

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG TRONG JAVA

1. **Tính Đóng Gói** : Java khai báo bổ từ truy cập(private, public, protect) , constructor (mặc định và có tham số) và để truy cập đến các thành phần của đối tượng phải thông qua các phương thức getter – setter .

```
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;
    }
}
```

2. **Tính Kế Thừa** : Java sử dụng từ khoá extends

```

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;
    }
}

```

3. Tính Trừu Tượng: 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() {...}
    //thuc thi phuong thuc truu tuong
    public void In() {
        System.out.println("Chu vi:"+chuVi);
        System.out.println("Dien tich:"+dienTich);
    }
}

```

4. Tính Đa Hình :

```

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();
    }
}

```

5. Interface : 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));
    }
}

```

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;
}

```