WEEK 12

Controlling Accessibility

Nurnasran Puteh

Outline

- Inheritance revisited
- Accessibility Modifiers
- Four types of Java Accessibility Modifiers
- Accessibility Rule in Overriding Method

Learning Objectives

- To understand the use of accessibility modifiers related to inheritance.
- To explain all Java Accessibility Modifiers:
  - private, protected, public, default

Inheritance revisited

- In inheritance, a child class inherits all members (data and methods) of its parent class (except the parent class constructors).

- What if the parent wants only certain members (i.e. not all members) to be inherited by its child class?

- In general, a class can control which other classes can access its members.

Accessibility Modifiers

- Focusing on member-level access modifiers
- Parent can use accessibility modifiers to control which members that can be accessed by its child class.
- Parent class can do this by assigning the specific access modifier to its data and methods.
- Enforces the concept of encapsulation (data & method hiding)
- Also known as member-level visibility modifier

Java Accessibility Modifiers

There are 4 accessibility modifiers in Java

A parent class can assign its members as one of these 4 modifiers

- public
- private
- protected
- default (if no modifier is assigned)

Note: The keyword default is not a used in assigning default accessibility

Java Accessibility Modifiers (Member Level)

```
public class MyClass {
 //data members
 _____ int data;

 //method members
 _____ int method() {

 }
}
```

Any one of four modifiers below can be assigned here.

- public
- protected
- private
- (no modifier)

Example 1:

public

protected

private

```
public class MyClass {
 //data members
  private      int data;

 //method members
  public      int method() {

 }
}
```

data is assigned with private modifier

method is assigned with public modifier

Example 2:

public

protected

private

```
public class MyClass {
  //data members
    int data;

  //method members
  protected int method() {

  }
}
```

data is assigned with default modifier

method is assigned with protected modifier

Java Accessibility Modifiers

	Can be accessed by:				
Access Modifier	Same Class?	Class in Same Package?	Class in Different Package?	Child Class in Same Package?	Child Class in Different Package?
private	Yes	No	No	No	No
default	Yes	Yes	No	Yes	No
protected	Yes	Yes	No	Yes	Yes
public	Yes	Yes	Yes	Yes	Yes

Example: public Access Modifier

Class Person in package my.edu.oop1

All data & method members with **public** access modifiers

```
package my.edu.oop1;

public class Person {
    public String name;
    public int age;

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

    public void displayInfo() {
        System.out.println("Name = " + name);
        System.out.println("Age = " + age);
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }
```

Example: public Access Modifier

Class Contact in package my.edu.oop1 (same package with Person class)

Create Person object

Can access name
Can access method getAge()

Class in same package

```
package my.edu.oop1;

public class Contact {
    Person p;
    String phoneNo;

    public Contact(String name,int age, String phoneNo) {
        this.p = new Person(name,age);
        this.phoneNo = phoneNo;
    }
    public void displayInfo() {
        System.out.println("Name = " + p.name);
        System.out.println("Age = " + p.getAge());
        System.out.println("Phone No = " + phoneNo);
    }
}
```

Example: public Access Modifier

Class Contact in package my.edu.oop2 (different package with Person class)

Create Person object

Can access name
Can access method getAge()

Class in different package

```
package my.edu.oop2;

import my.edu.oop1.*;

public class Contact {
    Person p;
    String phoneNo;

    public Contact(String name,int age, String phoneNo) {
        this.p = new Person(name,age);
        this.phoneNo = phoneNo;
    }
    public void displayInfo() {
        System.out.println("Name = " + p.name);
        System.out.println("Age = " + p.getAge());
        System.out.println("Phone No = " + phoneNo);
    }
}
```

Example: public Access Modifier

Child Class in same package

Class Student in package my.edu.oop1 (same package with Person class)

Can access name

Can access method getAge()

```
package my.edu.oop1;

public class Student extends Person {

    int studID;
    double mark;

    public Student(String name, int age, int studID, double mark) {
        super(name, age);
        this.studID = studID;
        this.mark = mark;
    }

    @Override
    public void displayInfo() {
        System.out.println("Name = " + name);
        System.out.println("Age = " + getAge());
        System.out.println("Stud ID = " + studID);
        System.out.println("Mark = " + mark);
    }
```

