTin() ...
```

```
public virtual void SuaThongTin() ...
```

```
public virtual void XoaThongTin() ...
```

C#.NET for Base

❖ Polymorphism of method

■ Ví dụ:

Quản lý sinh viên - [frmDanhSachSinhVien]

Hệ thống | Danh mục | Nghiệp vụ | Báo cáo | Trợ giúp

Đăng xuất | Danh sách sinh viên | Tra cứu điểm thi

Thông tin chuyên khoa

Công nghệ thông tin
Điện tử
Kinh tế

Thông tin tìm kiếm sinh viên

Mã SV: Họ tên:

	ID	Mã SV	Họ Tên	Giới Tính	Ngày Sinh	Điện Thoại	Email	Địa Chỉ
▶	1	TLA0001	Tiến Mạnh	False	12/01/1991	09867788789	tienmanh@yahoo.com	HN
	2	TLA0002	Trần Thanh Phương	False	01/01/1983	456789		Hà Nội
	3	TLA0003	Bùi Thị Trang	True	01/01/1990			Nam Định
	15	TLA0005	Nguyễn Tiến Dũng	False	12/01/1991	09867788789	tienmanh@yahoo.com	HN
	28	BKA0002	Nguyễn Hải Hà	False	12/12/1994	Hà Nội	nguyenhue86hn@gmail.com	Hà Nội
	33	DCN0012	Nguyễn Thành Công	False	27/07/1990	09678678	thanhcong@gmail.com	
	38	BKA00123	Lưu Nhật Nam			098436546555		
	42	BKA0015	Nguyễn Mạnh Hà				hanm@gmail.com	HN
*								

Thêm mới Sửa Xóa Đóng

Status

C#.NET for Base

❖ Polymorphism of method

- Ví dụ:

```
public partial class frmDanhSachSinhVien : Base.frmBase
{
    public frmDanhSachSinhVien()
    {
        InitializeComponent();
    }

    public override void ThemMoiThongTin()...

    public override void SuaThongTin()...

    public override void XoaThongTin()...

}
}
```

C#.NET for Base

❖ Abstract of Objects Class

- Sử dụng từ khóa **abstract** để tạo các phương thức ảo nhưng có những đặc điểm khác khi dùng virtual như sau:
 - Lớp chứa các phương thức sử dụng abstract phải là lớp abstract
 - Phương thức sử dụng abstract được khai báo định danh, không viết chi tiết nội dung xử lý trong hàm.
 - Lớp kế thừa bắt buộc phải khai báo các phương thức trong lớp cơ sở sử dụng từ khóa abstract.

C#.NET for Base

❖ Abstract of Objects Class

Ví dụ:

```
abstract public class ClassAbstract
{
    abstract protected void MethodA();

    abstract protected int PropertiesA {
        get;
        set;
    }
}

public class ClassDelivery : ClassAbstract
{
    int i;
    protected override void MethodA() {
        //Code here
    }

    protected override int PropertiesA {
        get { return i; }
        set { i = value; }
    }
}
```

C#.NET for Base

❖ Abstract of Objects Class

Ví dụ:

```
namespace QLSinhVien.Base
{
    abstract class Base
    {
        //Chỉ khai báo định danh các phương thức
        public abstract void ThemMoiThongTin();
        public abstract void SuaThongTin();
    }
}
```

C#.NET for Base

❖ Abstract of Objects Class

Ví dụ:

```
namespace QLSinhVien.Base
{
    class Parent:Base
    {
        public override void ThemMoiThongTin()
        {
            //Viết công việc xử lý thêm mới
        }
        public override void SuaThongTin()
        {
            //Viết công việc xử lý sửa
        }
    }
}
```


C#.NET for Base

❖ Virtual & Abstract Properties

Cú pháp:

```
public abstract[or virtual] string PropertiesName  
{  
    get;  
    set;  
}
```

ABSTRACT

C#.NET for Base

❖ Sealed of Objects Class

- Là một lớp cô lập không cho phép các lớp khác kế thừa từ các lớp sử dụng từ khóa sealed.

Cú pháp:

```
sealed [public,internal] class Class1
{
    //Code here
}
```

C#.NET for Base

❖ Sealed of Objects Class

- Ví dụ: Lớp **FileInfo** cho phép người lập trình làm việc với file trong namespace **System.IO**

```
namespace System.IO
{
    ...public sealed class FileInfo : FileSystemInfo
    {
        ...public FileInfo(string fileName);

        ...public DirectoryInfo Directory { get; }
        ...public string DirectoryName { get; }
        ...public override bool Exists { get; }
        ...public bool IsReadOnly { get; set; }
        ...public long Length { get; }
```

C#.NET for Base

Interface

C#.NET for Base

❖ Interface

- ✓ Là một lớp giao diện quy định các phương thức sẽ được sử dụng ở lớp thực hiện (**implement**)
- ✓ Nếu không khai báo đầy đủ các phương thức trong lớp thực hiện sẽ báo lỗi
- ✓ Sử dụng từ khóa **interface** để khai báo lớp giao diện, **các phương thức chỉ thực hiện khai báo định danh, không có nội dung xử lý.**
- ✓ Các phương thức đều sử dụng **public**

C#.NET for Base

❖ Interface

✓ Cú pháp:

```
interface Ten_lop_giao_dien
```

```
{
```

```
    //Khai báo định danh các phương thức
```

```
}
```

```
class Ten_lop_su_dung : Ten_lop_giao_dien
```

```
{
```

```
    //Khai báo và viết chi tiết nội dung xử lý của các phương thức
```

```
}
```

C#.NET for Base

❖ Interface

✓ Ví dụ:

```
interface IBusiness<T, tp>
{
    DataTable GetDanhSach()

    T GetById(tp id);

    bool Save(T obj);

    bool Add(T obj);

    bool Delete(tp id);
}
```

```
class SinhvienController : ISinhvienBusiness
{
    SqlDataProvider _provider = null;

    public SinhvienController()
    {
        _provider = new SqlDataProvider();
    }

    public DataTable GetDanhSach()
    {
        return _provider.GetListSinhVien();
    }

    public bool Delete(int id)
    {
        return _provider.SinhVienDelete(id);
    }
}
```

Thank You !

Stanford – Dạy kinh nghiệm lập trình