

# 中国矿业大学计算机学院

## 2019 级本科生课程报告

课程名称 Java 语言与网络编程

学生姓名 李 春 阳

学 号 10193657

班 级 信息安全 2019-1 班

任课教师 张 爱 娟

报告时间 2021.12.5

## 目录

1. 问题一 .....	3
1.1 问题描述 .....	3
1.2 问题分析 .....	3
1.3 代码展示 .....	4
1.4 运行结果 .....	5
2. 问题二 .....	5
2.1 问题描述 .....	5
2.2 问题分析 .....	6
2.3 代码展示 .....	6
2.4 运行结果 .....	6
3. 问题三 .....	7
3.1 问题描述 .....	7
3.2 问题分析 .....	7
3.3 代码展示 .....	8
3.4 运行结果 .....	12

## 1. 问题一

### 1.1 问题描述

读程序，写结果：

```
1.  class A {
2.      public String Show(D obj) { return ("A and D"); }
3.      public String Show(A obj) { return ("A and A"); }
4.  }
5.  class B extends A {
6.      public String Show(B obj) { return ("B and B"); }
7.      public String Show(A obj) { return ("B and A"); }
8.  }
9.  class C extends B {
10.     public String Show(C obj) { return ("C and C"); }
11.     public String Show(B obj) { return ("C and B"); }
12. }
13. class D extends B {
14.     public String Show(D obj) { return ("D and D"); }
15.     public String Show(B obj) { return ("D and B"); }
16. }
17.
18. public class mainTest {
19.     public static void main(String args[]){
20.         A a1 = new A();
21.         A a2 = new B();
22.         B b = new B();
23.         C c = new C();
24.         D d = new D();
25.         System.out.println(a1.Show(b));
26.         System.out.println(a1.Show(c));
27.         System.out.println(a1.Show(d));
28.         System.out.println(a2.Show(b));
29.         System.out.println(a2.Show(c));
30.         System.out.println(a2.Show(d));
31.         System.out.println(b.Show(b));
32.         System.out.println(b.Show(c));
33.         System.out.println(b.Show(d));
34.     }
35. }
```

### 1.2 问题分析

#### 动态链编规则

程序运行并且采用动态绑定调用方法时，虚拟机必须调用与  $x$  所引用对象的实际类型对应的那个方法。假设  $x$  的实际类型是  $D$ ，它是  $C$  类的子类。如果  $D$  类定义了方法  $f(\text{String})$ ，就会调用这个方法；否则，将在  $D$  类的超类中寻找  $f(\text{String})$ ，以此类推。

调用方法时先在本身的类寻找方法，找不到则在父类中寻找。

## 寻找方法的顺序

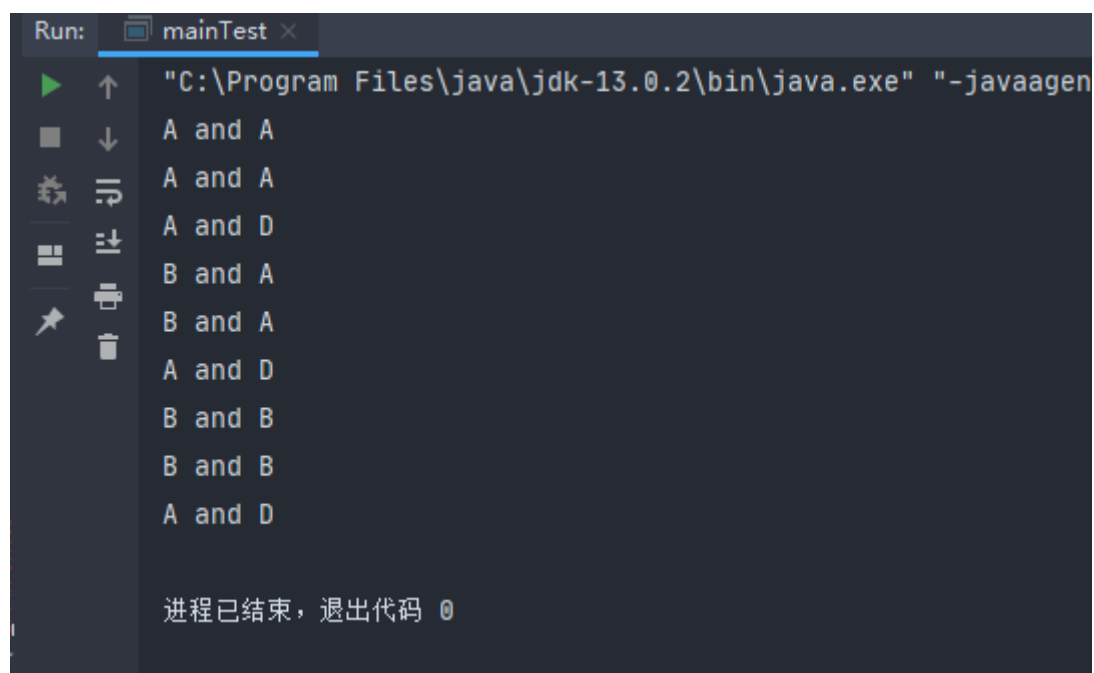
根据编译时类型:首先在本类中寻找方法,若未找到则到超类中寻找,若还是未找到,则根据向上转型,现在本类中寻找,未找到则在超类中寻找。

