

## SO SÁNH PHƯƠNG PHÁP LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG TRONG JAVA VÀ C#

- Tính Đóng Gói** : Java và C# có cùng khai báo bổ từ truy cập( private, public, protect) , constructor ( mặc định và có tham số) . Nhưng trong java để truy cập đến các thành phần của đối tượng phải thông qua các phương thức get – set, trong khi đó C# dùng các properties .

### 1.1 . Java

```
public class HìnhChuNhat{
    private float chieuDai;
    private float chieuRong;

    public HìnhChuNhat() {
        this.chieuDai = 0;
        this.chieuRong = 0;
    }

    public HìnhChuNhat(float chieuDai, float chieuRong) {
        this.chieuDai = chieuDai;
        this.chieuRong = chieuRong;
    }

    public float getChieuDai() {
        return chieuDai;
    }

    public void setChieuDai(float chieuDai) {
        this.chieuDai = chieuDai;
    }

    public float getChieuRong() {
        return chieuRong;
    }

    public void setChieuRong(float chieuRong) {
        this.chieuRong = chieuRong;
    }
}
```

### 1.2 . C#

```

public class HìnhChuNhat
{
    private float chieuDai;
    private float chieuRong;

    public HìnhChuNhat()
    {
        this.chieuDai = 0;
        this.chieuRong = 0;
    }
    public HìnhChuNhat(float chieuDai, float chieuRong)
    {
        this.chieuDai = chieuDai;
        this.chieuRong = chieuRong;
    }
    public float ChieuDai
    {
        get { return chieuDai; }
        set { chieuDai = value; }
    }

    public float ChieuRong
    {
        get { return chieuRong; }
        set { chieuRong = value; }
    }
}

```

## 2. Tính Kế Thừa : Java sử dụng từ khoá extends , C# sử dụng dấu “:”

### 2.1 .Java

```

public class HìnhHoc
{
    protected float chuVi;
    protected float dienTich;
    protected void In() {
        System.out.println("Chu vi:"+chuVi);
        System.out.println("Dien tich:"+dienTich);
    }
}

```

```

public class HìnhChuNhat extends HìnhHoc {
    private float chieuDai;
    private float chieuRong;
    public HìnhChuNhat() {...}
    public HìnhChuNhat(float chieuDai, float chieuRong) {...}
    public float getChieuDai() {...}
    public void setChieuDai(float chieuDai) {...}
    public float getChieuRong() {...}
    public void setChieuRong(float chieuRong) {
        this.chieuRong = chieuRong;
    }
    public void tinhChuViDienTich() {
        //chuVi va dienTich ke thua tu lop hình học
        chuVi = (chieuDai + chieuRong) * 2;
        dienTich = chieuDai * chieuRong;
    }
}

```

## 2.2 . C#

```

public class HìnhHoc
{
    protected float chuVi;
    protected float dienTich;
    protected void In() {
        Console.WriteLine("Chu vi:"+chuVi);
        Console.WriteLine("Dien tich:"+dienTich);
    }
}
public class HìnhChuNhat:HìnhHoc
{
    private float chieuDai;
    private float chieuRong;

    public HìnhChuNhat()...
    public HìnhChuNhat(float chieuDai, float chieuRong)...
    public float ChieuDai...
    public float ChieuRong...
    public void tinhChuViDienTich() {
        //chuVi va dienTich ke thua tu lop hình học
        chuVi = (chieuDai + chieuRong) * 2;
        dienTich = chieuDai * chieuRong;
    }
}

```

### 3. Tính Trừu Tượng :

#### 3.1. Java : sử dụng từ khoá abstract

```
public abstract class HìnhHoc
{
    protected float chuVi;
    protected float dienTich;
    //Khai báo phương thức trừu tượng
    abstract public void In() ;
}

public class HìnhChuNhat extends HìnhHoc {
    private float chieuDai;
    private float chieuRong;
    public HìnhChuNhat() {...}
    public HìnhChuNhat(float chieuDai, float chieuRong) {...}
    public float getChieuDai() {...}
    public void setChieuDai(float chieuDai) {...}
    public float getChieuRong() {...}
    public void setChieuRong(float chieuRong) {...}
    public void tinhChuViDienTich() {...}
    //thực thi phương thức trừu tượng
    public void In() {
        System.out.println("Chu vi:"+chuVi);
        System.out.println("Dien tich:"+dienTich);
    }
}
```

