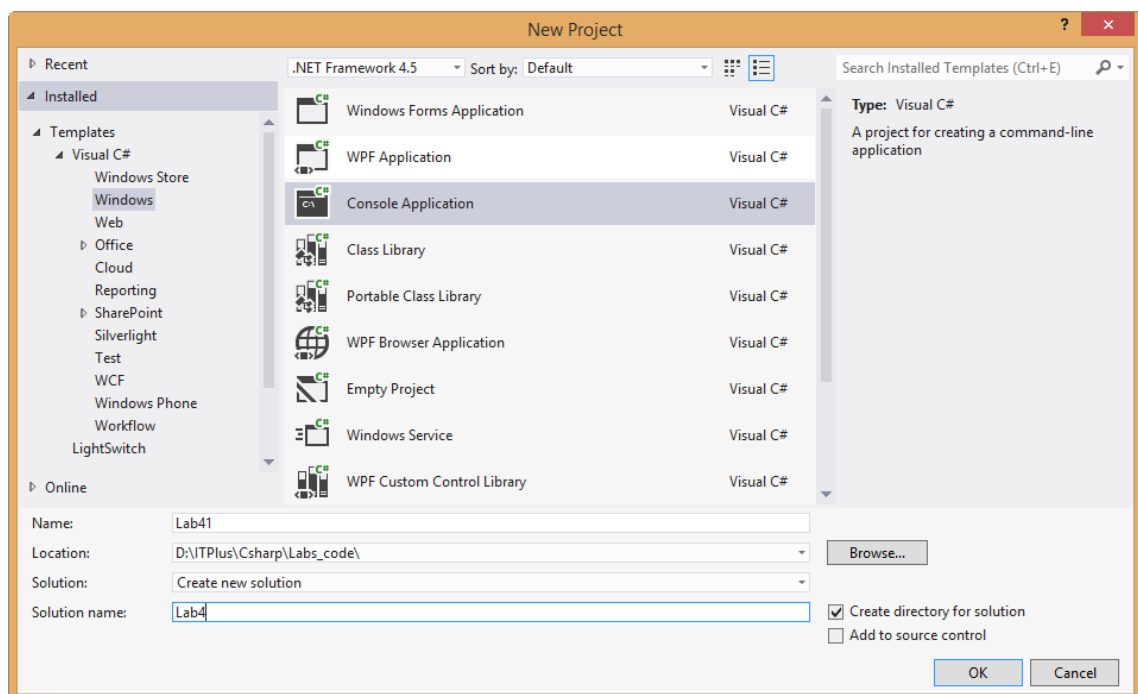


# LAB 9 Nội dung: Kế thừa và Đa hình

## I. Các bài thực hành step by step

**Bài 1** Tạo lớp Student với các trường (id,name, age), tạo các properties và hiển thị thông tin Student. Tạo lớp StudentAction và khởi tạo 1 mảng sinh viên, tạo các phương thức overloading để tìm kiếm sinh viên theo các tiêu chí khác nhau.

**Bước 1:** Mở visual 2012, tạo project có tên “Lab41” như hình sau.



**Bước 2:** Kích chuột phải vào project “Lab41” chọn Add-> Class và nhập tên class là “Student” -> nhấn nút Add.

**Bước 3:** Nhập mã cho lớp Student như sau.

```
class Student
{
    //tạo các thuộc tính tự động
    public int Id { get; set; }
    public string Name { get; set; }
```

```

    public int Age { get; set; }
    //phương thức hiển thị
    public void Display()
    {
        Console.WriteLine("Id:" + Id);
        Console.WriteLine("Name:" + Name);
        Console.WriteLine("Age:" + Age);
    }
}

```

**Bước 4:** Kích chuột phải vào project “Lab41” chọn Add-> Class và nhập tên class là “StudentAction” -> nhấn nút Add.

**Bước 5:** Code cho lớp “StudentAction” như sau.

```

class StudentAction
{
    //khai báo danh tập hợp chứa các sinh viên
    List<Student> liststudent;
    //phương thức khởi tạo
    public StudentAction()
    {
        //khởi tạo danh sách sinh viên
        liststudent = new List<Student> {
            new Student(){Id=1,Name="Dung",Age=20},
            new Student(){Id=2,Name="Tuan",Age=25},
            new Student(){Id=3,Name="Ha",Age=18},
            new Student(){Id=4,Name="Anh",Age=30},
            new Student(){Id=5,Name="Long",Age=24},
        };
    }
    #region các phương thức overloading
    //phương thức trả về danh sách sinh viên
    public List<Student> GetStudent()
    {
        return liststudent;
    }
    //phương thức trả về sinh viên theo id
    public Student GetStudent(int id)
    {
        Student st=null;
        foreach (var item in liststudent)
        {
            if (item.Id == id)
                st = item;
        }
        return st;
    }
}

```

```

//phương thức trả về sinh viên có tuổi từ x to y
public List<Student> GetStudent(int x, int y)
{
    List<Student> result = new List<Student>();
    foreach (var item in liststudent)
    {
        if (item.Age >= x && item.Age <= y)
            result.Add(item);
    }
    return result;
}
#endregion
}

