JDK1.8有的是为了解决匿名内部类过于繁琐而来的

Lambda的3种形式的语法:

1)(参数)->单行语句;

2)(参数)->{多行语句};

3)(参数)->表达式;

范例:匿名内部类的使用

|  |
| --- |
| public class test{  public static void main(String args[]) throws Exception {  fun(new IMessage() {  @Override  public void f(String str) {  System.out.print("你好" + str);  }  });  }  public static void fun(IMessage msg){  msg.f("hello,world");  }  }  interface IMessage{  public void f(String str);  } |

范例2:再使用lambda lambda的形式1 (参数)->单行语句;

|  |
| --- |
| public class test{  public static void main(String args[]) throws Exception {  fun((str)->System.out.print( "你好" + str));  }  public static void fun(IMessage msg){  msg.f("hello,world");  }  }  interface IMessage{  public void f(String str);  } |

Lambda形式2 (参数)->{多行语句};

|  |
| --- |
| public class test{  public static void main(String args[]) throws Exception {  int tmp = fun((i)->{  int x = 2;  return x \* i;  });  System.out.print(tmp);  }  public static int fun(IMessage msg){  return msg.f(100);  }  }  interface IMessage{  public int f(int i);  } |

Lambda形式3 (参数)->表达式;

|  |
| --- |
| public class test{  public static void main(String args[]) throws Exception {  int tmp = fun((i)->i);  System.out.print(tmp);  }  public static int fun(IMessage msg){  return msg.f(100);  }  }  interface IMessage{  public int f(int i);  } |