# 内部类定义

# 内部类优点

# 内部类分类

## 实例成员内部类

## 静态成员内部类

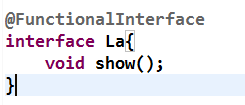
## 局部内部类

## 匿名内部类

# lambda(java8)

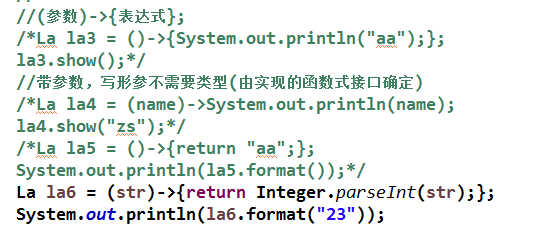
## 函数式接口

接口只能有一个抽象方法。可以有其他方法。



## 语法

(参数)->{表达式};



# 方法引用

lambda只是用一个方法。

|  |
| --- |
| //1.只调用一个方法，必须是类方法  String[] arr = {"aa","bb","cc"};  /\*La1 la = (arrs)->{return Arrays.toString(arrs);};  System.out.println(la.show(arr));\*/  /\*La1 la = Arrays::toString;  System.out.println(la.show(arr));\*/    //2.调用特定对象的实例方法  /\*La1 la = (str)->{return "aa".equals(str);};  System.out.println(la.show("aa"));\*/  /\*La1 la = "aa"::equals;  System.out.println(la.show("aa"));\*/  //System.out::println;    //3.对象的方法使用(str.length())  /\*La1 la = (str)->{return str.length();};  System.out.println(la.show("aaa"));\*/  /\*La1 la = String::length;  System.out.println(la.show("aaa"));\*/    //4.调用构造器  /\*La1 la = ()->{return new Person1();};  la.create().test();\*/  /\*La1 la = Person1::new;  la.create().test();\*/ |

# 匿名内部类和lambda区别

## lambda只能调用函数接口

匿名内部类可以是实现普通接口，抽象类，普通类

## lambda不能调用默认方法。

# 多态

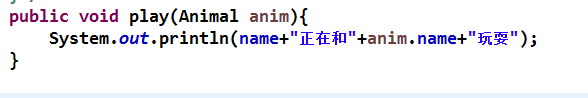
多态使用前提:继承(实现)基础之上

## 使用

父类引用 => 子类对象

## 使用场景

父类类型作为参数类型



父类类型作为返回值类型