#### 3.2. C# : sử dụng từ khoá abstract và override

```

public abstract class HìnhHoc
{
    protected float chuVi;
    protected float dienTich;
    abstract public void In();
}
public class HìnhChuNhat:HìnhHoc
{
    private float chieuDai;
    private float chieuRong;

    public HìnhChuNhat()...
    public HìnhChuNhat(float chieuDai, float chieuRong)...
    public float ChieuDai...
    public float ChieuRong...
    public void tinhChuViDienTich() ...
    public override void In()
    {
        Console.WriteLine("Chu vi:" + chuVi);
        Console.WriteLine("Dien tich:" + dienTich);
    }
}

```

#### 4. Tính Đa Hình : Java và C# giống nhau về cú pháp

##### 4.1. Java

```

public abstract class HìnhHoc
{
    protected float chuVi;
    protected float dienTich;
    //Khai bao phuong thuc truu tuong
    abstract public void In() ;
}

```

```

public class HìnhChuNhat extends HìnhHoc {
    private float chieuDai;
    private float chieuRong;
    public HìnhChuNhat() {...}
    public HìnhChuNhat(float chieuDai, float chieuRong) {...}
    public float getChieuDai() {...}
    public void setChieuDai(float chieuDai) {...}
    public float getChieuRong() {...}
    public void setChieuRong(float chieuRong) {...}
    public void tinhChuViDienTich() {...}
    //thuc thi phuong thuc truu tuong
    public void In() {
        System.out.println("Chu vi:"+chuVi);
        System.out.println("Dien tich:"+dienTich);
    }
}

public class Main {
    public static void main(String[] args) {
        HìnhChuNhat hcn = new HìnhChuNhat(1,2);
        hcn.tinhChuViDienTich();
        //goi phuog thuc In() tu doi tuong lop HìnhHoc
        HìnhHoc hh = hcn;
        hh.In();
    }
}

```

## 4.2. C#

```

public abstract class HìnhHoc
{
    protected float chuVi;
    protected float dienTich;
    abstract public void In();
}
public class HìnhChuNhat:HìnhHoc
{
    private float chieuDai;
    private float chieuRong;

    public HìnhChuNhat()...
    public HìnhChuNhat(float chieuDai, float chieuRong)...
    public float ChieuDai...
    public float ChieuRong...
    public void tinhChuViDienTich() ...
    public override void In()
    {
        Console.WriteLine("Chu vi:" + chuVi);
        Console.WriteLine("Dien tich:" + dienTich);
    }
}
class Program
{
    static void Main(string[] args)
    {
        HìnhChuNhat hcn = new HìnhChuNhat(1, 2);
        hcn.tinhChuViDienTich();
        //gọi phuong thuc In() tu doi tuong lop HìnhHoc
        HìnhHoc hh = hcn;
        hh.In();
    }
}

```

- 5. Đa Thừa Kế :** Cả Java và C# không hỗ trợ đa thừa kế ( một lớp được kế thừa từ nhiều lớp khác ), chỉ hỗ trợ “đa thừa kế” thông qua interface

#### 5.1. Java: Sử dụng từ khoá implement

```

public interface IMath {
    int Add(int a, int b);
}

```

```
public class MyMath implements IMath
{
    public int Add(int a, int b) {
        return a+b;
    }
}
public class Main {
    public static void main(String[] args) {
        IMath i = new MyMath();
        System.out.println(i.Add(2,3));
    }
}
```

## 5.2. C# : Sử dụng dấu ":"

```
public interface IMath
{
    int Add(int a, int b);
}
public class MyMath:IMath
{
    public int Add(int a, int b) {
        return a+b;
    }
}
class Program
{
    static void Main(string[] args)
    {
        IMath i = new MyMath();
        Console.WriteLine(i.Add(2,3));
    }
}
```

