

Lab 04

Inheritance and Polymorphism

Abstract Classes and Interfaces

Mục tiêu	2
Phần I Bài tập step by step	3
Bài 4.1	3
Bài 4.2	7
Bài 4.3	11
Bài 4.4	15
Bài 4.5	19
Phần II Bài tập tự làm	23
Bài 4.1	23
Bài 4.2	24
Bài 4.3	25
Bài 4.4	27

Mục tiêu

- Tạo và xây dựng các lớp có tính kế thừa
- Tạo lớp trừu tượng
- Tạo giao diện và thực thi giao diện
- Triển khai đa hình trong hướng đối tượng

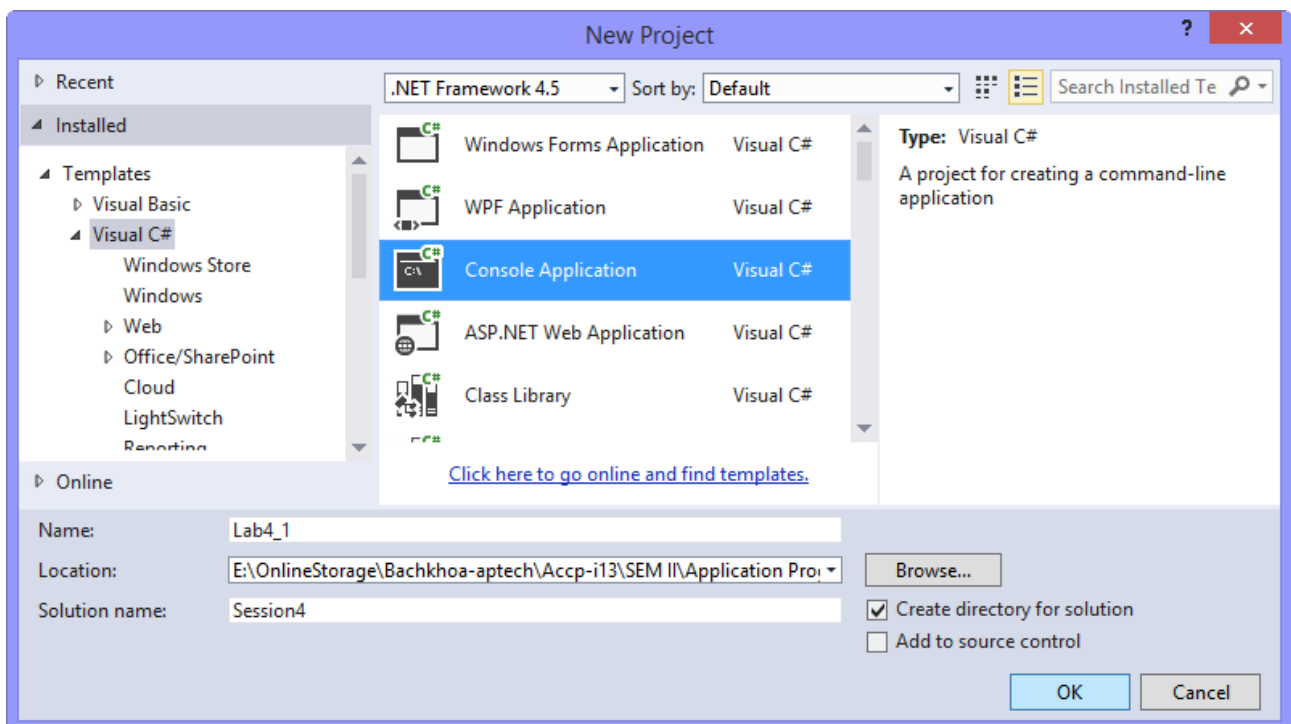
Phần I Bài tập step by step

Bài 4.1

Viết chương trình C# thực hiện các công việc sau:

- Tạo lớp Window với 2 trường top, left. Tạo phương thức DrawWindow và in ra thông báo.
- Tạo lớp Button kế thừa từ Window và ghi đè phương thức DrawWindow
- Tạo lớp ListBox kế thừa từ Window bổ sung thêm trường content và ghi đè phương thức DrawWindow.
- Trong program viết code test 2 lớp trên.

BƯỚC 1: Mở Visual Studio 2013, vào menu File -> New -> Project -> chọn loại project “Console Application”, nhập tên project, tên solution -> OK.



BƯỚC 2: Tạo lớp Windows với code gợi ý 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 3: Tạo tệp Button với code gợi ý 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 4: Tạo tệp ListBox với code gợi ý như sau:

```
//tạo lớp listbox kế thừa từ window
```

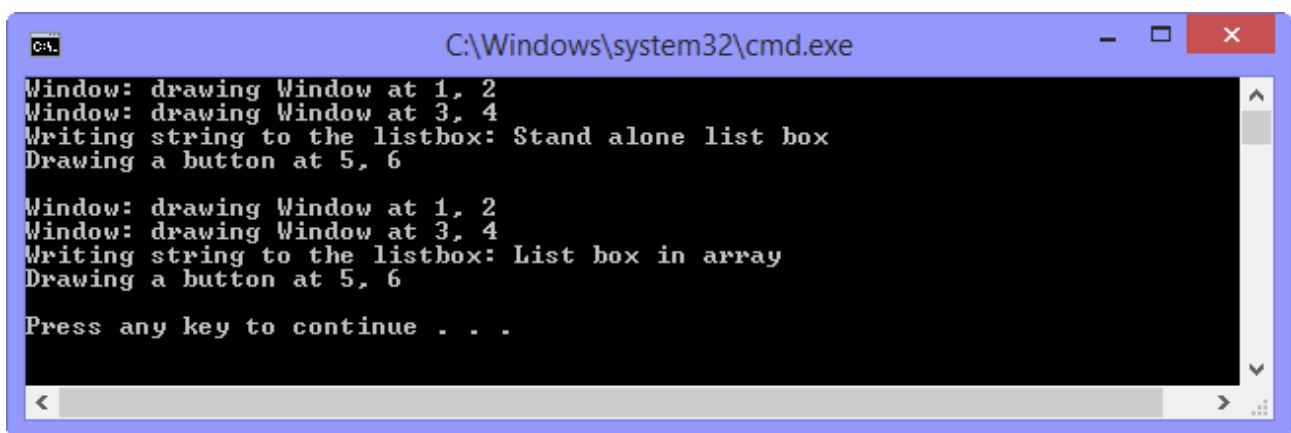
```
class ListBox : Window
{
    //Khai báo trường mới
    private string listBoxContents;
    //Phương thức khởi tạo
    public ListBox(int top, int left, string contents)
    : base(top, left) // gọi constructor lớp cơ sở
    {
        listBoxContents = contents;
    }
    // ghi đè phương thức DrawWindow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

BƯỚC 5: Mở tệp Program.cs và code cho hàm Main theo gợi ý sau:

```
static void Main(string[] args)
{
    //Khởi tạo lớp windows
    Window win = new Window(1, 2);
    //Khởi tạo lớp listbox
    ListBox lb = new ListBox(3, 4, "Stand alone list box");
    //khởi tạo lớp button
    Button b = new Button(5, 6);
    win.DrawWindow();
    lb.DrawWindow();
    b.DrawWindow();
    //khởi tạo mảng windows
    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 6: Nhấn Ctrl+F5 để chạy và xem kết quả



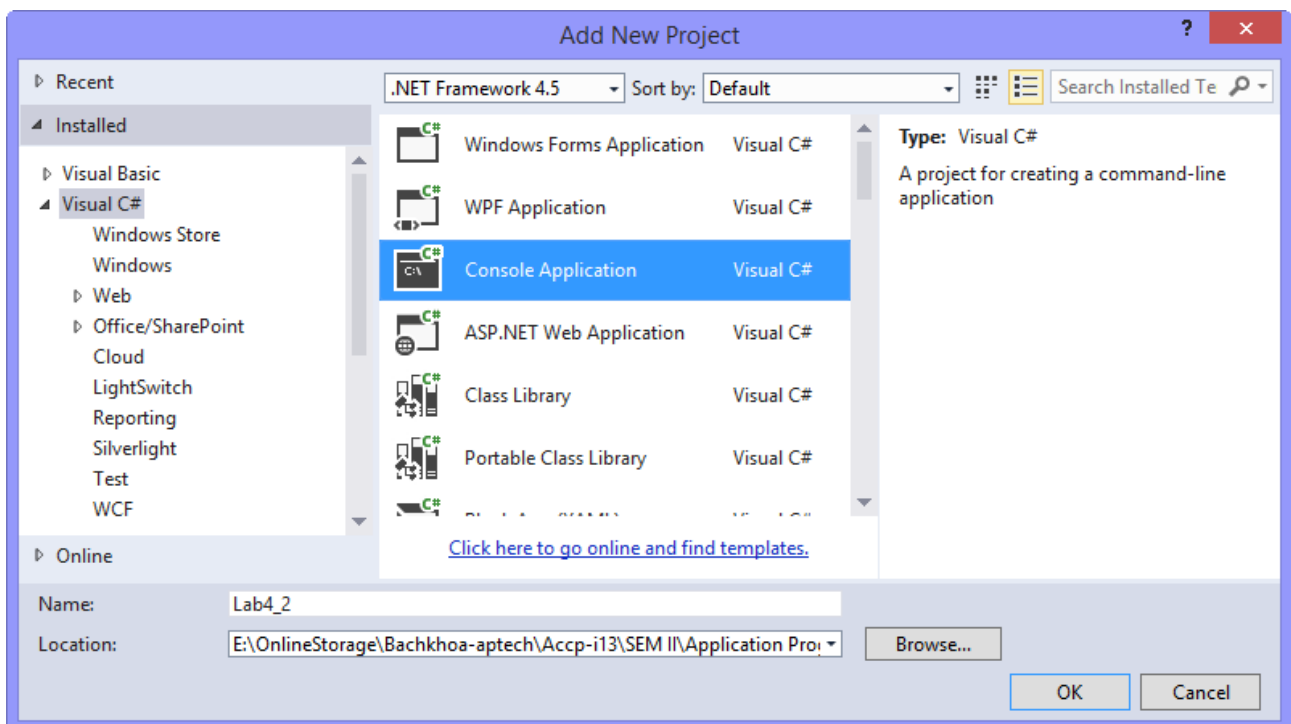
```
C:\Windows\system32\cmd.exe  
Window: drawing Window at 1, 2  
Window: drawing Window at 3, 4  
Writing string to the listbox: Stand alone list box  
Drawing a button at 5, 6  
  
Window: drawing Window at 1, 2  
Window: drawing Window at 3, 4  
Writing string to the listbox: List box in array  
Drawing a button at 5, 6  
  
Press any key to continue . . .
```

Bài 4.2

Viết chương trình C# thực hiện các công việc sau:

- Tạo lớp trừu tượng Student với các trường name, year, phương thức hiển thị, phương thức trừu tượng Average (tính trung bình điểm)
- Tạo lớp StudentCollege kế thừa từ Student và bổ sung 3 trường score1, score2, score3. Ghi đè phương thức Average và tạo mới phương thức hiển thị.
- Tạo lớp StudentUniversity kế thừa từ StudentCollege và bổ sung 1 trường score4. Ghi đè phương thức Average và tạo mới phương thức hiển thị.
- Trong Main viết code để test chương trình.

BƯỚC 1: Kích chuột phải vào Solution “Session4” chọn Add -> New Project -> nhập tên.



