HOA SEN UNIVERSITY
WORLD CLASS EDUCATION

Bài 8

Thừa kế
(Inheritance)

HOA SEN UNIVERSITY
WORLD CLASS EDUCATION

Nội dung

- Thừa kế (Inheritance)
- Phương thức Overriding và Hiding
- Từ khóa super
- Final Classes và Methods

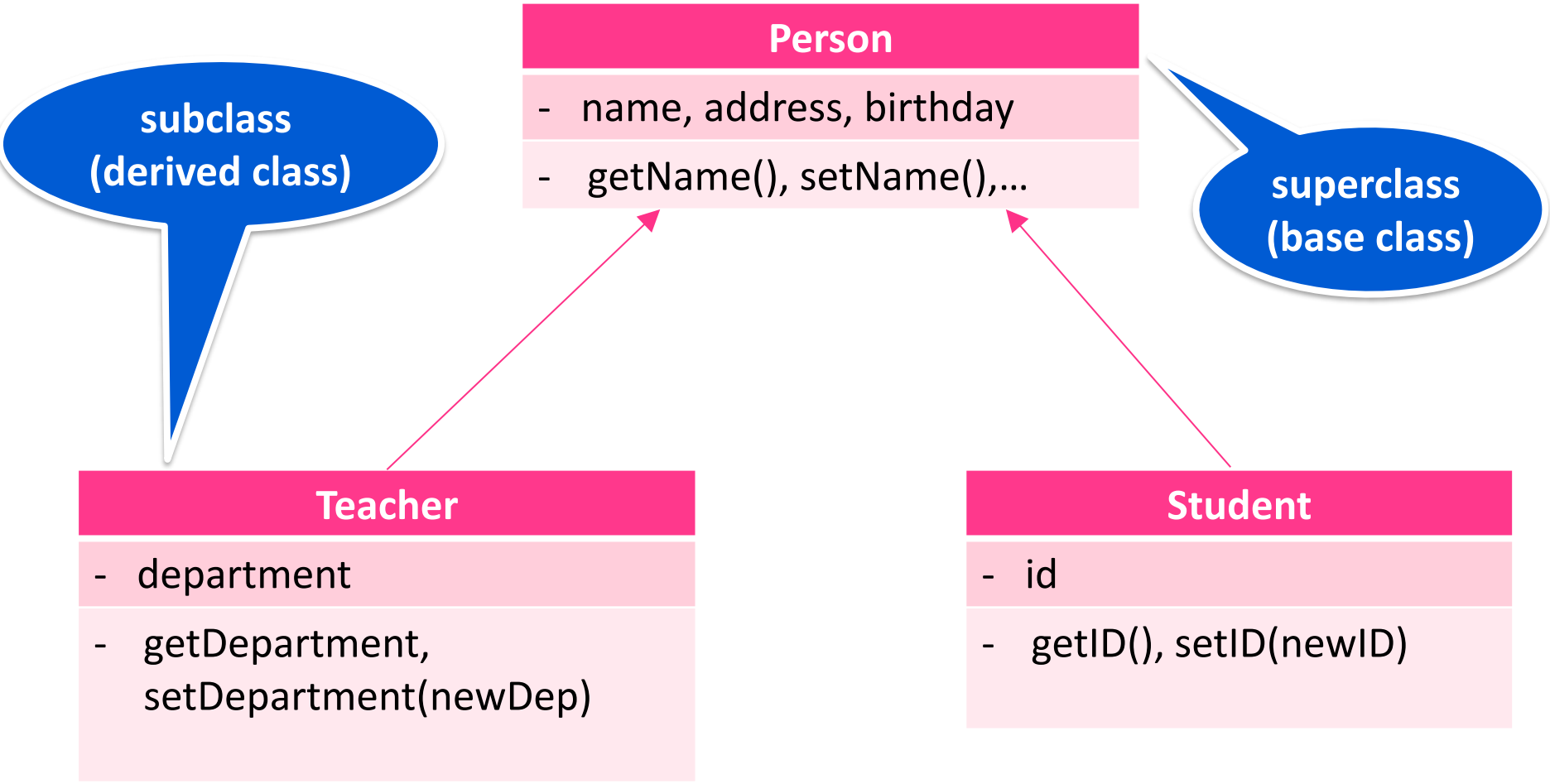
HOA SEN UNIVERSITY
WORLD CLASS EDUCATION

