实验4 内部类与匿名类、异常类

**一 、实验目的与要求**

1、掌握内部类的常用方法。

2、掌握匿名类的常用方法。

3、掌握接异常类的常用方法。

**二、 实验内容**

根据源代码，按照注释要求，完成代码填空，使程序能够运行得出结果。

1、红牛农场

**RedCowForm.java**

public class RedCowForm {

String formName;

//【代码1】//用内部类RedCow声明对象cow

RedCow cow;

RedCowForm(){

}

RedCowForm(String s){

//【代码2】//创建cow

cow =new RedCow(100,200,300);

formName=s;

}

public void showCowMess(){

cow.speak();

}

class RedCow{

String cowName="红牛";

int height,weight,price;

RedCow(int h,int w,int p){

height=h;

weight=w;

price=p;

}

void speak(){

System.out.println(cowName+"，身高："+height+"cm，体重："+weight+"kg");

System.out.println("生活在"+formName);

}

}

}

**MainClass.java**

public class MainClass {

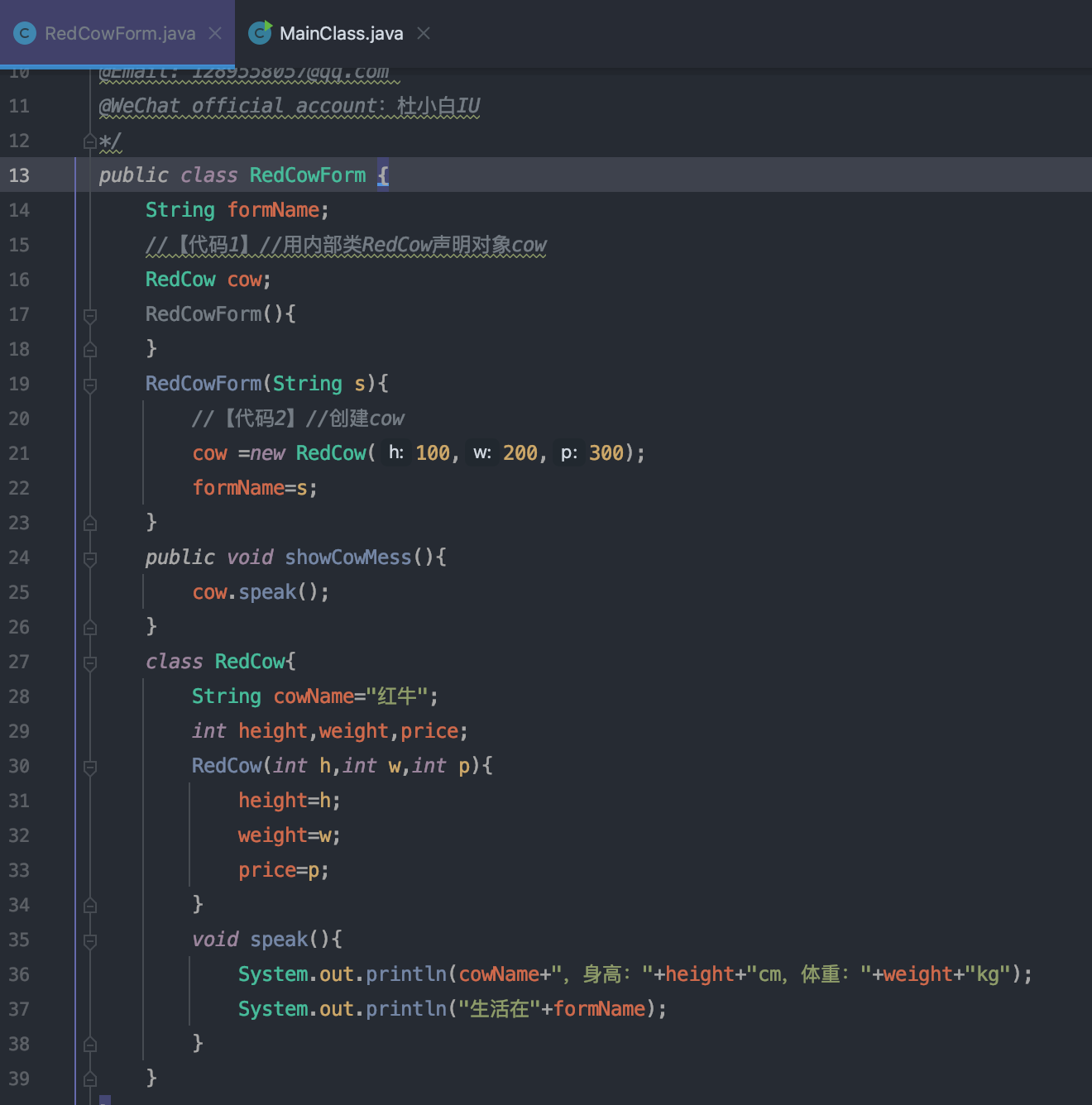
public static void main(String[] args) {