BƯỚC 2: Tạo lớp Student và code theo gợi ý sau:

```
abstract class Student
{
    protected string name;
```

```
protected int year;
// Phương thức khởi tạo
public Student(string name, int year)
{
    this.name = name;
    this.year = year;
}
// phương thức hiển thị thông tin
protected void Display()
{
    Console.WriteLine("Name: {0}", name);
    Console.WriteLine("Year: {0}", year);
}
//phương thức tính điểm trung bình
public abstract double Average();
}
```

Bước 3: Tạo lớp StudentCollege và code theo gợi ý sau:

```
/// <summary>
/// Lớp sinh viên cao đẳng kế thừa từ lớp sinh viên
/// </summary>
class StudentCollege: Student
{
    protected double score1, score2, score3;
    //constructor
    public StudentCollege(string name, int year, double score1, double
score2, double score3):base(name,year)
    {
        this.score1 = score1;
        this.score2 = score2;
        this.score3 = score3;
    }
    //ghi đè phương thức tính trung bình điểm
    public override double Average()
```



```
{  
    return (score1 + score2 + score3) / 3;  
}  
//hiển thị thông tin  
public new void Display()  
{  
    base.Display();  
    Console.WriteLine("Score 1: {0:N}", score1);  
    Console.WriteLine("Score 2: {0:N}", score2);  
    Console.WriteLine("Score 3: {0:N}", score3);  
}  
}
```

Bước 4: Tạo lớp StudentUniversity và code theo gợi ý sau:

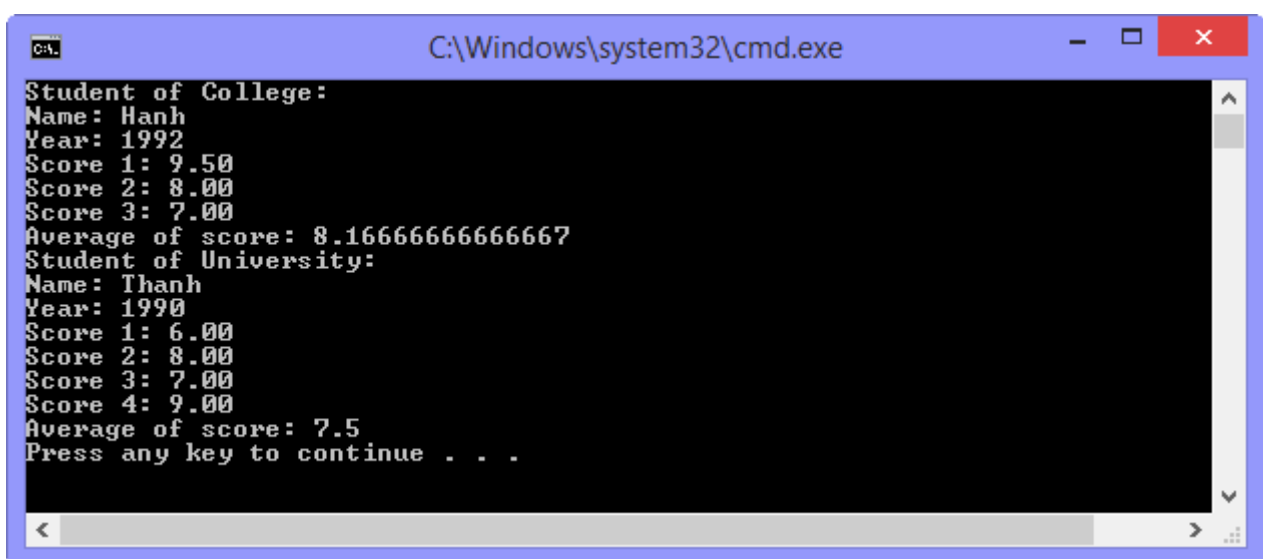
```
/// <summary>  
/// Lớp sinh viên đại học kế thừa từ sinh viên cao đẳng  
/// </summary>  
class StudentUniversity:StudentCollege  
{  
    protected double score4;  
    //constructor  
    public StudentUniversity(string name, int year, double score1, double  
score2, double score3, double score4)  
    : base(name, year, score1, score2, score3)  
    {  
        this.score4 = score4;  
    }  
    //phương thức tính trung bình điểm  
    public override double Average()  
    {  
        return (score1 + score2 + score3 + score4) / 4;  
    }  
    // phương thức hiển thị thông tin  
    public new void Display()
```

```
{  
    base.Display();  
    Console.WriteLine("Score 4: {0:N}", score4);  
}  
}
```

BƯỚC 5: Trong Program code test như sau:

```
static void Main(string[] args)  
{  
    //khởi tạo sinh viên cao đẳng và in ra thông tin  
    Console.WriteLine("Student of College:");  
    StudentCollege sv1 = new StudentCollege("Hanh", 1992, 9.5, 8, 7);  
    sv1.Display();  
    Console.WriteLine("Average of score: {0}", sv1.Average());  
    //khởi tạo sinh viên đại học và in ra thông tin  
    Console.WriteLine("Student of University:");  
    StudentUniversity sv2 = new StudentUniversity("Thanh", 1990, 6, 8, 7, 9);  
    sv2.Display();  
    Console.WriteLine("Average of score: {0}", sv2.Average());  
}
```