```

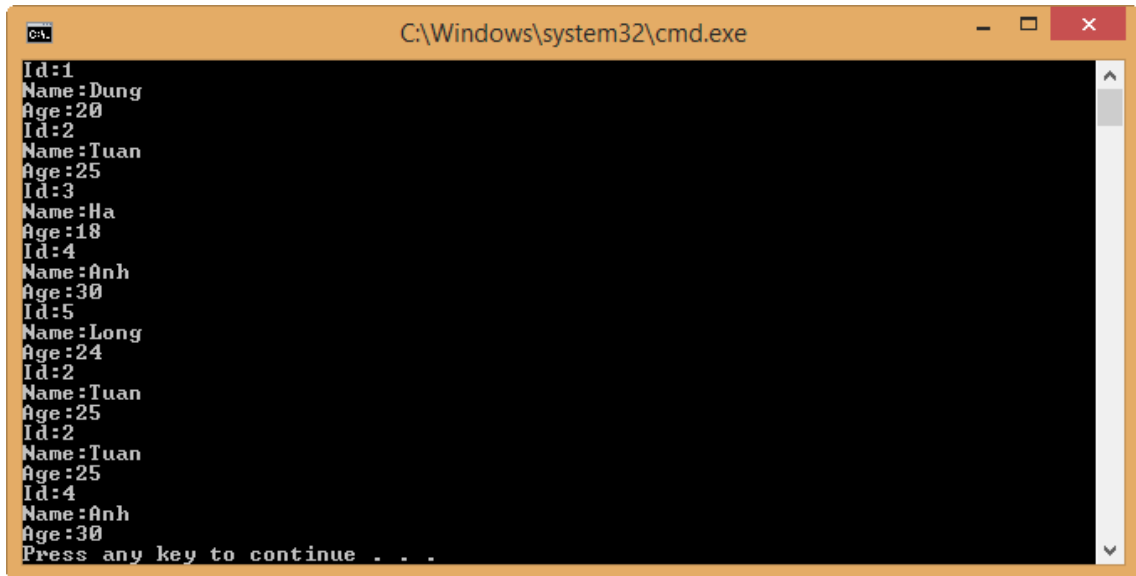
**Bước 6:** Mở tệp Program.cs gõ code như sau.

```

class Program
{
    static void Main(string[] args)
    {
        //tạo đối tượng
        StudentAction action = new StudentAction();
        //gọi phương thức lấy tất cả sinh viên
        List<Student> listall= action.GetStudent();
        //hiển thị
        foreach (var item in listall)
        {
            item.Display();
        }
        //gọi phương thức lấy sinh viên theo id
        Student st=action.GetStudent(2);
        //hiển thị
        st.Display();
        //gọi phương thức lấy sinh viên có tuổi từ 25-30
        List<Student> listage = action.GetStudent(25,30);
        //hiển thị
        foreach (var item in listage)
        {
            item.Display();
        }
    }
}

```

**Bước 7:** Nhấn **Ctrl+F5** để xem kết quả.



```
C:\Windows\system32\cmd.exe
Id:1
Name:Dung
Age:20
Id:2
Name:Tuan
Age:25
Id:3
Name:Ha
Age:18
Id:4
Name:Anh
Age:30
Id:5
Name:Long
Age:24
Id:2
Name:Tuan
Age:25
Id:2
Name:Tuan
Age:25
Id:4
Name:Anh
Age:30
Press any key to continue . . .
```

**Bài 2:** *Tạo ứng dụng mô phỏng kế thừa và đa hình theo các bước như sau.*

**Bước 1:** Kích chuột phải vào Solution “Lab4” -> Add->New Project ->Nhập tên project “Lab42” -> OK.

**Bước 2:** Kích chuột phải vào project “Lab42” chọn Add-> Class và nhập tên class là “Window” -> nhấn nút Add.

**Bước 3:** Viết code cho lớp Window như sau.

```
class Window
{
    //khai báo các trường
    protected int top;
    protected int left;
    //Phương thức khởi tạo
    public Window(int top, int left)
    {
        this.top = top;
        this.left = left;
    }

    //phương thức virtual vẽ cửa sổ
    public virtual void DrawWindow()
    {
        Console.WriteLine("Window: drawing Window at {0}, {1}",top, left);
    }
}
```

**Bước 4:** Kích chuột phải vào project “Lab42” chọn Add-> Class và nhập tên class là “ListBox” -> nhấn nút Add.

**Bước 5:** Viết code cho lớp ListBox như sau.

```
//tạo lớp listBox kế thừa từ window
class ListBox : Window
{
    //Khai báo trường mới
    private string listBoxContents;
    //Phương thức khởi tạo
    public ListBox(int top,int left, string contents)
        : base(top, left) // gọi constructor lớp cơ sở
    {
        listBoxContents = contents;
    }
    // ghi đè phương thức DrawWindow
    public override void DrawWindow()
    {
        base.DrawWindow(); // gọi phương thức lớp cơ sở
        Console.WriteLine("Writing string to the listbox: {0}",
            listBoxContents);
    }
}
```

