

# 安阳工学院 实验报告

---

实验项目名称： 类的重用

(所属课程： JAVA 语言程序设计)

院 系：经济管理学院 专业班级：12 信息管理 姓 名：马高飞、张廷立

崔颢 学 号：15073730121、15073730120、15073730138

实验日期： 实验地点：11 号楼 206 合作者： 指导教师：张莉

本实验项目成绩： 教师签字： 日期：

## 一、实验目的

1. 理解继承的概念，了解其在程序设计中的作用；
2. 熟悉类的继承定义方法，熟练地由系统类或者自定义的类派生出子类，并熟知子类对继承自父类的成员的访问性；
3. 理解属性隐藏与方法覆盖（重写）的概念，并会调用父类中被隐藏的属性或被覆盖的方法；
4. 理解在子类中调用父类的构造方法的原则与方法。

## 二、实验条件

Windows2000 或 XP，JDK1.6 与 Eclipse

## 三、实验内容

- 1、编写继承的程序
- 2、编写方法重写的程序
- 3、编写运行属性隐藏的程序。
- 4、运行有继承是的构造方法的程序

# 安阳工学院 实验报告

---

## 四、实验步骤

1.下面有两段程序 Base.java 和 Sub.java，试着把它们放在相同的包中和不同的包中，观察程序有哪些不同。

Base.java

```
public class Base {  
  
    public int publicVarOfBase = 1;  
  
    private int privateVarOfBase =1;  
  
    int defaultVarofBase =1;  
  
    protected void methodOfBase(){  
  
    }  
}
```

Sub.java

```
public class Sub extends Base {  
  
    public void methodOfSub(){  
  
        publicVarOfBase = 2;  
  
        defaultVarofBase = 2;  
  
        //privateVarOfBase = 2;  
  
        methodOfBase();  
  
    }  
  
    public static void main(String args[]){
```

# 安阳工学院 实验报告

---

```
Sub sub = new Sub();

sub.publicVarOfBase = 3;

//sub.privateVarOfBase = 3;

sub.defaultVarofBase = 3;

sub.methodOfBase();

sub.methodOfSub();

}

}
```

2.运行并观察下面程序的结果，分析运行过程。

```
class A

{

    static int x = 2;

    public void setx(int i){

        x = i;

    }

    void printa()

    {

        System.out.println(x);

    }

}
```

# 安阳工学院 实验报告

---

```
class B extends A
```

```
{
```

```
    int x=100;
```

```
    void printb()
```

```
    {
```

```
        super.x = super.x +10 ;
```

```
        System.out.println
```

```
            ("super.x= " + super.x +
```

```
                "  x= " + x);
```

```
    } }
```

```
public class Exam4_4Test
```

```
{
```

```
    public static void main(String[] args)
```

```
    {
```

```
        A a = new A ();
```

```
        a.setx(4);
```

```
        a.printa();
```

```
        B b = new B();
```

```
        b.printb();
```

```
        b.printa();
```

```
        b.setx(6); // 将继承来的x值设置为6
```

```
        b.printb();
```

# 安阳工学院 实验报告

---

```
b.printa();

a.printa();

}

}
```

3. 编写程序实现如下功能：已知 Person 类包含三个公共域变量（姓名、性别、年龄）和一个构造方法，Student 类是 Person 类的派生类，包含两个新的公共域变量(学号、班号)、两个公共方法（构造方法、修改年龄、显示基本信息）和一个构造方法。定义一组学生对象，并初始化他们的基本信息，然后依次输出。

步骤（1）：定义类 Person，并添加以下公共域变量

```
public String name; public char sex; public int age;

添加构造方法 Person(String name,char sex,int age){……}
```

步骤（2）：基于类 Person 派生类 Student,并添加如下公共域变量和方法

```
public String classno; public String sno;

public void updateAge(int age){……}

void toString(){… …}

Student(String name,char sex,int age,String sno,String name)

{ super(name,sex,age);

… …

}
```

步骤（3）：定义一个包含 main 方法的 Application 程序类 TestStudentPerson，初始化一个学生类对象数组,并利用循环语句显示所有学生基本信息。

4. 观察下面程序的实验结果，分析构造方法的调用过程

```
class Meal {}

class Bread {}

class Cheese {}
```

# 安阳工学院 实验报告

---

```
class Lettuse {}

class Lunch extends Meal {}

class PortableLunch extends Lunch {}

class Sandwich extends PortableLunch {

    Bread b = new Bread();

    Cheese c = new Cheese();

    Lettuse l = new Lettuse();

    public static void main(String [] args){

        new Sandwich();

    }

}
```