BƯỚC 6: Nhấn Ctrl+F5 để chạy và xem kết quả



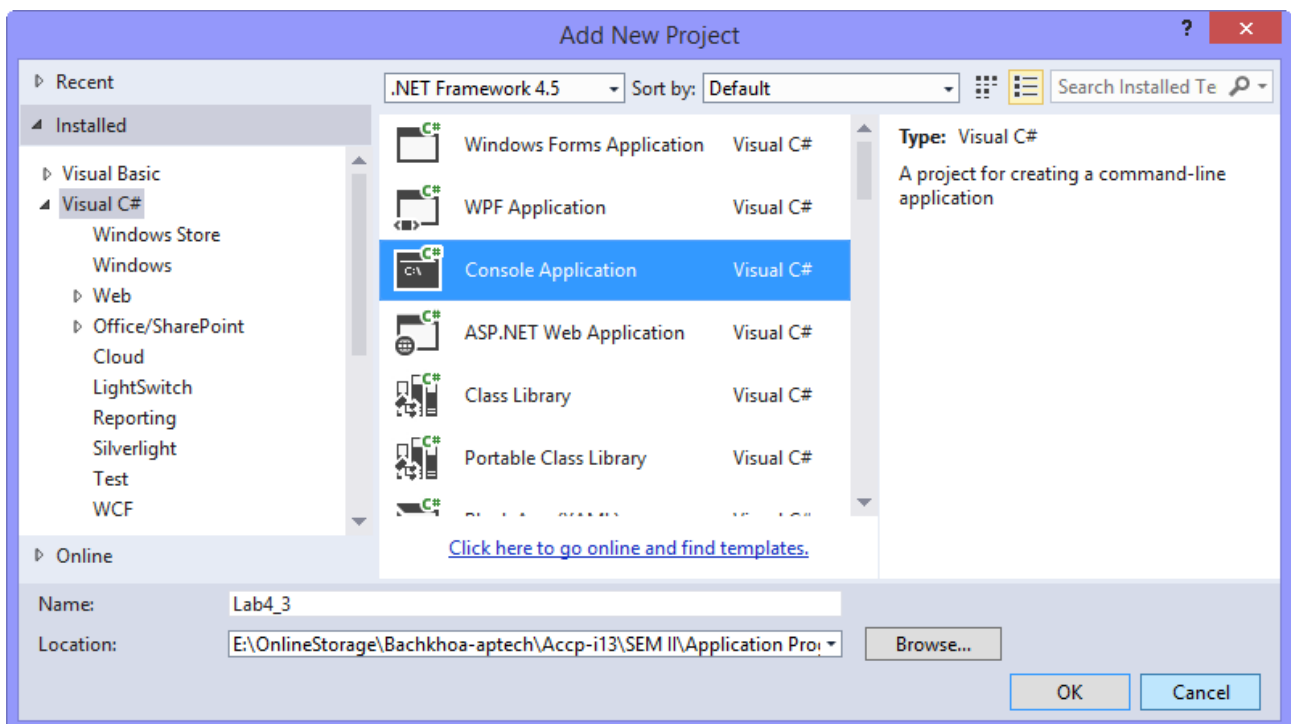
```
C:\Windows\system32\cmd.exe  
Student of College:  
Name: Hanh  
Year: 1992  
Score 1: 9.50  
Score 2: 8.00  
Score 3: 7.00  
Average of score: 8.166666666666667  
Student of University:  
Name: Thanh  
Year: 1990  
Score 1: 6.00  
Score 2: 8.00  
Score 3: 7.00  
Score 4: 9.00  
Average of score: 7.5  
Press any key to continue . . .
```

Bài 4.3

Viết chương trình C# thực hiện các công việc sau:

- Tạo lớp trừu tượng Shape với các trường radius, length, width. Khai báo hai phương thức trừu tượng tính diện tích Area và tính chu vi Circumference
- Tạo lớp Circle kế thừa từ Shape và thực thi các phương thức của lớp Shape
- Tạo lớp Rectangle kế thừa từ Shape và thực thi các phương thức của lớp Shape
- Trong Main viết code để test chương trình.

BƯỚC 1: Kích chuột phải vào Solution “Session4” chọn Add -> New Project -> nhập tên.



BƯỚC 2: Tạo lớp Shape và code theo gợi ý sau:

```
//định nghĩa lớp trừu tượng
abstract class Shape
{
    protected float radius, length, width;
    //Khai báo các phương thức trừu tượng
    public abstract float Area(); //tính diện tích
```

```
        public abstract float Circumference();//tính chu vi  
    }
```

Bước 3: Tạo lớp Circle và code theo gợi ý sau:

```
class Circle : Shape  
{  
    //phương thức nhập liệu  
    public void InputData()  
    {  
        Console.Write("Nhap ban kinh:");  
        radius = int.Parse(Console.ReadLine());  
    }  
    //ghi đè phương thức tính diện tích  
    public override float Area()  
    {  
        return 3.14F * radius * radius;  
    }  
    //ghi đè phương thức tính chu vi  
    public override float Circumference()  
    {  
        return 2 * 3.14F * radius;  
    }  
}
```

Bước 4: Tạo lớp Rectangle và code theo gợi ý sau:

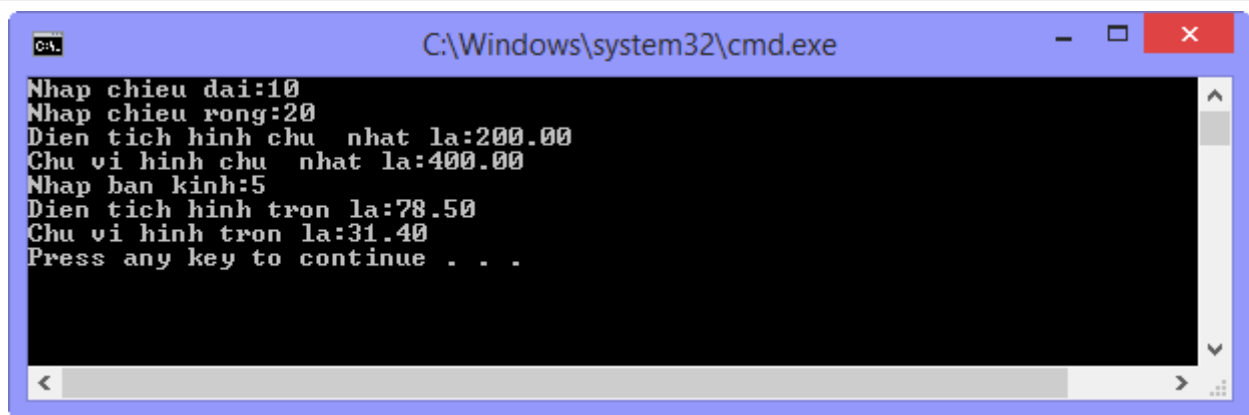
```
class Rectangle : Shape  
{  
    //phương thức nhập liệu  
    public void InputData()  
    {  
        Console.Write("Nhap chieu dai:");  
        length = int.Parse(Console.ReadLine());  
        Console.Write("Nhap chieu rong:");  
        width = int.Parse(Console.ReadLine());  
    }  
}
```

```
}  
//ghi đề phương thức tính diện tích  
public override float Area()  
{  
    return length * width;  
}  
//ghi đề phương thức tính chu vi  
public override float Circumference()  
{  
    return 2 * (length + width);  
}  
}
```