## 6. Khai báo lớp không thể kế thừa

### 6.1. Java : Sử dụng từ khoá final

```
public final class HìnhHoc
{
    protected float chuVi;
    protected float dienTich;
}
```

### 6.2. C# : Sử dụng từ khoá sealed



```
public sealed class HìnhHoc
{
    protected float chuVi;
    protected float dienTich;
}
```

7. Ngoài các đặc điểm trên , từ phiên bản 3.0 trở lên C# hỗ trợ thêm các tính năng sau :

### 7.1 Object Initialization

Khởi tạo đối tượng kiểu object Initialize gồm có 3 cách sau:

- Cách thông thường: Khởi tạo các đối tượng sau đó gán các thuộc tính.

Ví dụ:

```
Diem dinhA = new Diem();
dinhA.X = 10;
dinhA.Y = 10;
```

- Gán các thuộc tính ngay khi khởi tạo đối tượng:

Ví dụ:

```
Diem dinhB = new Diem() { X = 10, Y = 20 };
```

- Khởi tạo đối tượng với kiểu anonymous:

Ví dụ:

Object Initializers với anonymous types:

```
var dinhC = new { X = 9, Y = 11};
```

### 7.2 Anonymous Type

Anonymous Type - kiểu dữ liệu trừu tượng - được dùng khi khai báo đối tượng chưa xác định được kiểu. Kiểu dữ liệu của biến sẽ được xác định khi gán giá trị cụ thể cho biến.

```

class Program
{
    static void Main(string[] args)
    {
        var product = new { ProductID = "P001",
                             ProductName = "Coffee",
                             UnitPrice = 10.5f,
                             Quantity = 5
                           };

        Console.WriteLine("ProductID = {0}",
                           product.ProductID);

        Console.WriteLine("ProductName = {0}",
                           product.ProductName);
        Console.WriteLine("UnitPrice = {0}",
                           product.UnitPrice);
        Console.WriteLine("Quantity = {0}",
                           product.Quantity);

        //Console.WriteLine("Change Quantity to 20 ");
        //product.Quantity = 20; //error : read only

        Console.ReadLine();
    }
}

```

### 7.3 Automatic Properties

Nhằm đơn giản hóa việc phải định nghĩa các get/set giống nhau ở các properties, sử dụng automatic properties cho phép người dùng khai báo một cách chung chung get; set;. Thay vào đó trình biên dịch có thể tự động tạo ra các private field và những thao tác get/set mặc định.

```

class KhachHang
{
    //Khoi tạo các thành phần dữ liệu thông qua Automatic properties
    public KhachHang()
    {
        MaKhachHang = "KH0001";
        TenKhachHang = "ABC";
        Tuoi = 18;
    }
    //Khai báo các Automatic properties
    public string MaKhachHang { get; set; }
    public string TenKhachHang { get; set; }
    public int Tuoi { get; set; }
}

class Program {
    static void Main(string[] args)
    {
        //Tạo đối tượng Khách Hàng
        KhachHang c1 = new KhachHang();
        //Xuất các giá trị mặc định
        Console.WriteLine("Ma Khach Hang : {0}", c1.MaKhachHang);
        Console.WriteLine("Ten Khach Hang: {0}", c1.TenKhachHang);
        Console.WriteLine("Tuoi : {0}", c1.Tuoi);
        Console.WriteLine("-----*****-----");
        //Thay đổi các giá trị mới
        c1.MaKhachHang = "KH002";
        c1.TenKhachHang = "Tommy";
        c1.Tuoi = 30;
        //Xuất các giá trị sau khi thay đổi
        Console.WriteLine("Ma Khach Hang : {0}", c1.MaKhachHang);
        Console.WriteLine("Ten Khach Hang: {0}", c1.TenKhachHang);
        Console.WriteLine("Tuoi : {0}", c1.Tuoi);
        Console.ReadLine();
    }
}

```