

MÔN : LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Bài thực hành số 5.1 : Viết chương trình khảo sát việc quản lý đời sống đối tượng và 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 thừa kế, bao đóng.
- Giúp SV hiểu rõ chi tiết về đời sống của đối tượng, hành vi của lệnh new & các tác vụ constructors, cách thức tắt toán đối tượng trước khi xóa đối tượng.
- Giúp SV hiểu rõ cách thức sử dụng đối tượng, thấy lợi ích của đa xạ.

II. Nội dung :

- Đặc tả 1 interface chung và 3 class ClassA, ClassB, ClassC hiện thực interface này và dựa trên cơ chế thừa kế.
- Xây dựng chương trình nhỏ chạy ở chế độ textmode (console) tạo từng đối tượng thuộc từng class rồi sử dụng nó.

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 và chủ động kiểm soát được đời sống của từng đối tượng được dùng trong phần mềm.
- Sinh viên viết thành thạo các lệnh gọi thông điệp và nắm vững cơ chế đa xạ.

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. UseObject), 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 UseObject {  
    class Program {  
        static void Main(string[] args) {  
            Console.WriteLine("Lenh ClassA ca = new ClassA(); chạy. Hay quan sat trinh tu cac  
constructor.");  
            ClassA ca = new ClassA();  
            Console.WriteLine();  
            Console.WriteLine("Lenh ca.func1(); chạy. Hay quan sat ham cu the nao chay");  
            ca.func1();  
            Console.WriteLine();  
            Console.WriteLine("Lenh ca.func2(); chạy. Hay quan sat ham cu the nao chay");  
            ca.func2();  
            Console.WriteLine();  
            Console.WriteLine("Lenh ca = new ClassB (); chạy. Hay quan sat trinh tu cac  
constructor.");  
            ca = new ClassB ();  
            Console.WriteLine();  
            Console.WriteLine("Lenh ca.func1(); chạy. Hay quan sat ham cu the nao chay");  
            ca.func1();  
            Console.WriteLine();  
        }  
    }  
}
```

```

    Console.WriteLine("Lenh ca.func2(); chay. Hay quan sat ham cu the nao chay");
    ca.func2();
    Console.WriteLine();
    Console.WriteLine("Lenh ca = new ClassC (1,2); chay. Hay quan sat trinh tu cac
constructor.");
    ca = new ClassC (1,2);
    Console.WriteLine();
    Console.WriteLine("Lenh ca.func1(); chay. Hay quan sat ham cu the nao chay");
    ca.func1();
    Console.WriteLine();
    Console.WriteLine("Lenh ca.func2(); chay. Hay quan sat ham cu the nao chay");
    ca.func2();
    Console.WriteLine();
    Console.WriteLine("Lenh cc.func3(); chay. Hay quan sat ham cu the nao chay");
    cc.func3(); //bi loi vi func3 la ham private
    Console.WriteLine();
    Console.WriteLine("Lenh cc.func4(); chay. Hay quan sat ham cu the nao chay");
    ca.func4();
    Console.WriteLine();
    Console.Write("Ấn Enter để dừng chương trình : ");
    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 IA.cs, chọn button Add để máy tạo 1 interface mới.

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

```

namespace UseObject {
    interface IA {
        void func2(); //hàm dịch vụ duy nhất trong interface
    }
}

```

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 ClassA.cs, chọn button Add để máy tạo 1 class mới.

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

```

namespace UseObject {
    class ClassA : IA {
        private int i;
        protected double d;
        //constructor 0 tham số
        public ClassA() : base() {
            i = 1; d = 1.1416;
            Console.WriteLine("Constructor for ClassA() is running...");
        }

        //constructor 2 tham số
        public ClassA(int i, double d) : base() {
            this.i = i; this.d = d;
        }
    }
}

```

```

    Console.WriteLine("Constructor for ClassA(i,d) is running...");
}

//destructor
~ClassA() {
    Console.WriteLine("Destructor for ClassA is running...");
}

//hàm func1 không thể bị override
public int func1() {
    Console.WriteLine("Ham ClassA::func1 chay.");
    return 1;
}

//hàm func2 có thể bị override
virtual public void func2() {
    Console.WriteLine("Ham ClassA::func2 chay.");
}

//hàm func3 không thể bị override
private void func3() {
    Console.WriteLine("Ham ClassA::func3 chay.");
}

//hàm func4 không thể bị override
public void func4() {
    Console.WriteLine("Ham ClassA::func4 chay.");
    func1(); //lien ket tinh --> khong da xa
    func2(); //lien ket dong --> da xa
}
}
}

```

8. 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 ClassB.cs, chọn button Add để máy tạo 1 class mới.
9. Khi cửa sổ soạn code cho class ClassB hiển thị, hiệu chỉnh nội dung của class như sau :

```

namespace UseObject {
    class ClassB : ClassA {
        //constructor 0 tham số
        public ClassB() : base() {
            Console.WriteLine("Constructor for ClassB() is running...");
        }

        //constructor 2 tham số
        public ClassB(int i, double d) : base(i,d) {
            Console.WriteLine("Constructor for ClassB(i,d) is running...");
        }

        //destructor
        ~ClassB() {
            Console.WriteLine("Destructor for ClassB is running...");
        }
    }
}

```

```

}

/* không cho phép override hàm func1
public override int func1() {
    Console.WriteLine("Ham ClassB::func1 chay.");
    return 1;
} */

//override hàm func2
public override void func2() {
    Console.WriteLine("Ham ClassB::func2 chay.");
}
}
}

```

10. 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 ClassB.cs, chọn button Add để máy tạo 1 class mới.

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

```

namespace UseObject {
    class ClassC : ClassB {
        //constructor 0 tham số
        public ClassC() : base() {
            Console.WriteLine("Constructor for ClassC() is running...");
        }

        //constructor 2 tham số
        public ClassC(int i, double d) : base(i,d) {
            Console.WriteLine("Constructor for ClassC(i,d) is running...");
        }

        //destructor
        ~ClassC() {
            Console.WriteLine("Destructor for ClassC is running...");
            Console.WriteLine("Ấn Enter để chạy tiếp : ");
            Console.Read();
        }

        /* không cho phép override hàm func1
        public override int func1() {
            Console.WriteLine("Ham ClassC::func1 chay.");
            return 1;
        } */

        //override hàm func2
        public override void func2() {
            Console.WriteLine("Ham ClassC::func2 chay.");
        }
    }
}
}

```

12. Chọn menu Debug.Start Debugging để dịch và chạy ứng dụng. Ta thấy máy báo lỗi ở lệnh **cc.func3()**; vì func3() có tầm vực private nên không thể truy xuất từ ngoài.

13. Hãy chú thích 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.

14. Hãy hiệu chỉnh lệnh định nghĩa biến `ClassA` ca ở đầu hàm Main thành :

```
IA ca = new ClassA();
```

rồi dịch và chạy lại ứng dụng. Máy sẽ báo lỗi ở các hàng lệnh `ca.func1()`; Lý giải : vì biến ca có kiểu là interface IA 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.