Bước 5: Trong Program code test như sau:

```
static void Main(string[] args)  
{  
    //Khởi tạo hình chữ nhật  
    Rectangle rec = new Rectangle();  
    //nhập dữ liệu  
    rec.InputData();  
    //in thông tin  
    Console.WriteLine("Diện tích hình chữ nhật là:{0:N}", rec.Area());  
    Console.WriteLine("Chu vi hình chữ nhật là:{0:N}", rec.Circumference());  
    //Khởi tạo hình tròn  
    Circle cir = new Circle();  
    //nhập dữ liệu  
    cir.InputData();  
    //in thông tin  
    Console.WriteLine("Diện tích hình tròn là:{0:N}", cir.Area());  
    Console.WriteLine("Chu vi hình tròn là:{0:N}", cir.Circumference());  
}
```

Bước 6: Nhấn Ctrl+F5 để chạy và xem kết quả



A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window has a blue title bar with standard Windows window controls (minimize, maximize, close). The command prompt shows the output of a C# program. The text is as follows:

```
Nhap chieu dai:10
Nhap chieu rong:20
Dien tich hinh chu nhat la:200.00
Chu vi hinh chu nhat la:400.00
Nhap ban kinh:5
Dien tich hinh tron la:78.50
Chu vi hinh tron la:31.40
Press any key to continue . . .
```

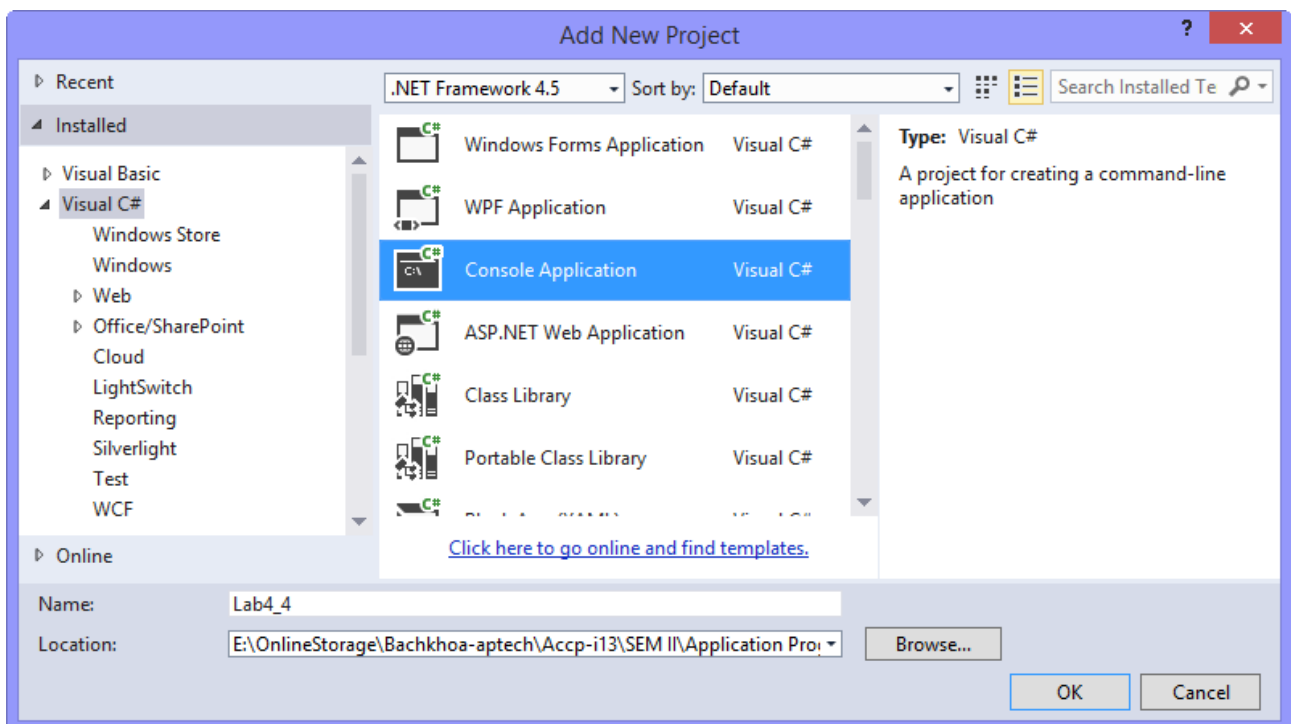
The output shows calculations for a rectangle and a circle. The rectangle has a length of 10 and a width of 20, resulting in an area of 200.00 and a perimeter of 400.00. The circle has a radius of 5, resulting in an area of 78.50 and a circumference of 31.40. The program ends with a "Press any key to continue" message.

Bài 4.4

Viết chương trình C# thực hiện các công việc sau:

- Tạo giao diện IPerson với các phương thức Insert, Delete, Update, Display.
- Tạo lớp Staff thực thi các phương thức trong giao diện IPerson.
- Tạo lớp Student với các trường id, name, age và thực thi các phương thức trong giao diện IPerson.
- Trong Main code để test.

BƯỚC 1: Kích chuột phải vào Solution “Session4” chọn Add -> New Project -> nhập tên.