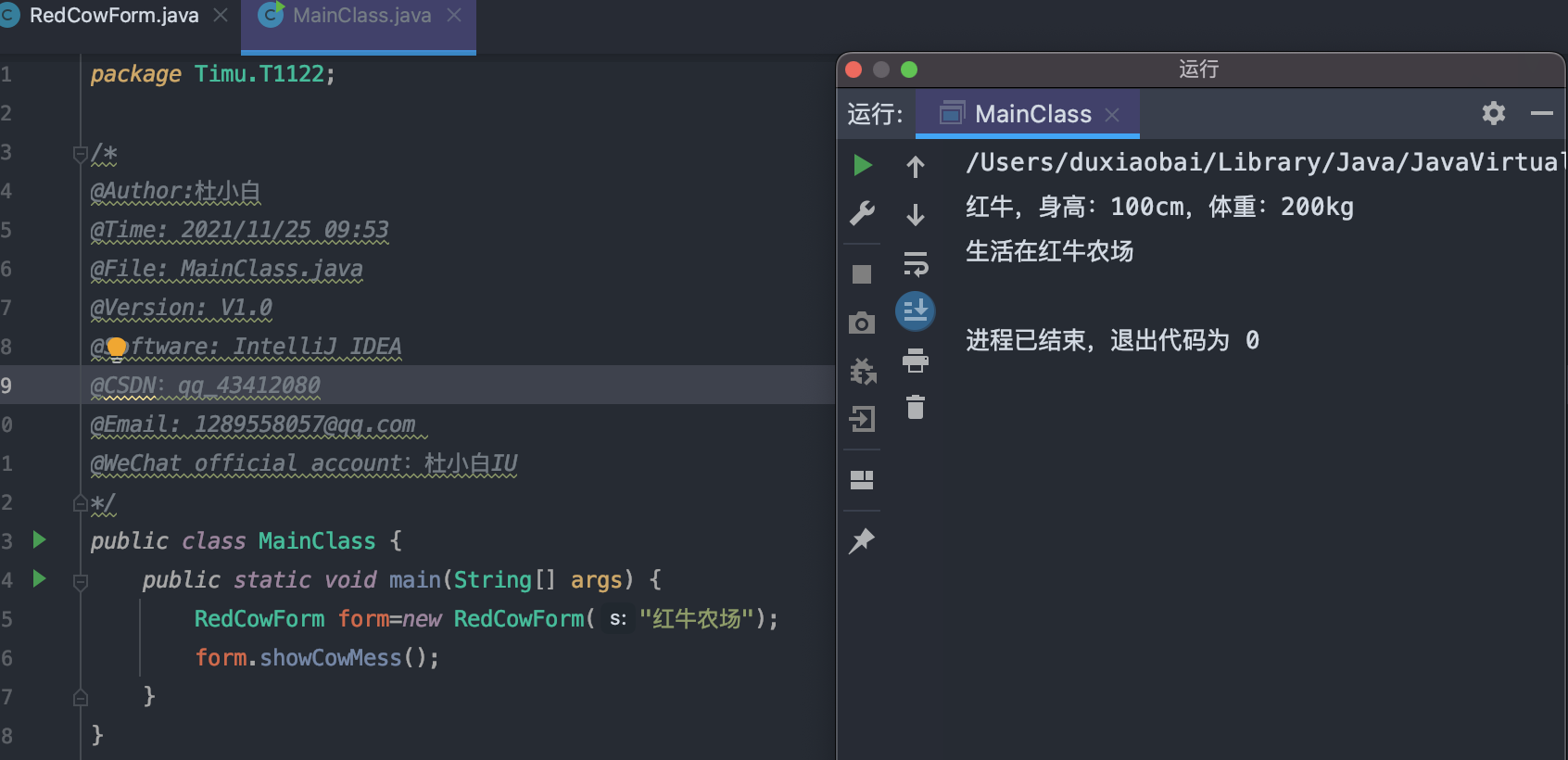
RedCowForm form=new RedCowForm("红牛农场");

form.showCowMess();

}

}





2、内部类的使用

**Outer.java**

public class Outer{

private static int size;

/\*\* 内部类Inner的声明 \*/

public class Inner{

private int size;

public void doStuff(int size){

【代码1】 //存取局部变量size加3

size=size+3;

【代码2】 //存取其内部类的成员变量size加2

this.size=size+2;

【代码3】 //存取其外部类的成员变量size加1

Outer.this.size=size+1;

System.out.println(size+" "+this.size+" "+Outer.this.size);

}

}//内部类Inner结束

/\*\* 类Outer中定义的实例方法testInner()方法 \*/

public void testInner(){

Inner i=new Inner();

【代码4】//调用Inner类中的doStuff方法

i.doStuff(size);