根据运行时类型:根据编译时确定的方法,在运行时类型的类中寻找匹配的方法,此时若该方法被继承则结果与执行父类的该方法一致,若该方法被重写则执行子类中重写的方法。

### 1.3 代码展示

```
36.  class A {
37.      public String Show(D obj) { return ("A and D"); }
38.      public String Show(A obj) { return ("A and A"); }
39.  }
40.  class B extends A {
41.      public String Show(B obj) { return ("B and B"); }
42.      public String Show(A obj) { return ("B and A"); }
43.  }
44.  class C extends B {
45.      public String Show(C obj) { return ("C and C"); }
46.      public String Show(B obj) { return ("C and B"); }
47.  }
48.  class D extends B {
49.      public String Show(D obj) { return ("D and D"); }
50.      public String Show(B obj) { return ("D and B"); }
51.  }
52.
53.  public class mainTest {
54.      public static void main(String args[]){
55.          A a1 = new A();
56.          A a2 = new B();
57.          B b = new B();
58.          C c = new C();
59.          D d = new D();
60.          System.out.println(a1.Show(b));
61.          System.out.println(a1.Show(c));
62.          System.out.println(a1.Show(d));
63.          System.out.println(a2.Show(b));
64.          System.out.println(a2.Show(c));
65.          System.out.println(a2.Show(d));
66.          System.out.println(b.Show(b));
67.          System.out.println(b.Show(c));
68.          System.out.println(b.Show(d));
69.      }
70.  }
```

## 1.4 运行结果



```
Run: mainTest x
"C:\Program Files\java\jdk-13.0.2\bin\java.exe" "-javaagen
A and A
A and A
A and D
B and A
B and A
A and D
B and B
B and B
A and D
进程已结束，退出代码 0
```

## 2. 问题二

### 2.1 问题描述

读程序，写结果：

```
1. class Base {
2.     private String name = "base";
3.     public Base() {
4.         tellName();}
5.
6.     public void tellName() {
7.         System.out.println("Base tell name: " + name); }
8. }
9. public class Dervied extends Base {
10.     private String name = "dervied";
11.     public Dervied() {
12.         tellName();}
13.
14.     public void tellName() {
15.         System.out.println("Dervied tell name: " + name);}
16.
17.     public static void main(String[] args){
18.
19.         new Dervied();
20.     }
21. }
```

## 2.2 问题分析

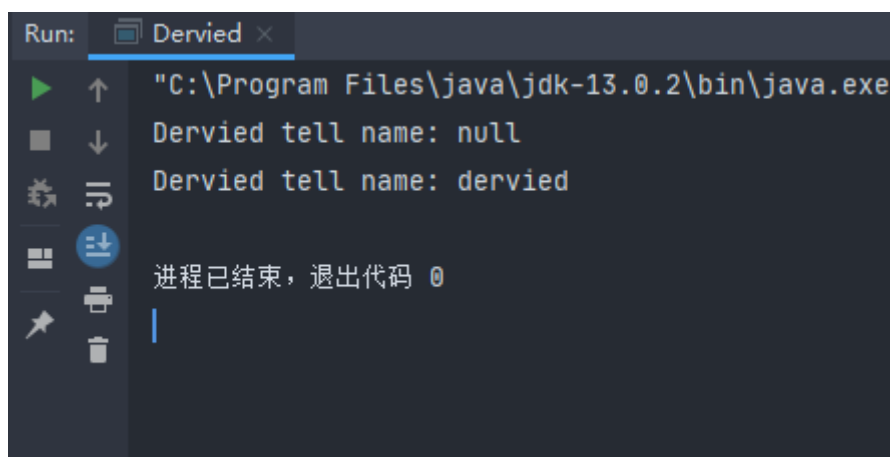
静态方法和静态变量、成员变量、构造函数，这里的成员变量和构造函数是指同一个类中的顺序，比如此题中先初始化父类的成员变量然后是父类构造函数，此时子类的成员变量还未初始化。