Inheritance

Teacher	Student
- name, address, birthday	- name, address, birthday
- department	- id
- getName(), setName(),…	- getName(), setName(),…
- getDepartment, setDepartment(newDep)	- getID(), setID(newID)

HOA SEN UNIVERSITY
WORLD CLASS EDUCATION

Inheritance



HOA SEN UNIVERSITY
WORLD CLASS EDUCATION

Implementing

```
24
25  class Person {
26      String name, address;
27      void intput(){
28          //input name and address from user
29      }
30  }
31  class Student extends Person{
32      int id;
33      void inputID(){
34          //input id from user
35      }
36      void output(){
37          System.out.println(id+"-"+name+"-"+address);
38      }
39  }
```

```
public static void main(String[] args) {
    Student stud = new Student();
    stud.intput();
    stud.inputID();
    stud.output();
}
```

The inherited fields/methods can be used directly

HOA SEN UNIVERSITY
WORLD CLASS EDUCATION

Protected Members trong Superclass

- A subclass does not inherit the private members of its parent class.
- *Protected members* (fields and methods) of superclass only be accessed by a subclass of its

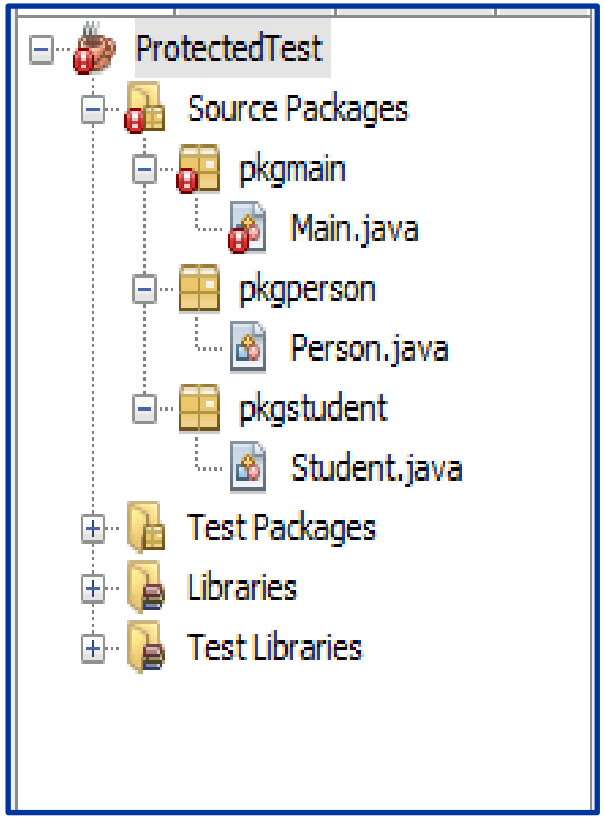
HOA SEN UNIVERSITY
WORLD CLASS EDUCATION

Access Levels

Modifier	Inside class	Inside package	Subclass-Outside package	Anywhere
private	✓			
No modifier *package-private*	✓	✓		
protected	✓	✓	✓	
public	✓	✓	✓	✓

