

Week 11-2 : Class Hierarchy

Part3. Casting

- Change an object type

Upcasting : a subclass' object change to a superclass' object.

Downcasting : Changing an Upcasted object to original. Need to type defining explicitly.

```
class Person {
    String name;
    String id;

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

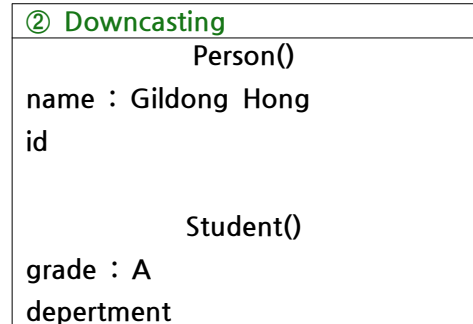
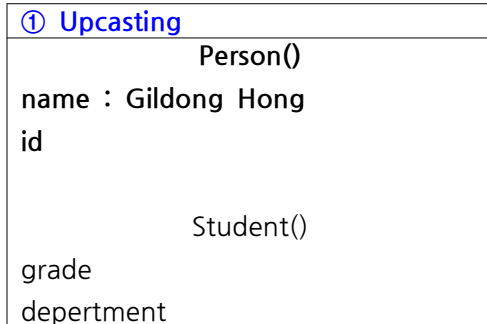
class Student extends Person {
    String grade;
    String department;

    public Student(String name) {
        super(name);
    }
}

public class Casting {
    public static void main(String[] args) {
        Person p = new Student("Gildong Hong"); // ① Upcasting
        System.out.println(p.name);
        p.grade = "A";
        p.department = "Computer";

        Student s = (Student)p; // ② Downcasting
        System.out.println(s.name);
        s.grade = "A";
    }
}
```

```
}  
}
```



Part4. Overriding

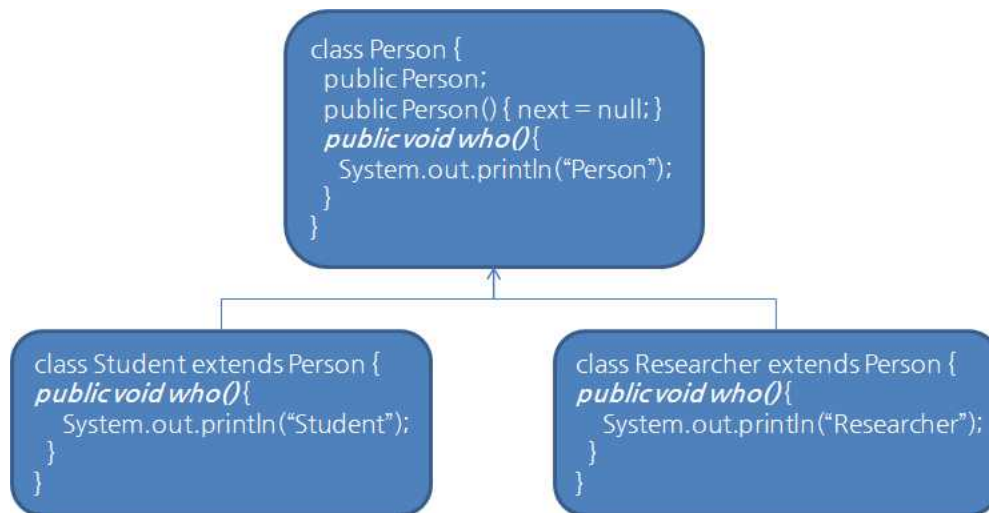
Relation between superclass and subclass' methods.

Rewriting a subclass' method which have same name with superclass'.

Possibly accessing superclass' member and method using 'super' keyword.

- Conditions of overriding

- ① Re-defining a subclass' method which is same with superclass'.
- ② Not possibly have a small scope than an access modifier of superclass' methods.
- ③ Not possibly have a different return type only.



```

public class Overriding {
    public static void main(String[] args) {
        Person p = new Person();
        Student st = new Student();
        Person p1 = new Researcher();
        Person p2 = st;

        p.who();
        st.who();
        p1.who();
        p2.who();
    }
}
  
```

☞ Result : Person / Student / Researcher / Student
 Student and Researcher's who() method is overriding. so, a subclass' method executes.

- Method overloading vs Method overriding

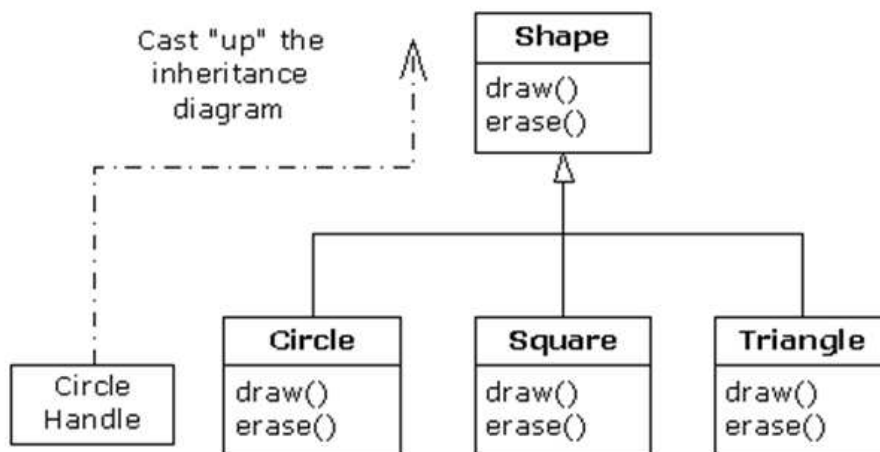
	Overloading	Overriding
Definition	Rewrite a same named method in a same class or an inheritance relationship	Rewriting a subclass' method which have same name with superclass'.
Relation	a same class or an inheritance relationship	an inheritance relationship

Goal	Improving using the several same named methods.	Re-defining a new method in a subclass.
Condition	a same named method, but a different name, number, or type of method's paramter.	Everything is same form except implementing.
Biding	Static binding	Dynamic binding

[Exercise]

1. Generate a new Student class inherited from a Person class. Make a overriding getName() method in the Student class using superclass. Re-define a new "Overriding" class including main function and print out same name value using a parent and child class. (a Person class has a name value and a student class has a name and id value)

2. This image shows inheritance relationship about Shape and its sub-classes. Make a code based on this structure. And print out each shape's draw method. Use up-casting in the main method.



Source : Lecture17, Thinking in java