

## MÔN : LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

### Bài thực hành số 4.1 : Viết chương trình khảo sát việc định nghĩa & sử dụng đối tượng

#### I. Mục tiêu :

- Giúp SV làm quen với việc đặc tả interface & class của đối tượng.
- Giúp SV hiểu rõ chi tiết về các mức độ bao đóng thành phần trong đối tượng.
- Giúp SV hiểu rõ cách thức sử dụng đối tượng.

#### II. Nội dung :

- Đặc tả 1 interface và 1 class hiện thực interface này.
- Xây dựng chương trình nhỏ demo các tình huống dùng đối tượng khác nhau.

#### III. Chuẩn đầu ra :

- Sinh viên nắm vững và viết thành thạo các lệnh interface và class để định nghĩa 2 góc nhìn khác nhau của từng đối tượng được dùng trong phần mềm.
- Sinh viên nắm vững các mức độ bao đóng thành phần trong đối tượng, từ đó chủ động kiểm soát việc dùng đối tượng trong các tình huống sử dụng khác nhau.

#### IV. Qui trình :

1. Chạy VS .Net, chọn menu File.New.Project để hiển thị cửa sổ New Project.
2. Mở rộng mục Visual C# trong TreeView "Project Types", chọn mục Windows, chọn icon "Console Application" trong listbox "Templates" bên phải, thiết lập thư mục chứa Project trong listbox "Location", nhập tên Project vào textbox "Name:" (td. DefAndUseObject), click button OK để tạo Project theo các thông số đã khai báo.
3. Ngay sau khi Project vừa được tạo ra, cửa sổ soạn code cho chương trình được hiển thị. Hiệu chỉnh code của file Program.cs để có nội dung như sau :

```
namespace DefAndUseObject {
    class Program {
        //định nghĩa 1 hàm dịch vụ
        static void button2_Click(object sender, System.EventArgs e) {
            Console.WriteLine("Ham xu ly Click cua class Program chay");
        }

        //điểm nhập của chương trình
        static void Main(string[] args) {
            //1. tạo đối tượng thuộc class MyClass
            MyClass obj = new MyClass();
            //2. thử truy xuất các thành phần private và protected
            //quan sát phản ứng của chương trình dịch
            obj.m_x = 5;
            obj.arr[1] = 10;
            //3. truy xuất thuộc tính giao tiếp
            obj.x = 5;
            //4. gọi thông điệp chạy tác vụ chức năng
            obj.func2();
            //5. tạo sự kiện Click để xem hàm nào chạy
            Object sender = new Object();
            EventArgs e = new EventArgs();
            obj.OnClick(sender,e);
            //6. thiết lập lại hàm xử lý sự kiện Click
            obj.Click -= new MyClass.EventHandler(obj.button1_Click);
```

```

obj.Click += new MyClass.EventHandler(button2_Click);
//7. tạo sự kiện Click để xem hàm nào chạy
obj.OnClick(sender, e);
//8. dùng đối tượng như danh sách các số nguyên
int i;
//chứa 10 số nguyên từ 0-9 vào danh sách
for (i = 0; i < 10; i++) obj[i] = i;
//đọc lại các số nguyên trong danh sách và hiển thị
for (i = 0; i < 10; i++)
    Console.WriteLine("phan tu " + i + " = " + obj[i]);
//9.chờ người dùng ấn phím để đóng cửa sổ
Console.Write("Hay an phim Enter de dung chuong trinh : ");
Console.Read();
}
} //kết thúc class
} //kết thúc namespace

```

4. Dời chuột về phần tử gốc của cây Project trong cửa sổ “Solution Explorer”, ấn phải chuột vào nó để hiển thị menu lệnh, chọn chức năng Add.New Item để hiển thị cửa sổ “Add New Item”, chọn mục “Interface”, hiệu chỉnh tên interface là IMyInterface.cs, chọn button Add để máy tạo 1 interface mới.

5. Cửa sổ soạn code cho interface IMyInterface được hiển thị, ta định nghĩa interface đơn giản như sau :

```

namespace DefAndUseObject {
    interface IMyInterface {
        void func2(); //chỉ có 1 tác vụ chức năng
    }
}

```

6. Dời chuột về phần tử gốc của cây Project trong cửa sổ “Solution Explorer”, ấn phải chuột vào nó để hiển thị menu lệnh, chọn chức năng Add.Class để hiển thị cửa sổ “Add New Item”, chọn mục “Class”, hiệu chỉnh tên class là MyClass.cs, chọn button Add để máy tạo 1 class mới.

7. Khi cửa sổ soạn code cho class MyClass hiển thị, hiệu chỉnh nội dung của class như sau :

```

namespace DefAndUseObject {
    class MyClass : IMyInterface {
        //1. định nghĩa các thuộc tính vật lý
        private int m_x;
        protected int[] arr = new int[100];
        //2. định nghĩa các tác vụ quản lý đối tượng
        public MyClass() {
            this.Click += new EventHandler(button1_Click);
        }
        ~MyClass() {} //hàm destructor
        //3. định nghĩa đối tượng đại diện hàm chức năng
        public delegate void EventHandler(Object sender, EventArgs e);
        //4. định nghĩa sự kiện Click được xử lý bởi delegate EventHandler.
        public event EventHandler Click;
        //5. định nghĩa các tác vụ & toán tử chức năng
        public void button1_Click(object sender, System.EventArgs e) {
            Console.WriteLine("Ham xu ly Click cua class MyClass chay");
        }
        public void func2() {

```

```

    Console.WriteLine("Ham MyClass.func2() đang chạy");
}

//6. định nghĩa tác vụ kích hoạt sự kiện Click
public void OnClick(Object sender, EventArgs e) {
    if (Click != null) Click(sender, e);
}

//7. định nghĩa thuộc tính luận lý x
public int x {
    get { return m_x; }
    set { m_x = value; }
}

//8. định nghĩa indexer
public int this[int index] {
    get {
        //kiểm tra giới hạn để quyết định
        if (index < 0 || index >= 100) { return 0; }
        else { return arr[index]; }
    }
    set {
        if (0 <= index && index < 100) { arr[index] = value; }
    }
}
} //kết thúc class
} //kết thúc namespace

```

8. Chọn menu Debug.Start Debugging để dịch và chạy ứng dụng. Ta thấy máy báo lỗi ở 2 lệnh truy xuất thành phần bị che dấu bởi đối tượng (có tầm vực private, protected).
9. Hãy chú thích 2 dòng lệnh bị lỗi rồi dịch và chạy lại ứng dụng, cửa sổ Console hiển thị các thông báo xuất của chương trình. Hãy kiểm tra từng lệnh xem điều gì xảy ra và cố gắng lý giải kết quả.
10. Hãy hiệu chỉnh lệnh định nghĩa biến `MyClass` obj=... ở đầu hàm Main thành :

```
IMyInterface obj = new MyClass();
```

rồi dịch và chạy lại ứng dụng. Máy sẽ báo lỗi ở các hàng lệnh truy xuất đối tượng obj, máy chỉ cho phép dùng đúng lệnh gọi thông điệp `obj.func2()`; Lý giải : vì biến obj có kiểu là interface IMyInterface nên ta chỉ có thể truy xuất đúng 1 hàm func2 trong interface mà thôi, mọi chi tiết của đối tượng được tham khảo đều bị che dấu.