5. 根据下面的程序构造“Teacher”类，继承person类。要求：

- ( 1 ) 增加“职称(String)”属性
- ( 2 ) 具有和“Student”类累死的重载构造方法
- ( 3 ) 重写“Person”类的info()方法，增加职称信息。

```
class Person{

    private String name;

    private String location;

    Person(String name){

        this.name = name;
```

# 安阳工学院 实验报告

---

```
        location = "beijing";

    }

    Person(String name,String location){

        this.name = name;

        this.location = location;

    }

    public String info(){

        return "name:" + name + "location:" + location;

    }

}

class Student extends Person {

    private String school;

    Student (String name,String school){

        this(name,school,"beijing");

    }

    Student (String n,String school, String l){

        super(n,l);

        this.school = school;

    }

    public String info(){

        return super.info() + "school:" + school;

    }

}
```

# 安阳工学院 实验报告

---

```
}
```

```
public class TestConstruct {  
  
    public static void main(String[] args) {  
  
        Person p1 = new Person("A");  
  
        Person p2 = new Person("B","shanghai");  
  
        Student s1 = new Student("C","s1");  
  
        Student s2 = new Student("D","s2","shanghai");  
  
        System.out.println(p1.info());  
  
        System.out.println(p2.info());  
  
        System.out.println(s1.info());  
  
        System.out.println(s2.info());  
  
    }  
}
```

6.请在下面代码的基础上增加 Bird 类，并且体现 Bird 类的 enjoy 方法。

```
class Animal{  
  
    private String name;  
  
    Animal (String name){  
  
        this.name = name;  
  
    }  
  
    public void enjoy(){  
  
        System.out.println("叫声");  
  
    }  
}
```



# 安阳工学院 实验报告

---

```
    }  
}  
  
class Cat extends Animal{  
    private String eyescolor;  
  
    Cat3(String n, String c){  
        super(n);  
        eyescolor = c;  
    }  
  
    public void enjoy(){  
        System.out.println("猫叫声");  
    }  
}  
  
class Dog extends Animal{  
    private String furcolor;  
  
    Dog3(String n, String c){  
        super(n);  
        furcolor = c;  
    }  
  
    public void enjoy(){  
        System.out.println("狗叫声");  
    }  
}
```

# 安阳工学院 实验报告

---

```
class Lady {  
  
    private String name;  
  
    private Animal pet;  
  
    Lady (String name, Animal pet){  
  
        this.name = name;  
  
        this.pet = pet;  
  
    }  
  
    public void myPetEnjoy(){  
  
        pet.enjoy();  
  
    }  
  
}  
  
public class TestBinding {  
  
    public static void main(String[] args) {  
  
        Cat c = new Cat ("catname","blue");  
  
        Dog d = new Dog ("dogname","black");  
  
        Lady l1 = new Lady("l1",c);  
  
        Lady l2 = new Lady("l2",d);  
  
        l1.myPetEnjoy();  
  
        l2.myPetEnjoy();  
  
    }  
  
}
```

7.试运行下面的程序，观察实验结果，分析实验过程。

# 安阳工学院 实验报告

---

```
class Shape{

    void draw(){

    }

    void erase(){

    }

}

class Circle extends Shape{

    @Override

    void draw() {

        System.out.println("Circle.draw()");

    }

    @Override

    void erase() {

        System.out.println("Circle..erase()");

    }

}

class Square extends Shape{

    @Override

    void draw() {

        System.out.println("Square.draw()");

    }

    @Override
```

# 安阳工学院 实验报告

---

```
void erase() {  
  
    System.out.println("Square.erase()");  
  
}  
  
}  
  
class Triangle extends Shape{  
  
    @Override  
  
    void draw() {  
  
        System.out.println("Triangle.draw()");  
  
    }  
  
    @Override  
  
    void erase() {  
  
        System.out.println("Triangle.erase()");  
  
    }  
  
}  
  
public class BindingTester {  
  
    public static void main(String[] args) {  
  
        Shape[] shape = new Shape[9];  
  
        int n;  
  
        for(int i = 0; i<shape.length;i++){  
  
            n = (int)(Math.random()*3);
```

# 安阳工学院 实验报告

---

```
switch(n){  
  
    case 0:  
  
        shape[i] = new Circle();  
  
        break;  
  
    case 1:  
  
        shape[i] = new Square();  
  
        break;  
  
    case 2:  
  
        shape[i] = new Triangle();  
  
        break;  
  
    }  
  
}  
  
for (Shape oneS:shape){  
  
    oneS.draw();  
  
}  
  
}
```

