KẾ THỪA
INHERITANCE

tailieusharefree.blogspot.com

1. Đặt vấn đề:

VẤN ĐỀ

- Giả sử phần mềm của bạn cần quản lý thông tin của những đối tượng Sinh Viên, Giáo Viên, Nhân Viên của một trường đại học.
- Các đối tượng này có những thuộc tính chung ví dụ như tên, tuổi, ngày sinh, địa chỉ.

Vậy để có thể tránh được việc dư thừa code và tái sử dụng phần mềm thì hướng giải quyết là gì ?

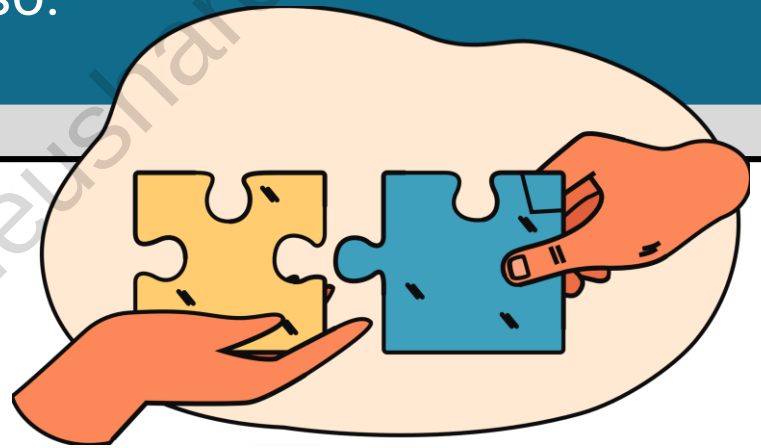
tailieushare.free.blogspot.com

1. Đặt vấn đề:

GIẢI PHÁP

Xây dựng một lớp cơ sở (base class) chứa các thuộc tính chung của 3 lớp này, sau đó cho các lớp này kế thừa từ lớp cơ sở.

Khi đó ta cần bổ sung các thuộc tính, phương thức của từng lớp dẫn xuất (derived class).



tailieusharefree.blogspot.com

Lớp trong Java có thể mở rộng, tạo một lớp mới từ một lớp cũ mà vẫn bảo toàn được những đặc điểm của lớp cũ. Quá trình này gọi là kế thừa, kế thừa liên quan tới các khái niệm như lớp cha (base class hoặc super class), lớp con (derived class hoặc sub class).

2. Cú pháp kế thừa:

CÚ PHÁP

```
public class SubClass extends SuperClass{}
```

Khi lớp con kế thừa lớp cha, nó có đầy đủ các thuộc tính và phương thức của lớp cha, ngoài ra bạn có thể bổ sung thêm thuộc tính và phương thức cần thiết cho lớp con.

Tuy nhiên nếu các thuộc tính của lớp cha là private bạn cũng ko thể truy cập vào các thuộc tính này từ lớp con. Nếu muốn thì bạn khai báo phạm vi truy cập là protected.

tailieusharefree.blogspot.com

2. Cú pháp kế thừa:

Ví dụ:

```
public class Person {
    private String name, birth;
    public Person(String name, String birth){
        this.name = name;
        this.birth = birth;
    }
    public String getName(){
         return this.name
    }
}

public class Main {
    public static void main(String[] args) {
        Student s = new Student("CNTT1", 3.2, "Nguyen Van A", "22/12/2002");
        System.out.println(s.getName());
    }
}
```

```
public class Student extends Person{
    private String lop;
    private double gpa;
    public Student(String lop, double gpa, String name,
                                                String birth){
        super(name, birth);
        this.lop = lop;
        this.gpa = gpa;
    }
}
```

OUTPUT

Nguyen Van A

tailieusharefree.blogspot.com

3. Từ khóa super:

Khi muốn gọi các phương thức của lớp cha tại lớp con ta có thể sử dụng **từ khóa super.**

Gọi PTKT của lớp cha từ lớp con

```
public Student(String lop, double gpa, String name, String birth)
{
        super(name, birth);
        this.lop = lop;
        this.gpa = gpa;
}
```