BƯỚC 2: Tạo giao diện IPerson và code theo gợi ý sau:

```
//định nghĩa giao diện
interface IPerson
{
    //phương thức insert
    void Insert(object obj);
    //phương thức xóa
```

```
void Delete(object obj);  
//phương thức sửa  
void Update(object obj);  
//phương thức hiển thị  
void Display(object obj);  
}
```

Bước 3: Tạo lớp Staff và code theo gợi ý sau:

```
//Định nghĩa lớp nhân viên thực thi từ giao diện IPerson  
class Staff : IPerson  
{  
    //thực thi các phương thức của giao diện  
    public void Insert(object obj)  
    {  
        Console.WriteLine("Đã lưu đối tượng: " + obj);  
    }  
  
    public void Delete(object obj)  
    {  
        Console.WriteLine("Đã xóa đối tượng: " + obj);  
    }  
  
    public void Update(object obj)  
    {  
        Console.WriteLine("Đã cập nhật đối tượng: " + obj);  
    }  
  
    public void Display(object obj)  
    {  
        Console.WriteLine("Thông tin đối tượng: " + obj);  
    }  
}
```

Bước 4: Tạo lớp Student và code theo gợi ý sau:


```
//định nghĩa lớp sinh viên thực thi từ giao diện IPerson
class Student : IPerson
{
    //khai báo thuộc tính
    public string Id { get; set; }
    public string Name { get; set; }
    public int Age { get; set; }
    //thực thi các phương thức từ giao diện
    public void Insert(object obj)
    {
        Console.WriteLine("Da luu doi tuong: " + obj);
    }

    public void Delete(object obj)
    {
        Console.WriteLine("Da xoa doi tuong: " + obj);
    }

    public void Update(object obj)
    {
        Console.WriteLine("Da cap nhat doi tuong: " + obj);
    }

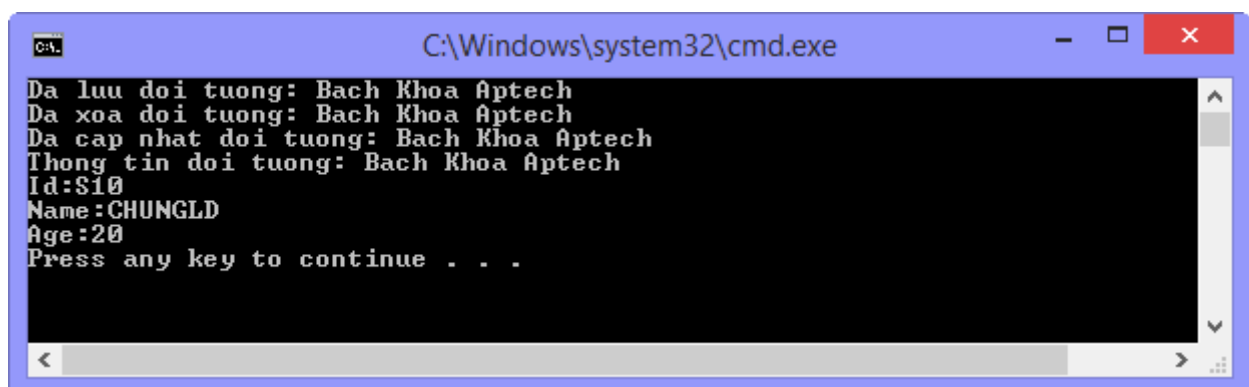
    public void Display(object obj)
    {
        Student st = (Student)obj;
        Console.WriteLine("Id:" + st.Id);
        Console.WriteLine("Name:" + st.Name);
        Console.WriteLine("Age:" + st.Age);
    }
}
```

BƯỚC 5: Trong Program code test như sau:

```
static void Main(string[] args)
```

```
{  
    //Tạo một nhân viên  
    IPerson staff = new Staff();  
    object data = "Bach Khoa Aptech";  
    //gọi các hành động  
    staff.Insert(data);  
    staff.Delete(data);  
    staff.Update(data);  
    staff.Display(data);  
    //tạo sinh viên  
    IPerson student = new Student() { Id = "S10", Name = "CHUNGLD", Age = 20  
    };  
    //hiển thị dữ liệu  
    student.Display(student);  
}
```