HOA SEN UNIVERSITY
WORLD CLASS EDUCATION

Protected Members Demo

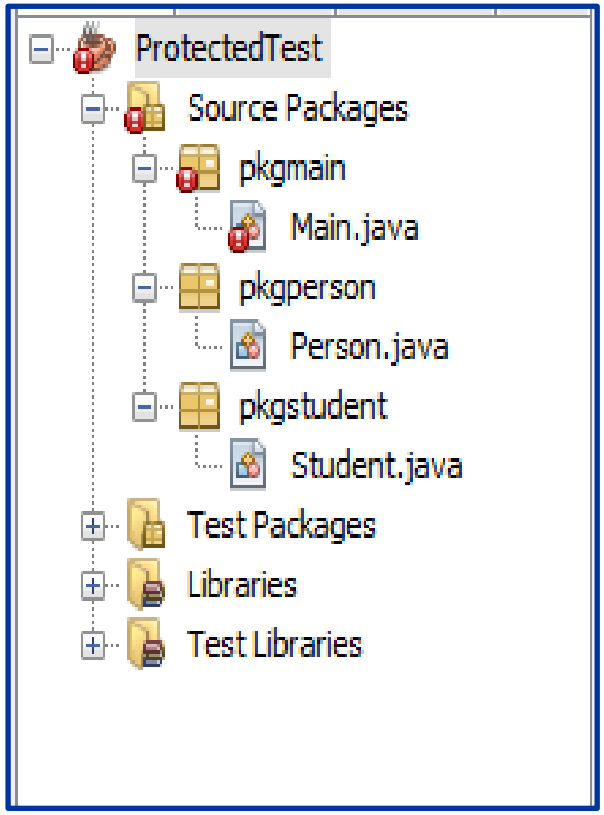



```java
package pkgperson;

public class Person {
    protected String name, address;
    protected void intput(){
        //input name and address from user
    }
}
```

```java
package pkgstudent;

import pkgperson.Person;

public class Student extends Person {
    int id;
    public void inputID(){
        //input id from user
    }
    public void output(){
        System.out.println(id+"-"+name+"-"+address);
    }
}
```

HOA SEN UNIVERSITY
WORLD CLASS EDUCATION

Protected Members Demo




```
6   package pkgmain;
7
8   import pkgstudent.Student;
9
10  public class Main {
11      public static void main(String[] args) {
12          Student stud = new Student();
13
14
15
            stud.intput();
17          stud.inputID();
18          stud.output();
19      }
20  }
```

intput() has protected access in Person

(Alt-Enter shows hints)

Subclass constructor

```
class Person{
    String name, address;

    public Person(String name, String address) {
        this.name = name;
        this.address = address;
    }
}

class Student extends Person{
    int id;

    public Student(int id, String name, String address) {
        super(name, address);
        this.id = id;
    }
}
```

*Invoke constructor of superclass using the keyword **super***

HOA SEN UNIVERSITY
WORLD CLASS EDUCATION

Subclass constructor

- Constructors không được thừa kế
- Khi tạo đối tượng của lớp con thì đối tượng của lớp cha phải được tạo trước
- Cho nên constructor của lớp con phải chỉ định constructor của lớp cha
- Nếu không chỉ định thì constructor không tham số của lớp cha sẽ được gọi

HOA SEN UNIVERSITY
WORLD CLASS EDUCATION

Subclass constructor

```
class Person{
    String name, address;

    public Person(String name, String address) {
        this.name = name;
        this.address = address;
    }
}

constructor Person in class Person cannot be applied to given types;
  required: String,String
  found: no arguments
  reason: actual and formal argument lists differ in length
----
(Alt-Enter shows hints)

    public Student(int id, String name, String address) {
        //super(name, address);
        this.id = id;
    }
}
```

Invoke constructor of superclass ***implicitly***

Output?

null-null
123-Tom-HSU

```
class Person{
    String name, address;
    public Person() {
        System.out.println(name+"-"+address);
    }
}
class Student extends Person{
    int id;
    public Student(int id, String name, String address) {
        this.id = id;
        this.name = name;
        this.address = address;
    }
    void output(){
        System.out.println(id+"-"+name+"-"+address);
    }
}
```

HOA SEN UNIVERSITY
WORLD CLASS EDUCATION

Overriding and Hiding Methods

- You can write a new *instance* method in the subclass that has the same signature and return type as the one in the superclass, thus *overriding* it.
- You can write a new *static* method in the subclass that has the same signature as the one in the superclass, thus *hiding* it.