Example: public Access Modifier

Class Student in package my.edu.oop2 (different package with Person class)

Can access name

Can access method getAge()

```java
package my.edu.oop2;

import my.edu.oop1.*;

public class Student extends Person {

    int studID;
    double mark;

    public Student(String name, int age, int studID, double mark) {
        super(name, age);
        this.studID = studID;
        this.mark = mark;
    }

    @Override
    public void displayInfo() {
        System.out.println("Name = " + name);
        System.out.println("Age = " + getAge());
        System.out.println("Stud ID = " + studID);
        System.out.println("Mark = " + mark);
    }
```

Example: private Access Modifier

Class Person in package my.edu.oop1

All data members with **private** access modifiers

Method getAge() with **private** access modifiers

```
package my.edu.oop1;

public class Person {
    private String name;
    private int age;

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

    public void displayInfo() {
        System.out.println("Name = " + name);
        System.out.println("Age = " + age);
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public String getName() {
        return name;
    }

    private int getAge() {
        return age;
    }
```

Example: private Access Modifier

Class in same package

```
package my.edu.oop1;

public class Contact {
    Person p;
    String phoneNo;

    public Contact(String name,int age, String phoneNo) {
        this.p = new Person(name,age);
        this.phoneNo = phoneNo;
    }
    public void displayInfo() {
        System.out.println("Name = " + p.name);
        System.out.println("Age = " + p.getAge());
        System.out.println("Phone No = " + phoneNo);
    }
```

Class Contact in package my.edu.oop1 (same package with Person class)

Cannot access name

Cannot access method getAge()

Example: private Access Modifier

Class in different package

Class Contact in package my.edu.oop2 (different package with Person class)

Cannot access name

Cannot access method getAge()

```
package my.edu.oop2;

import my.edu.oop1.*;

public class Contact {
    Person p;
    String phoneNo;

    public Contact(String name,int age, String phoneNo) {
        this.p = new Person(name,age);
        this.phoneNo = phoneNo;
    }
    public void displayInfo() {
        System.out.println("Name = " + p.name);
        System.out.println("Age = " + p.getAge());
        System.out.println("Phone No = " + phoneNo);
    }
```

Example: private Access Modifier

Child Class in same package

```
package my.edu.oop1;

public class Student extends Person {

    int studID;
    double mark;

    public Student(String name, int age, int studID, double mark) {
        super(name, age);
        this.studID = studID;
        this.mark = mark;
    }

    @Override
    public void displayInfo() {
        System.out.println("Name = " + name);
        System.out.println("Age = " + getAge());
        System.out.println("Stud ID = " + studID);
        System.out.println("Mark = " + mark);
    }
```

Class Student in package my.edu.oop1 (same package with Person class)

Cannot access name

Cannot access method getAge()

Example: private Access Modifier

Child Class in different package

Class Student in package my.edu.oop2 (different package with Person class)

Cannot access name

Cannot access method getAge()

```
package my.edu.oop2;

import my.edu.oop1.*;

public class Student extends Person {

    int studID;
    double mark;

    public Student(String name, int age, int studID, double mark) {
        super(name, age);
        this.studID = studID;
        this.mark = mark;
    }

    @Override
    public void displayInfo() {
        System.out.println("Name = " + name);
        System.out.println("Age = " + getAge());
        System.out.println("Stud ID = " + studID);
        System.out.println("Mark = " + mark);
    }
```

Example: default Access Modifier

Class Person in package my.edu.oop1

All data members with **default** access modifiers

Method getAge() with **default** access modifiers

```
package my.edu.oop1;

public class Person {
    String name;
    int age;

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

    public void displayInfo() {
        System.out.println("Name = " + name);
        System.out.println("Age = " + age);
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public String getName() {
        return name;
    }

    int getAge() {
        return age;
    }
```

Example: default Access Modifier

Class in same package

Class Contact in package my.edu.oop1 (same package with Person class)