Bước 6: Nhấn Ctrl+F5 để chạy và xem kết quả

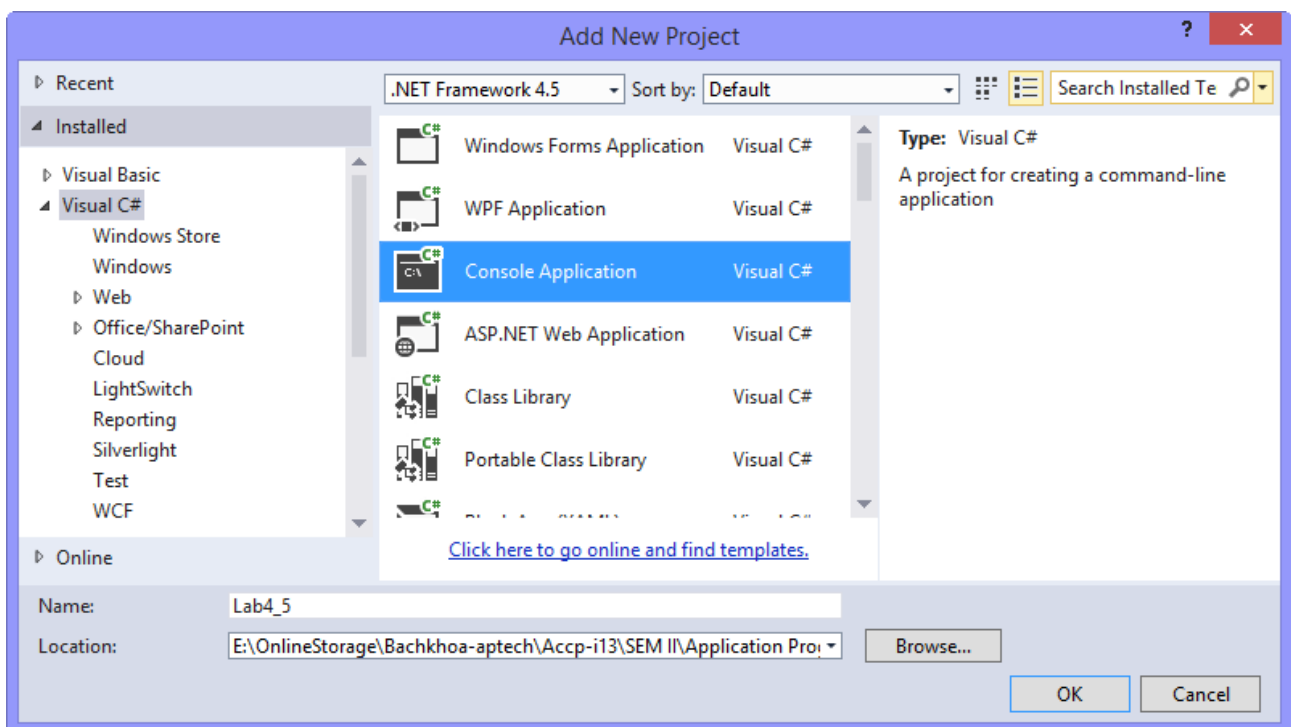


Bài 4.5

Viết chương trình C# thực hiện các công việc sau:

- Tạo giao diện IDrivable với hai phương thức Start và Stop và thuộc tính chỉ đọc Started.
- Tạo giao diện ISteerable với hai phương thức TurnLeft, TurnRight
- Tạo giao diện IMovable kế thừa từ 2 giao diện trên bổ sung thêm hai phương thức Accelerate và Brake
- Tạo lớp Car thực thi từ giao diện IMovable và code cho tất cả các phương thức có thể
- Trong Main code để test lớp Car.

BƯỚC 1: Kích chuột phải vào Solution “Session4” chọn Add -> New Project ->nhập tên.



BƯỚC 2: Tạo tệp MultipleInterface và cod cho các giao diện theo gợi ý sau:

```
// define the IDrivable interface  
public interface IDrivable
```

```
{  
    // method declarations  
    void Start();  
    void Stop();  
    // property declaration  
    bool Started  
    {  
        get;  
    }  
}  
// define the ISteerable interface  
public interface ISteerable  
{  
    // method declarations  
    void TurnLeft();  
    void TurnRight();  
}  
// define the IMovable interface (derived from IDrivable and ISteerable)  
public interface IMovable : IDrivable, ISteerable  
{  
    // method declarations  
    void Accelerate();  
    void Brake();  
}
```

BƯỚC 3: Tạo lớp Car và code theo gợi ý sau:

```
// Car class implements the IMovable interface  
public class Car : IMovable  
{  
    // declare the underlying field used by the  
    // Started property of the IDrivable interface  
    private bool started = false;  
  
    // implement the Start() method of the IDrivable interface
```

```
public void Start()
{
    Console.WriteLine("car started");
    started = true;
}
// implement the Stop() method of the IDrivable interface
public void Stop()
{
    Console.WriteLine("car stopped");
    started = false;
}
// implement the Started property of the IDrivable interface
public bool Started
{
    get
    {
        return started;
    }
}
// implement the TurnLeft() method of the ISteerable interface
public void TurnLeft()
{
    Console.WriteLine("car turning left");
}

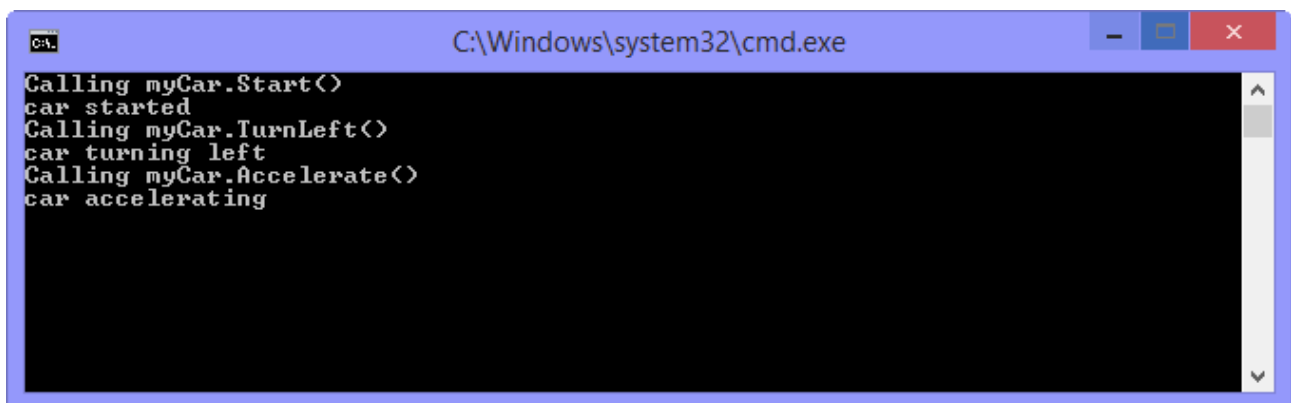
// implement the TurnRight() method of the ISteerable interface
public void TurnRight()
{
    Console.WriteLine("car turning right");
}
// implement the Accelerate() method of the IMovable interface
public void Accelerate()
{
    Console.WriteLine("car accelerating");
}
```

```
// implement the Brake() method of the IMovable interface
public void Brake()
{
    Console.WriteLine("car braking");
}
}
```

BƯỚC 4: Trong Program code test như sau:

```
static void Main(string[] args)
{
    // create a Car object
    Car myCar = new Car();
    // call myCar.Start()
    Console.WriteLine("Calling myCar.Start()");
    myCar.Start();
    // call myCar.TurnLeft()
    Console.WriteLine("Calling myCar.TurnLeft()");
    myCar.TurnLeft();
    // call myCar.Accelerate()
    Console.WriteLine("Calling myCar.Accelerate()");
    myCar.Accelerate();
    Console.ReadLine();
}
```

BƯỚC 5: Nhấn Ctrl+F5 để chạy và xem kết quả

A screenshot of a Windows Command Prompt window. The title bar shows the path "C:\Windows\system32\cmd.exe". The window contains the following text output:

```
Calling myCar.Start()
car started
Calling myCar.TurnLeft()
car turning left
Calling myCar.Accelerate()
car accelerating
```