对于静态变量、静态初始化块、变量、初始化块、构造器，它们的初始化顺序依次是（静态变量、静态初始化块）>（变量、初始化块）>构造器。

## 2.3 代码展示

```
22. class Base {
23.     private String name = "base";
24.     public Base() {
25.         tellName();
26.
27.     public void tellName() {
28.         System.out.println("Base tell name: " + name); }
29. }
30. public class Dervied extends Base {
31.     private String name = "dervied";
32.     public Dervied() {
33.         tellName();
34.
35.     public void tellName() {
36.         System.out.println("Dervied tell name: " + name);}
37.
38.     public static void main(String[] args){
39.
40.         new Dervied();
41.     }
42. }
```

## 2.4 运行结果



```
Run: Dervied x
C:\Program Files\java\jdk-13.0.2\bin\java.exe
Dervied tell name: null
Dervied tell name: dervied
进程已结束，退出代码 0
```

### 3. 问题三

#### 3.1 问题描述

生成动物

1. 循环通过标准输入端输入需要生成的动物，当遇到结束标志，则结束程序运行。
2. 每次生成动物，通过标准输出端显示动物的信息。
3. 动物的信息包括：目前所有动物的总数，当前这一类动物的总数。
4. 整个程序结构用工厂模式设计，保证将来动物园有新的动物加入时，程序可扩展。