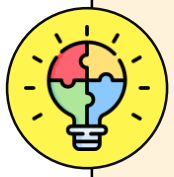
tailieusharefree.blogspot.com

Gọi PT toString của lớp cha từ lớp con

```
public class Person {
    private String name, birth;
    public Person(String name, String birth) {
        this.name = name;
        this.birth = birth;
    }
    @Override
    public String toString(){
        return this.name + " " + this.birth;
    }
}
public class Student extends Person{
    private String lop;
    private double gpa;
    public Student(String lop, double gpa, String name, String birth) {
        super(name, birth);
        this.lop = lop;
        this.gpa = gpa;
    }
    public String toString(){
        return super.toString() + " " + this.lop + " " + String.format("%.2f", this.gpa);
    }
}
```

taileusharetree.blogspot.com

4. Overriding:



Ghi đè trong kế thừa là khi ở lớp cha và lớp con có một phương thức giống nhau. Nếu lớp con không ghi đè phương thức này thì phương thức của lớp cha sẽ được gọi khi bạn gọi phương thức này từ đối tượng của lớp con.


```java
public class Person {
    public void greet(){
        System.out.println("Person !");
    }
}
public class Student extends Person{
     public void greet(){
        System.out.println("Student !");
    }
}
```

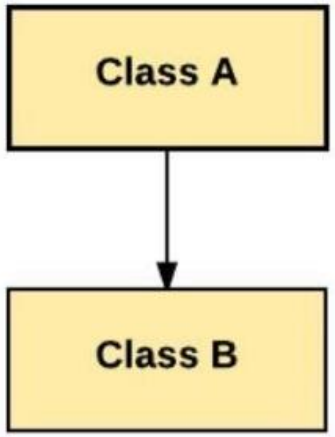
```java
public class Main {
    public static void main(String[] args){
        Student s = new Student();
        s.greet();
    }
}
```

OUTPUT

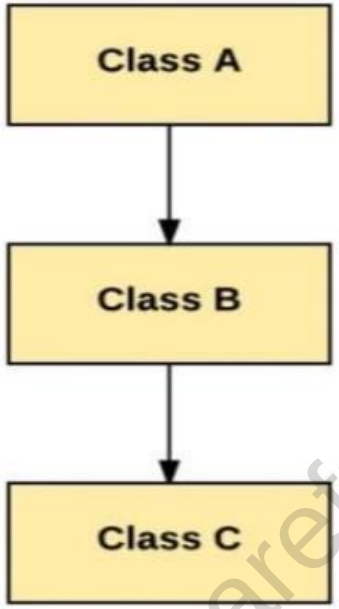
```
Student !
```

tailieushare.blogspot.com

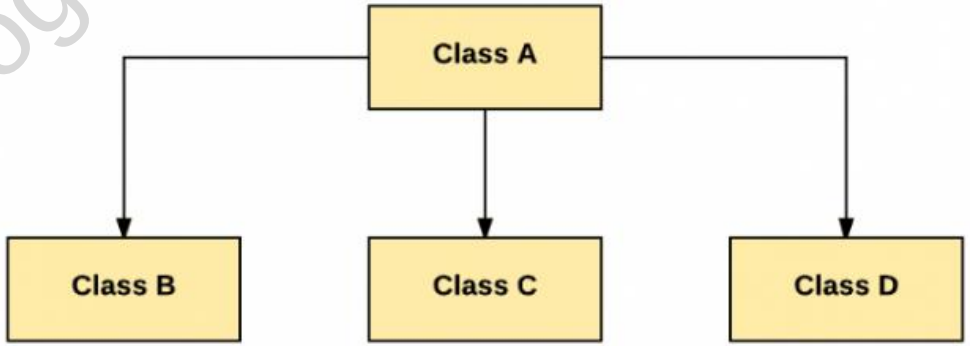
5. Các loại kế thừa:



Kế thừa đơn
(Single Inheritance)



Kế thừa nhiều mức
(Multi-level Inheritance)



Kế thừa thứ bậc
(Hierarchical Inheritance)

tailieuchiasefree.blogspot.com

5. Các loại kế thừa:


```
public class Person {
    public String name, birth;
    public Person(String name, String birth) {
        this.name = name;
        this.birth = birth;
    }
}
public class Student extends Person{
    private String lop;
    private double gpa;
    public Student(String lop, double gpa, String name, String birth) {
        super(name, birth);
        this.lop = lop;
        this.gpa = gpa;
    }
}
public class Pupil extends Student{
    private String fatherName;
    public Pupil(String fatherName, String lop, double gpa, String name, String birth) {
        super(lop, gpa, name, birth);
        this.fatherName = fatherName;
    }
}
```

tailieusharefree.blogspot.com