Can access name

Can access method getAge()

```
package my.edu.oop1;

public class Contact {
    Person p;
    String phoneNo;

    public Contact(String name,int age, String phoneNo) {
        this.p = new Person(name,age);
        this.phoneNo = phoneNo;
    }
    public void displayInfo() {
        System.out.println("Name = " + p.name);
        System.out.println("Age = " + p.getAge());
        System.out.println("Phone No = " + phoneNo);
    }
```

Example: default Access Modifier

Class Contact in package my.edu.oop2 (different package with Person class)

Cannot access name

Cannot access method getAge()

Class in different package

```
package my.edu.oop2;

import my.edu.oop1.*;

public class Contact {
    Person p;
    String phoneNo;

    public Contact(String name,int age, String phoneNo) {
        this.p = new Person(name,age);
        this.phoneNo = phoneNo;
    }
    public void displayInfo() {
        System.out.println("Name = " + p.name);
        System.out.println("Age = " + p.getAge());
        System.out.println("Phone No = " + phoneNo);
    }
```

Example: default Access Modifier

Class Student in package my.edu.oop1 (same package with Person class)

Can access name

Can access method getAge()

Child Class in same package

```
package my.edu.oop1;

public class Student extends Person {

    int studID;
    double mark;

    public Student(String name, int age, int studID, double mark) {
        super(name, age);
        this.studID = studID;
        this.mark = mark;
    }

    @Override
    public void displayInfo() {
        System.out.println("Name = " + name);
        System.out.println("Age = " + getAge());
        System.out.println("Stud ID = " + studID);
        System.out.println("Mark = " + mark);
    }
```

Example: default Access Modifier

Class Student in package my.edu.oop2 (different package with Person class)

Cannot access name

Cannot access method getAge()

Child Class in different package

```
package my.edu.oop2;

import my.edu.oop1.*;

public class Student extends Person {

    int studID;
    double mark;

    public Student(String name, int age, int studID, double mark) {
        super(name, age);
        this.studID = studID;
        this.mark = mark;
    }

    @Override
    public void displayInfo() {
        System.out.println("Name = " + name);
        System.out.println("Age = " + getAge());
        System.out.println("Stud ID = " + studID);
        System.out.println("Mark = " + mark);
    }
```

Example: protected Access Modifier

Class Person in package my.edu.oop1

All data members with **protected** access modifiers

Method getAge() with **protected** access modifiers

```
package my.edu.oop1;

public class Person {
    protected String name;
    protected int age;

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

    public void displayInfo() {
        System.out.println("Name = " + name);
        System.out.println("Age = " + age);
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public String getName() {
        return name;
    }

    protected int getAge() {
        return age;
    }
```

Example: protected Access Modifier

Class in same package

```
package my.edu.oop1;

public class Contact {
    Person p;
    String phoneNo;

    public Contact(String name,int age, String phoneNo) {
        this.p = new Person(name,age);
        this.phoneNo = phoneNo;
    }
    public void displayInfo() {
        System.out.println("Name = " + p.name);
        System.out.println("Age = " + p.getAge());
        System.out.println("Phone No = " + phoneNo);
    }
```

Class Contact in package my.edu.oop1 (same package with Person class)

Can access name

Can access method getAge()

Example: protected Access Modifier

Class Contact in package my.edu.oop2 (different package with Person class)

Cannot access name

Cannot access method getAge()

Class in different package

```
package my.edu.oop2;

import my.edu.oop1.*;

public class Contact {
    Person p;
    String phoneNo;

    public Contact(String name,int age, String phoneNo) {
        this.p = new Person(name,age);
        this.phoneNo = phoneNo;
    }
    public void displayInfo() {
        System.out.println("Name = " + p.name);
        System.out.println("Age = " + p.getAge());
        System.out.println("Phone No = " + phoneNo);
    }
```

Example: protected Access Modifier

Class Student in package my.edu.oop1 (same package with Person class)

Can access name

Can access method getAge()