**Bước 6:** Kích chuột phải vào project “Lab42” chọn Add-> Class và nhập tên class là “Button” -> nhấn nút Add.

**Bước 7:** Viết code cho lớp Button như sau.

```
class Button: Window
{
    //phương thức khởi tạo
    public Button(int top, int left)
        : base(top, left)//gọi constructor lớp cơ sở
    {
    }
    // Ghi đè phương thức
    public override void DrawWindow()
    {
        Console.WriteLine("Drawing a button at {0}, {1}\n", top, left);
    }
}
```

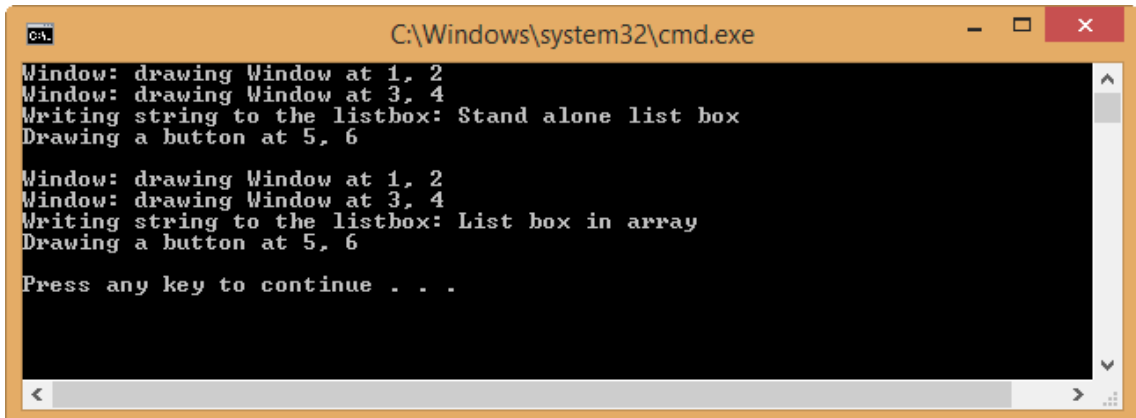
**Bước 8:** Mở tệp Program.cs gõ code như sau.

```

class Program
{
    static void Main(string[] args)
    {
        Window win = new Window(1, 2);
        ListBox lb = new ListBox(3, 4, "Stand alone list box");
        Button b = new Button(5, 6);
        win.DrawWindow();
        lb.DrawWindow();
        b.DrawWindow();
        Window[] winArray = new Window[3];
        winArray[0] = new Window(1, 2);
        winArray[1] = new ListBox(3, 4, "List box in array");
        winArray[2] = new Button(5, 6);
        for (int i = 0; i < 3; i++)
        {
            winArray[i].DrawWindow();
        }
    }
}

```

**Bước 7:** Nhấn **Ctrl+F5** để xem kết quả.



```

C:\Windows\system32\cmd.exe
Window: drawing Window at 1, 2
Window: drawing Window at 3, 4
Writing string to the listbox: Stand alone list box
Drawing a button at 5, 6

Window: drawing Window at 1, 2
Window: drawing Window at 3, 4
Writing string to the listbox: List box in array
Drawing a button at 5, 6

Press any key to continue . . .

```

## II. Bài tập tự làm

### 1. Tạo ứng dụng C# với các yêu cầu sau.

- Lớp *Account* lưu thông tin về tài khoản (*Balance* – số tiền trong tài khoản). Tạo 1 phương thức constructor với tham số đầu vào là *initialize* (số tiền ban đầu).
- Tạo 2 phương thức cho lớp *Account*.

- *Deposit(money)*: là phương thức gửi tiền cho ghi đề, money là số tiền cần gửi, hãy kiểm tra các thông tin cho phù hợp và xuất thông báo.
- *Withdraw(money)*: là phương thức rút tiền cho ghi đề, money là số tiền rút, hãy kiểm tra các thông tin cho phù hợp và xuất thông báo.
- *GetBalance()*: là phương thức trả về số tiền trong tài khoản.
- *Lớp SavingAccount* (gửi tiết kiệm) kế thừa từ lớp *Account* có thêm thông số *rate* (tỉ lệ lãi suất), tạo phương thức constructor với 2 tham số đầu vào, *initialize* (số tiền ban đầu), *rate*, kế thừa constructor của lớp *Account*. Tạo phương thức *GetInterest()* (lấy tiền lãi).
- *Lớp CheckAccount* (vấn tin, gửi tiền, rút tiền) kế thừa từ lớp *Account* và có thêm biến *FeeTransfer* (phí giao dịch). Tạo phương thức constructor với 2 tham số đầu vào *Balance*, *FeeTransfer*, sau đó ghi đề 2 phương thức *Deposit* và *Withdraw*, *GetBalance*, mỗi khi gọi phương thức thành công thì số tiền trong tài khoản sẽ bị trừ đi 1 khoản bằng *FeeTransfer*.
- *Lớp Program* để kiểm thử ứng dụng.