#### 3.2 问题分析

##### 工厂模式

工厂模式 (Factory Pattern) 是 Java 中最常用的设计模式之一。这种类型的设计模式属于创建型模式，它提供了一种创建对象的最佳方式。

在工厂模式中，我们在创建对象时不会对客户端暴露创建逻辑，并且是通过使用一个共同的接口来指向新创建的对象。

##### 介绍

意图：定义一个创建对象的接口，让其子类自己决定实例化哪一个工厂类，工厂模式使其创建过程延迟到子类进行。

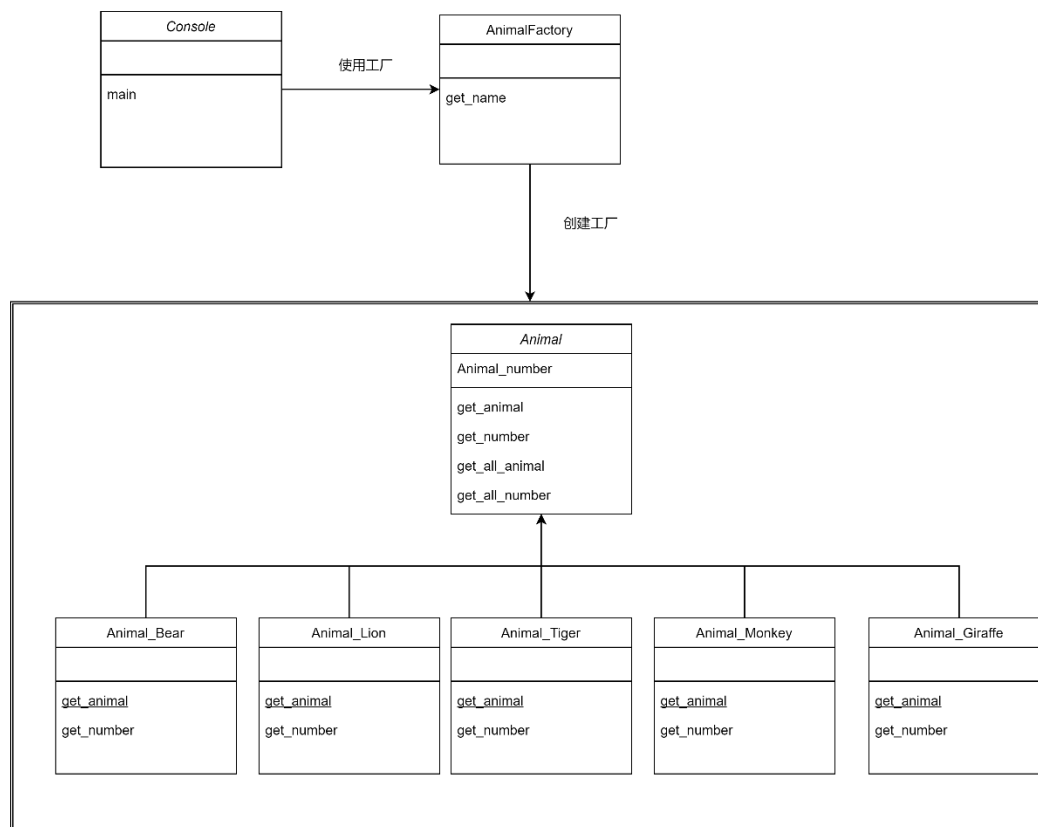
主要解决：主要解决接口选择的问题。

何时使用：我们明确地计划不同条件下创建不同实例时。

如何解决：让其子类实现工厂接口，返回的也是一个抽象的产品。

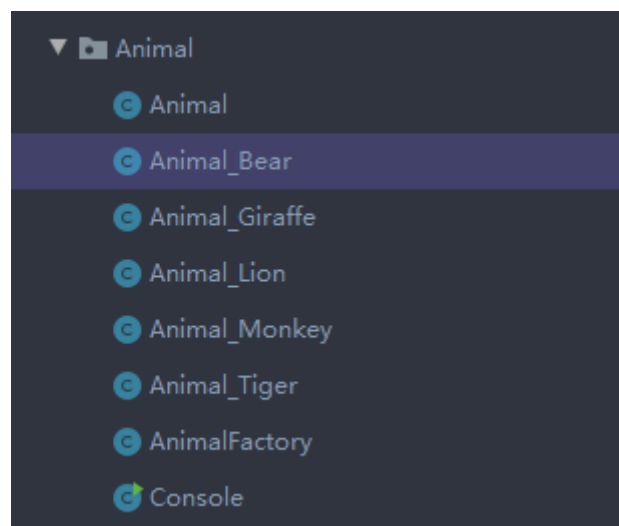
关键代码：创建过程在其子类执行。

下面是设计类图：



### 3.3 代码展示

**Animal** 包文件:



#### Console

```
1. package Animal;
2. import java.util.Scanner;
3.
4. public class Console {
5.     public static void main(String[] args) {
6.         Scanner scan = new Scanner(System.in);
```



```
7.         System.out.println("*****欢迎来到 lowly-li 的
动物园!*****");
8.         while (true) {
9.             System.out.println("请选择你的操作: \t[0]退出\t[1]添
加动物\t[2]查看全部动物");
10.            int option = scan.nextInt();
11.            if(option == 0){
12.                System.out.println("欢迎下次光临动物园");
13.                break;
14.            }
15.            else if(option == 1){
16.                //循环添加:
17.                while (true){
18.                    System.out.println("*****
*****");
19.                    System.out.println("请选择添加的动物: \t[0]结
束添加\t[1]熊\t[2]狮子\t[3]老虎\t[4]猴子\t[5]长颈鹿");
20.                    int animal_index = scan.nextInt();
21.                    if (animal_index == 0){
22.                        break;
23.                    }
24.                    Animal animal = AnimalFactory.get_name(anim
al_index);
25.                    if(animal != null){
26.                        System.out.println(animal.get_animal())
;
27.                        System.out.println(animal.get_number())
;
28.                    }
29.                    else{
30.                        System.out.println("其他动物还未引进, 请
重新输入");
31.                    }
32.                }
33.            }
34.            else if(option == 2){
35.                Animal animal = new Animal();
36.                System.out.println("动物园的动物有: ");
37.                System.out.println(animal.get_all_animal());
38.                animal.get_all_number();
39.            }
40.            else {
41.                System.out.println("输入错误, 请重新输入");
42.            }
43.            System.out.println();
44.            System.out.println("*****
*****");
45.        }
46.        scan.close();
47.    }
48. }
```