Child Class in same package

```
package my.edu.oop1;

public class Student extends Person {

    int studID;
    double mark;

    public Student(String name, int age, int studID, double mark) {
        super(name, age);
        this.studID = studID;
        this.mark = mark;
    }

    @Override
    public void displayInfo() {
        System.out.println("Name = " + name);
        System.out.println("Age = " + getAge());
        System.out.println("Stud ID = " + studID);
        System.out.println("Mark = " + mark);
    }
```

Example: protected Access Modifier

Child Class in different package

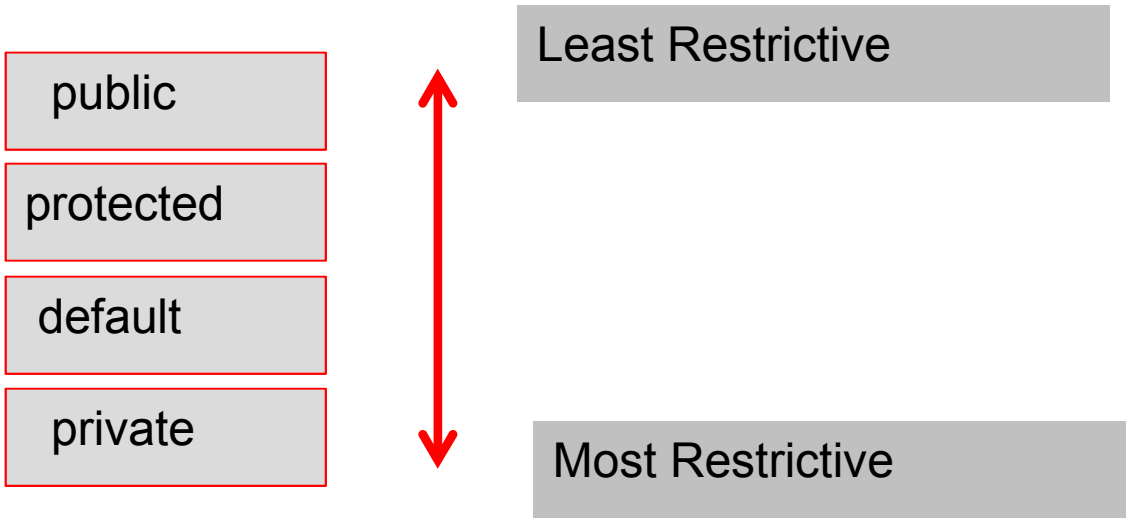
Class Student in package my.edu.oop2 (different package with Person class)

Can access name

Can access method getAge()

```
package my.edu.oop2;

import my.edu.oop1.*;

public class Student extends Person {

    int studID;
    double mark;

    public Student(String name, int age, int studID, double mark) {
        super(name, age);
        this.studID = studID;
        this.mark = mark;
    }

    @Override
    public void displayInfo() {
        System.out.println("Name = " + name);
        System.out.println("Age = " + getAge());
        System.out.println("Stud ID = " + studID);
        System.out.println("Mark = " + mark);
    }
```

Java Accessibility Modifiers: Restrictiveness Ranking



Accessibility Rule in Method Overriding

If a child class is overriding any of its parent's method, the overriding method in the child class must not be more restrictive.

(Remember the accessibility restrictiveness ranking in the previous slide)

Example (Refer to Person & Student classes)

The Person class displayInfo() method:

```
public void displayInfo() {
    System.out.println("Name = " +getName());
    System.out.println("Age = " +getAge());
}
```

Error! Because protected is more restrictive than public.

The Student class displayInfo() method:

```
protected void displayInfo() {
    super.displayInfo();
    System.out.println("Stud ID = " + getID());
    System.out.println("Mark = " + getMark());
}
```

Summary

- A class can control its members accessibilities by using the accessibility modifiers
- Accessibility modifiers can enforce encapsulation through data & method hiding
- There are 4 access modifiers in Java which are public, private, protected and default.
- public modifier is the most unrestricted modifier, followed by protected modifier, and then followed by default modifier. The most restricted one is private modifier
- An overriding method in child class cannot be more restrictive than the parent method.