

Bài Học

Kỹ thuật viết Kế thừa từ Class



Nội dung bài học

- ❖ Lớp cơ sở (base class): làm cơ sở để các lớp khác kế thừa
- ❖ Lớp nhận (derived class): kế thừa đặc điểm của lớp cơ sở
- ❖ Khai báo
 - `class LopNhan extends LopCoSo`
 - Ví dụ:
 - `class XeDap extends PhuongTienGiaoThong`
 - `class XeMay extends PhuongTienGiaoThong`

Constructor

- ❖ Không được kế thừa
- ❖ Lớp con truy cập bằng từ khóa **super**

```
class HìnhHoc
{
    ...
    public HìnhHoc(double chuVi, double dienTich)
    {
        ChuVi = chuVi;
        DienTich = dienTich;
    }
}
class HìnhTron extends HìnhHoc
{
    ...
    public HìnhTron(double chuVi, double dienTich) {
        super(chuVi, dienTich);
    }
}
```

Abstract class

- ❖ Lớp Abstract là một lớp trừu tượng, không thể khởi tạo nó bằng toán tử **new**.
- ✓ Trong lớp có tồn tại phương thức abstract thì lớp đó cũng được định nghĩa là abstract.
- ✓ Phương thức trừu tượng là phương thức chỉ có định nghĩa tên hàm, các đối số (không có nội dung hàm)
- ✓ Bất cứ class nào kế thừa abstract class nào đó phải định nghĩa lại các abstract methods của lớp mà nó thừa kế hoặc không định nghĩa lại nhưng phải ghi lại abstract methods đó.



Abstract class

```
abstract class A{  
    abstract void method_1();  
}  
public class B extends A{  
    @Override  
    public void method_1(){  
        // cài đặt chi tiết cho phương thức method_1  
        // trong lớp con B.  
    }  
}  
public class C extends A{  
    @Override  
    public void method_1(){  
        // cài đặt chi tiết cho phương thức method_1  
        // trong lớp con C.  
    }  
}
```

END