## AnimalFactory

```
1.  package Animal;
2.
3.  public class AnimalFactory {
4.      public static Animal get_name(int index){
5.          Animal animal = null;
6.          if(index == 1){
7.              animal = new Animal_Bear();
8.          }
9.          else if(index == 2){
10.             animal = new Animal_Lion();
11.          }
12.          else if(index == 3){
13.             animal = new Animal_Tiger();
14.          }
15.          else if(index == 4){
16.             animal = new Animal_Monkey();
17.          }
18.          else if(index == 5){
19.             animal = new Animal_Giraffe();
20.          }
21.          return animal;
22.      }
23.  }
```

## Animal

```
1.  package Animal;
2.
3.  public class Animal {
4.      static int bear=0,lion=0,tiger=0,monkey=0,giraffe=0;
5.      String get_animal(){
6.          return null;
7.      }
8.      String get_number(){
9.          return null;
10.     }
11.     String get_all_animal(){
12.         String names = "";
13.         if(bear > 0){
14.             names = names + "熊,";
15.         }
16.         if(lion > 0){
17.             names = names + "狮子,";
18.         }
19.         if(tiger > 0){
20.             names = names + "老虎,";
21.         }
22.         if(monkey > 0){
23.             names = names + "猴子,";
24.         }
25.         if(giraffe > 0){
26.             names = names + "长颈鹿,";
```

```
27.         }
28.         return names;
29.     }
30.     void get_all_number(){
31.         System.out.println("动物园的动物及其数量有: ");
32.         System.out.println("熊的数量:"+ bear);
33.         System.out.println("狮子的数量:"+ lion);
34.         System.out.println("老虎的数量:"+ tiger);
35.         System.out.println("猴子的数量:"+ monkey);
36.         System.out.println("长颈鹿的数量:"+ giraffe);
37.     }
38. }
```

## Animal\_Bear

```
1.     package Animal;
2.
3.     public class Animal_Bear extends Animal{
4.         public String get_animal(){
5.             return "这是一只熊";
6.         }
7.         public String get_number(){
8.             bear++;
9.             return "熊的数量:"+ bear;
10.        }
11.    }
```

## Animal\_Lion

```
1.     package Animal;
2.
3.     public class Animal_Lion extends Animal{
4.         public String get_animal(){
5.             return "这是一只狮子";
6.         }
7.         public String get_number(){
8.             lion++;
9.             return "狮子的数量:"+ lion;
10.        }
11.    }
```

## Animal\_Tiger

```
1.     package Animal;
2.
3.     public class Animal_Tiger extends Animal {
4.         public String get_animal(){
5.             return "这是一只老虎";
6.         }
7.         public String get_number(){
8.             tiger++;
9.             return "老虎的数量:"+ tiger;
10.        }
```

```
11.    }
```

## Animal\_Monkey

```
1.    package Animal;
2.
3.    public class Animal_Monkey extends Animal {
4.        public String get_animal(){
5.            return "这是一只猴子";
6.        }
7.        public String get_number(){
8.            monkey++;
9.            return "猴子的数量:"+ monkey;
10.       }
11.    }
```

## Animal\_Giraffe

```
1.    package Animal;
2.
3.    public class Animal_Giraffe extends Animal {
4.        public String get_animal(){
5.            return "这是一只长颈鹿";
6.        }
7.        public String get_number(){
8.            giraffe++;
9.            return "长颈鹿的数量:"+ giraffe;
10.       }
11.    }
```

## 3.4 运行结果

```
*****欢迎来到 lowly-li 的动物园!*****
请选择你的操作:  [0]退出 [1]添加动物[2]查看全部动物
1
*****
请选择添加的动物:  [0]结束添加[1]熊   [2]狮子 [3]老虎 [4]猴子 [5] 长 颈
鹿
1
这是一只熊
熊的数量:1
*****
请选择添加的动物:  [0]结束添加[1]熊   [2]狮子 [3]老虎 [4]猴子 [5] 长 颈
鹿
2
这是一只狮子
```