Phần II Bài tập tự làm

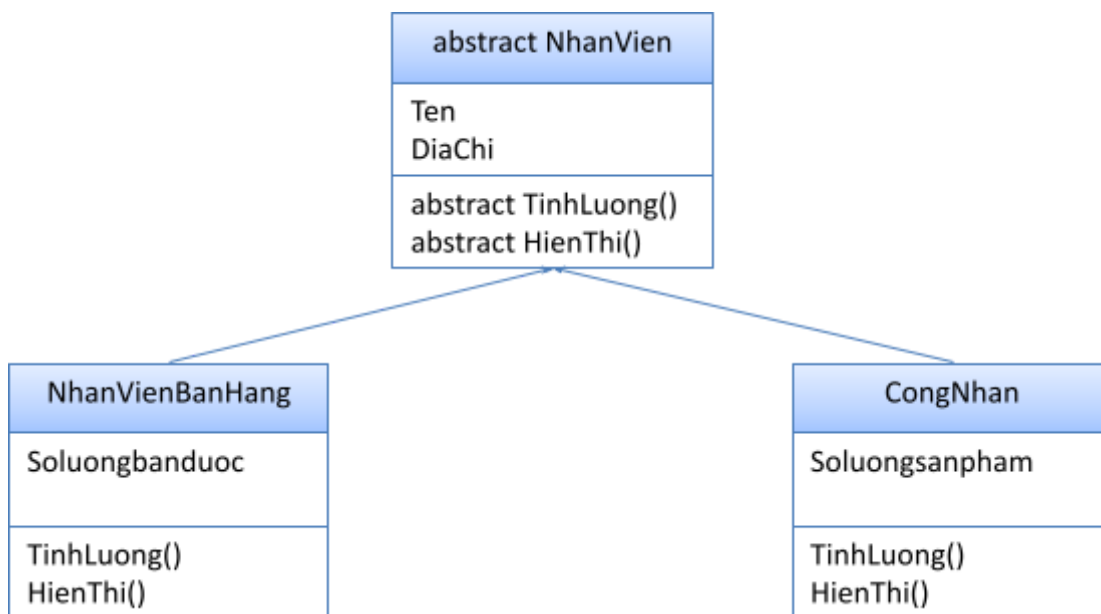
Bài 4.1

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.
 - o 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.
 - o 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.
 - o 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(tỷ 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.

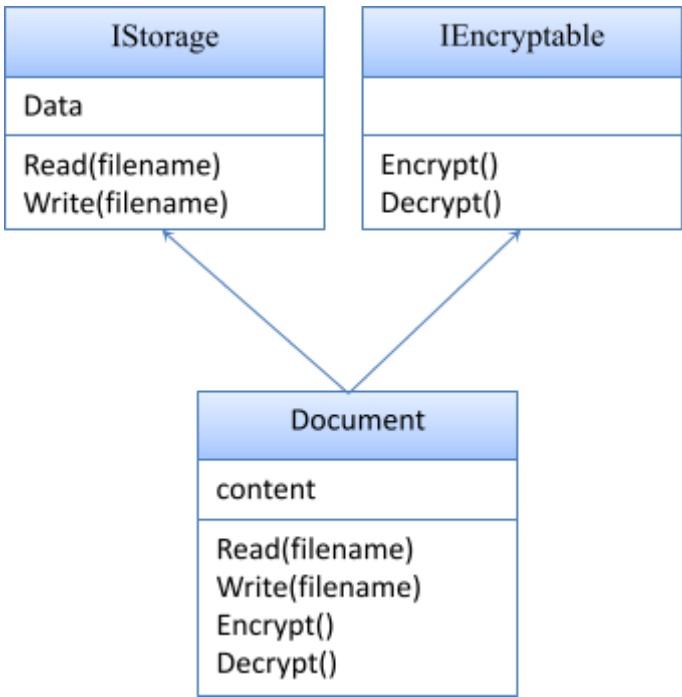
Bài 4.2

Tạo ứng dụng C# minh họa lớp abstract và kế thừa như hình sau.



Bài 4.3

Tạo ứng dụng C# minh họa interface như sau.



Yêu cầu: Tạo một giao diện tên là `IStoreable` với các phương thức `Write()` để lưu nội dung của đối tượng vào file và phương thức `Read()` để đọc dữ liệu từ file. Sau đó tạo lớp `Document` thực thi giao diện `IStoreable` để các đối tượng thuộc lớp này có thể đọc từ cơ sở dữ liệu hoặc lưu trữ vào cơ sở dữ liệu. Việc mở file được thực hiện thông qua đối tượng `fs` thuộc lớp `FileStream`, việc ghi và đọc file thông qua đối tượng thuộc các lớp `StreamWriter` và `StreamReader`. Đồng thời lớp `Document` cũng thực thi một giao diện khác tên là `IEncryptable`, giao diện này có hai phương thức là mã hóa (`Encrypt()`) và giải mã (`Decrypt()`). Các phương thức được thực thi ở `Document` chỉ cần in ra thông báo tương ứng.

Bài 4.4

Implement a class named **Person** and two sub classes of Person named **Student** and **Employee**. Make **Faculty** and **Staff** sub classes of Employee. A Person has a name, phone number and email address. A student has a program to which he/she enrolled (Business, Computer Science...) . An Employee has a department, salary and the date hired. A faculty member has office hours and a rank. A staff member has a title. You are required to

- Override the **ToString()** to display the class name and the person's name and email address.
- Provide properties in each class to read and write it's fields
- Define a **CalculateBonus** and **CalculateVacation** as abstract methods in Employee class and implement them in Faculty and Staff as follows
 - o Faculty get $1000 + 0.05 \times \text{Salary}$ and Staff get $0.06 \times \text{Salary}$
 - o Faculty get 5 weeks if they are employed more than 3 years and additional one week if he/she is

"Senior Lecturer". Otherwise 4 weeks. Staff get 4 weeks for 5 year service. Otherwise get 3 weeks

HẾT