8.观察下面程序的实验结果，试比较动态绑定与静态绑定的区别。

```
class Horse {  
  
    public void running(){  
  
        System.out.println("Horse running()");  
  
    }  
  
}
```

# 安阳工学院 实验报告

---

```
public static void walk(){

    System.out.println("Horse walk()");

}

}

class WhiteHorse extends Horse {

    public void running(){

        System.out.println("WhiteHorse running()");

    }

    public static void walk(){

        System.out.println("WhiteHorse walk()");

    }

}

public class MethodHidden {

    public static void main(String[] args) {

        Horse horse = new WhiteHorse();

        horse.running();

        horse.walk();

        WhiteHorse whiteHorse = (WhiteHorse)horse;

        whiteHorse.running();

        whiteHorse.walk();

    }

}
```

# 安阳工学院 实验报告

---

9. 运行下面的程序，试着在 Cat 重写 equals 方法。并给出核心代码。

```
public class TestEquals {

    public static void main(String[] args) {

        Cat c1 = new Cat(1,2,3);

        Cat c2 = new Cat(1,2,3);

        Cat c3 = c1;

        System.out.println(c1==c2);

        System.out.println(c1.equals(c2));

        System.out.println(c3==c1);

    }

}

class Cat{

    int color;

    int height,weight;

    public Cat(int color, int height, int weight) {

        super();

        this.color = color;

        this.height = height;

        this.weight = weight;

    }

}
```

# 安阳工学院 实验报告

---

```
}
```

```
}
```

10.运行 p116 的例 4-10 观察并分析实验结果。

## 五、实验步骤

1. Error: (19, 12) java: defaultVarofBase 在 Assignment3.Base 中不是公共的; 无法从外部程序包中对其进行访问

两个类的被访问权限不同

2. 运行结果是:

4

super. x= 14    x= 100

14

super. x= 16    x= 100

16

16

3. 

```
public class TestStudentPerson {  
  
    public static void main(String args[]){  
  
        Student a = new Student();
```



# 安阳工学院 实验报告

---

```
a.classno="100";a.sno="101";a.name="zhanghan";
```

```
System.out.print(a.classno);
```

```
}
```

```
}
```

```
class Person{    //构造函数 Person
```

```
    public String name; public char sex; public int age; //公共域,姓名性别年
```

```
    龄
```

```
    public void Person(String name,char sex,int age){    //构造方法
```

```
}
```

```
}
```

```
class Student extends Person{
```

```
    public void Student(String name,char sex, int age,String sno,String
```

```
classno){
```

```
        //super(name,sex,age);语句错误
```

```
        super.Person(name, sex, age);
```

```
}
```

```
    public String classno;public String sno;    //班号,学号
```

```
    public void updateAge(int age){                //构造方法,更新年龄
```

# 安阳工学院 实验报告

---

```
}
```

```
@Override
```

```
public String toString() {
```

```
    return super.toString();
```

```
}
```

```
}
```

## 4. 执行顺序为

a) Meal

b) Lunch

c) PortableLunch

d) Bread

e) Cheese

f) Lettuse

## 5. `class Teacher extends Person{`     //构造 Teacher 类

```
    private String zhicheng;
```

```
    Teacher(String name,String zhicheng){
```

# 安阳工学院 实验报告

---

```
        this(name,zhicheng,"zhicheng");
    }

    Teacher(String n,String S,String zhicheng){

        super(n,S);

        this.zhicheng=zhicheng;

    }

    public String info(){

        return super.info()+"zhicheng"+zhicheng;

    }

}

6.    class Bird extends Animal{

        private String birdcolor;

        Bird(String n,String birdcolor){

            super(n);

            birdcolor=birdcolor;

        }

        public void enjoy(){System.out.print("鸟叫声");}

    }
```

## 7. 运行结果为

Triangle.draw()

Square.draw()

# 安阳工学院 实验报告

---

Circle.draw()

Triangle.draw()

Triangle.draw()

Circle.draw()

Triangle.draw()

Triangle.draw()

Circle.draw()

## 8. 实验结果

WhiteHorse running()

Horse walk()

WhiteHorse running()

WhiteHorse walk()

## 9.     public boolean equals(Object o){

        Cat c=(Cat)o;

        return (c.color==color) && (c.height==height)&&(c.weight==weight);}

# 安阳工学院 实验报告

---

六、讨论

七、参考文献