



# 南昌大学实验报告

学生姓名：\_\_\_\_\_ 学 号： 8003119100 专业班级： 信息安全 193  
班\_\_\_\_\_

实验类型： ☒ 验证 ☐ 综合 ☐ 设计 ☐ 创新 实验日期： 10.30 实验成绩： \_\_\_\_\_

## 一、实验项目名称

有关子类和父类继承调用函数结果的实验  
用类模拟小狗和小猫行为的实验

## 二、实验目的

熟练掌握 java 语言中，类的继承与类中的函数调用。

## 三、实验任务

完成教材 141 页的四个例题的调试和测试，分析程序运行结果和数据。  
编程题：设计一个动物声音模拟器，希望模拟器可以模拟许多动物的叫声。

## 四、主要仪器设备及耗材

IntelliJ IDEA 环境软件

java version "14.0.2" 2020-07-14

## 五、实验步骤

实验一、调试和查看结果有关子类和父类继承调用函数结果的实验

第一个程序:

```
package com.exercise;

class A{
    double f(double x,double y){
        return x+y;
    }
}

class B extends A{
    double f(int x,int y){
        return x*y;
    }
}

public class E {
    public static void main(String[] args) {

        B b = new B();
        System.out.println(b.f(x: 3, y: 5));
        System.out.println(b.f(x: 3.0, y: 5.0));
    }
}
```

E:\JAVA\JDK\bin\java.exe -Didea.launcher.port=50091 "-  
15.0  
8.0

第二个程序:

```
package com.exercise;

class A{
    public int getNumber(int a){
        return a+1;
    }
}

public class B extends A{
    public int getNumber(int a){
        return a+100;
    }

    public static void main(String[] args) {
        A a =new A();
        System.out.println(a.getNumber( a: 10)); //调用父类的getNumber() 函数
        a = new B();
        System.out.println(a.getNumber( a: 10)); //调用子类重写的getNumber() 函数
    }
}
```

```
E x
E:\JAVA\JDK\bin\java.exe -Didea.launcher.port=50091
15.0
8.0

Process finished with exit code 0
```

第三个程序:

```
class A1 {
    double f(double x, double y) {
        return x+y;
    }
    static int g(int n) {
        return n*n;
    }
}

class B1 extends A1 {
    double f(double x, double y) {
        double m = super.f(x, y);
        return m+x*y;
    }
    static int g(int n) {
        int m = A1.g(n);
        return m+n;
    }
}

public class E1 {
    public static void main(String[] args) {
        B1 b = new B1();
        System.out.println(b.f(x: 10.0, y: 8.0)); //调用B子类的double f()方法
        System.out.println(b.g(n: 3)); //调用子类的static int g(int n)方法
        A1 a = new B1();
        System.out.println(a.f(x: 10.0, y: 8.0)); //调用子类重写的double f()方法
        System.out.println(a.g(n: 3));
    }
}
```

```
E1 x
E:\JAVA\JDK\bin\java.exe -Didea.launcher.port=50126 "-I
98.0
12
98.0
9

Process finished with exit code 0
```

第四个程序:

```
class A2{
    int m;
    int getM(){
        return m;
    }
    int seeM(){
        return m;
    }
}

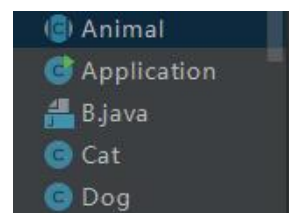
class B2 extends A2{
    int m;
    int getM(){
        return m+100;
    }
}

public class E2 {
    public static void main(String[] args) {
        B2 b = new B2();
        b.m=20;
        System.out.println(b.getM());
        A2 a = b;
        a.m=-100;    //上转型对象访问的是被隐藏的m
        System.out.println(a.getM()); //上转型对象调用的是子类重写的getM()方法
        System.out.println(b.seeM()); //子类继承的seeM()方法操作的m是被子类隐藏的m
    }
}
```

```
E:\JAVA\JDK\bin\java.exe -Didea.launcher.port=50131 "-Didea
120
120
-100

Process finished with exit code 0
```

实验二、编程题(动物模拟器)



有五个文件分别是 Animal.java, Application.java, Cat.java 和

Dog.java

Animal.java, Simulator.java 文件

```

package com.exercise;

public abstract class Animal {
    public abstract void cry();
    public abstract String getAnimalName();
}

```

Cat.java 文件

```

package com.exercise;

public class Cat extends Animal {
    public void cry() {
        System.out.println("喵喵");
    }
    public String getAnimalName() {
        return "猫";
    }
}

```

Dog.java 文件

```

1 package com.exercise;
2
3 public class Dog extends Animal {
4     public void cry() {
5         System.out.println("汪汪");
6     }
7     public String getAnimalName() {
8         return "狗";
9     }
10 }

```

Application.java

```

package com.exercise;

public class Application {
    public static void main(String[] args) {
        Simulator simutor = new Simulator();
        simutor.playSound(new Dog());
        simutor.playSound(new Cat());
    }
}

```

运行结果如下:

```

E:\JAVA\JDK\bin\java.exe -Didea.launcher.po
现在播放狗类的声音
汪汪
现在播放猫类的声音
喵喵

Process finished with exit code 0

```

## 六、实验数据及处理结果

### 实验一结果分析:

- 第一个程序中 B 类中重载了 double f()函数,当 b 调用函数时的运行结果视传入的参数而定。
- 第二个程序中 B 继承 A 类并重写了 getNumber()函数,当 A 类型的 a 调用它时自然是 A 类中函数,所以结果为 10+1=11; 当 a 变为 B 类的上转型对象时, 那么此时调用的就是子类重写的 getNumber()函数,所以结果为 100+10=110。
- 第三个程序中当 B1 类型的 b 调用 f()函数时,此时是指的子类的方法,而当定义一个 A1 a = new B1(), 此时 a 是 B 类的上转型对象, 则调用子类重写的 double f()方法
- 第四个程序中也是如此上转型对象调用的是子类的重写函数, 如果子类继承了父类的某个方法而没有重写或重载, 那么子类调用该函数

时操作的是父类的成员变量。

#### 实验二结果分析:

- 1、Animal 类是虚拟类，有两个虚拟方法 `public abstract void cry();`  
`public abstract String getAnimalName();`
- 2、Cat 和 Dog 类分别都继承了 Animal 类, 并实现了 `cry()` 和 `getAnimalName()` 方法;
- 3、Simulator 类中定义了 `playSound()` 方法，通过 Animal 类型的参数传递到其中并调用 Animal 的各个子类所实现的方法;
- 4、Application 类是主类，`public static void main()` 方法中定义一个 Simulator 类型的变量, 并调用 `playSound()` 方法, 实现程序.