狮子的数量:1

\*\*\*\*\*

请选择添加的动物: [0]结束添加[1]熊 [2]狮子 [3]老虎 [4]猴子 [5] 长 颈  
鹿

3

这是一只老虎

老虎的数量:1

\*\*\*\*\*

请选择添加的动物: [0]结束添加[1]熊 [2]狮子 [3]老虎 [4]猴子 [5] 长 颈  
鹿

4

这是一只猴子

猴子的数量:1

\*\*\*\*\*

请选择添加的动物: [0]结束添加[1]熊 [2]狮子 [3]老虎 [4]猴子 [5] 长 颈  
鹿

5

这是一只长颈鹿

长颈鹿的数量:1

\*\*\*\*\*

请选择添加的动物: [0]结束添加[1]熊 [2]狮子 [3]老虎 [4]猴子 [5] 长 颈  
鹿

6

其他动物还未引进, 请重新输入

\*\*\*\*\*

请选择添加的动物: [0]结束添加[1]熊 [2]狮子 [3]老虎 [4]猴子 [5] 长 颈  
鹿

4

这是一只猴子

猴子的数量:2

\*\*\*\*\*

请选择添加的动物: [0]结束添加[1]熊 [2]狮子 [3]老虎 [4]猴子 [5] 长 颈  
鹿

2

这是一只狮子

狮子的数量:2

\*\*\*\*\*

请选择添加的动物: [0]结束添加[1]熊 [2]狮子 [3]老虎 [4]猴子 [5] 长 颈  
鹿

2

这是一只狮子

狮子的数量:3

\*\*\*\*\*

请选择添加的动物： [0]结束添加[1]熊 [2]狮子 [3]老虎 [4]猴子 [5]长颈鹿

0

\*\*\*\*\*

请选择你的操作： [0]退出 [1]添加动物[2]查看全部动物

2

动物园的动物有：

熊,狮子,老虎,猴子,长颈鹿,

动物园的动物及其数量有：

熊的数量:1

狮子的数量:3

老虎的数量:1

猴子的数量:2

长颈鹿的数量:1

\*\*\*\*\*

请选择你的操作： [0]退出 [1]添加动物[2]查看全部动物

0

欢迎下次光临动物园

进程已结束，退出代码 0

```
"C:\Program Files\java\jdk-13.0.2\bin\java.exe" "-javaagent:D:\Program Files
*****欢迎来到Lowly-li的动物园!*****
请选择你的操作:    [0]退出  [1]添加动物  [2]查看全部动物
1
*****
请选择添加的动物:  [0]结束添加  [1]熊  [2]狮子  [3]老虎  [4]猴子  [5]长颈鹿
1
这是一只熊
熊的数量:1
*****
请选择添加的动物:  [0]结束添加  [1]熊  [2]狮子  [3]老虎  [4]猴子  [5]长颈鹿
2
这是一只狮子
狮子的数量:1
*****
请选择添加的动物:  [0]结束添加  [1]熊  [2]狮子  [3]老虎  [4]猴子  [5]长颈鹿
3
这是一只老虎
老虎的数量:1
*****
请选择添加的动物:  [0]结束添加  [1]熊  [2]狮子  [3]老虎  [4]猴子  [5]长颈鹿
4
这是一只猴子
猴子的数量:1
*****
请选择添加的动物:  [0]结束添加  [1]熊  [2]狮子  [3]老虎  [4]猴子  [5]长颈鹿
5
这是一只长颈鹿
长颈鹿的数量:1
*****
请选择添加的动物:  [0]结束添加  [1]熊  [2]狮子  [3]老虎  [4]猴子  [5]长颈鹿
6
其他动物还未引进, 请重新输入
*****
请选择添加的动物:  [0]结束添加  [1]熊  [2]狮子  [3]老虎  [4]猴子  [5]长颈鹿
```

```
*****
请选择添加的动物： [0]结束添加 [1]熊 [2]狮子 [3]老虎 [4]猴子 [5]长颈鹿
4
这是一只猴子
猴子的数里:2
*****
请选择添加的动物： [0]结束添加 [1]熊 [2]狮子 [3]老虎 [4]猴子 [5]长颈鹿
2
这是一只狮子
狮子的数里:2
*****
请选择添加的动物： [0]结束添加 [1]熊 [2]狮子 [3]老虎 [4]猴子 [5]长颈鹿
2
这是一只狮子
狮子的数里:3
*****
请选择添加的动物： [0]结束添加 [1]熊 [2]狮子 [3]老虎 [4]猴子 [5]长颈鹿
0

*****
请选择你的操作： [0]退出 [1]添加动物 [2]查看全部动物
2
动物园的动物有：
熊,狮子,老虎,猴子,长颈鹿,
动物园的动物及其数里有：
熊的数里:1
狮子的数里:3
老虎的数里:1
猴子的数里:2
长颈鹿的数里:1

*****
请选择你的操作： [0]退出 [1]添加动物 [2]查看全部动物
0
欢迎下次光临动物园
```