HOA SEN UNIVERSITY
WORLD CLASS EDUCATION

Overriding Methods

```
class Person{
    String name, address;
    void output(){
        System.out.print(name + "-" + address);
    }
}

class Student extends Person{
    int id;
    void output(){
        System.out.print(id + "-");
        super.output();
    }
}
```

Phương thức của lớp con định nghĩa lại phương thức của lớp cha, có cùng tên và kiểu trả về

HOA SEN UNIVERSITY
WORLD CLASS EDUCATION

@Override annotation

```
class Person {
    public void output(){
    }
}
class Student extends Person{
    @Override
    public void output(){
    }
}
```

HOA SEN UNIVERSITY
WORLD CLASS EDUCATION

Hiding Methods

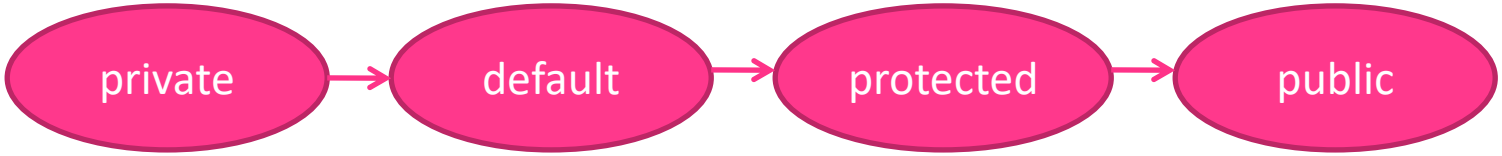
```
class Person{
    public void output(){
    }
    public static void input(){
    }
}

class Student extends Person{
    @Override
    public void output(){
    }

    // Hiding
    public static void input(){
    }
}
```

Lớp cha và lớp con định nghĩa phương thức tĩnh cùng tên và kiểu trả về

HOA SEN UNIVERSITY
WORLD CLASS EDUCATION

Modifiers for overriding methods

- The access specifier for an overriding method (of subclass) must be **more access than** the overridden method (of superclass)




```
 8    class Person {
 9        public void output(){
10        }
11
12        protected void input(){
13        }
14    }
```

```
17    class Student extends Person{
19        protected void output(){
20        }
22        void input(){
23        }
    }
```

Final Classes and Methods

- *final* method: the method **cannot be overridden** by subclasses
- *final* class: the class **can not have sub-class**
- *final* data is a **constant**.

```
final public double PI = 3.14;
```

HOA SEN UNIVERSITY
WORLD CLASS EDUCATION

final Methods Demo

```
27
    class Person{ //extends Object class implicitly
29      String name, address;
30      public Person(String name, String address) {
31          this.name = name;
32          this.address = address;
33      }
34      @Override
        public final String toString(){
36          return name+"-"+address;
37      }
38  }
39
40  class Student extends Person{
41      int id;
42      public Student(int id, String name, String address) {
43          super(name, address);
44          this.id = id;
45
46
47
        public String toString(){
49          return id+"-"+super.toString();
50      }
51  }
```

toString() in Student cannot override toString() in Person
overridden method is final

(Alt-Enter shows hints)

HOA SEN UNIVERSITY
WORLD CLASS EDUCATION

final Class Demo

```
    final class Person{ //extends Object class implicitly
29      String name, address;
30      public Person(String name, String address) {
31          this.name = name;
32          this.address = address;
33      }
34      @Override
        public String toString(){
36          return name+"-"+address;
37      }
38  }
39  class Student extends Person{
41      int id;
42      public Student(int id, String name, String address) {
43          super(name, address);
44          this.id = id;
45      }
46      @Override
        public String toString(){
48          return id+"-"+super.toString();
49      }
50  }
```

cannot inherit from final Person

(Alt-Enter shows hints)

HOA SEN UNIVERSITY
WORLD CLASS EDUCATION

HỎI ĐÁP