}

/\*\* main()方法 \*/

public static void main(String args[]){

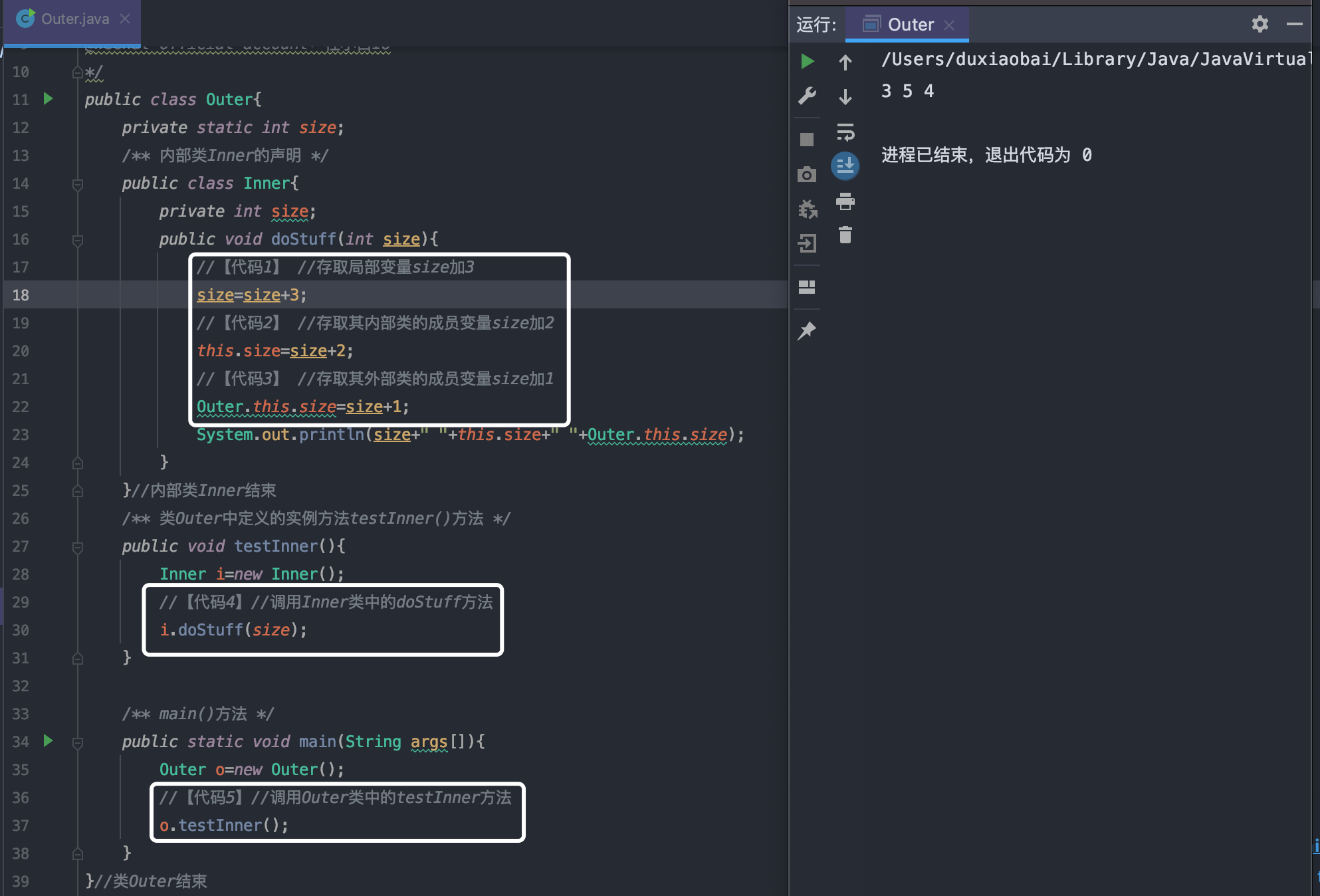
Outer o=new Outer();

【代码5】//调用Outer类中的testInner方法

o.testInner();

}

}//类Outer结束



3、匿名类的使用方法

interface A{

public void printInfo() ; //

}

class X {

public void fun2(A a){

a.printInfo() ;

}

public void fun1(){

this.fun2(　【代码1】//书写匿名内部类

new A(){

@Override

public void printInfo() { // 实现接口中的抽象方法

System.out.println("实现了接口中的抽象方法");

}

}

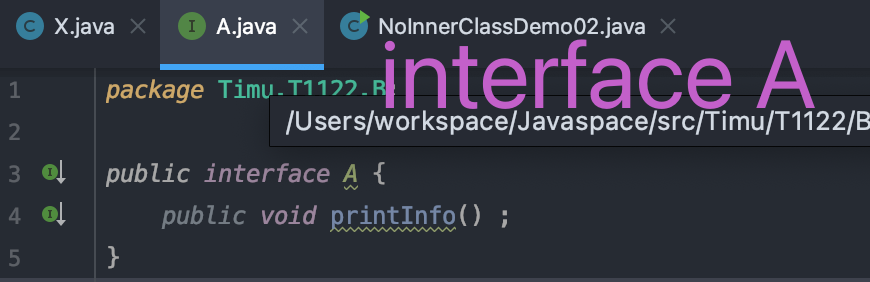
) ; } }

public class NoInnerClassDemo02{

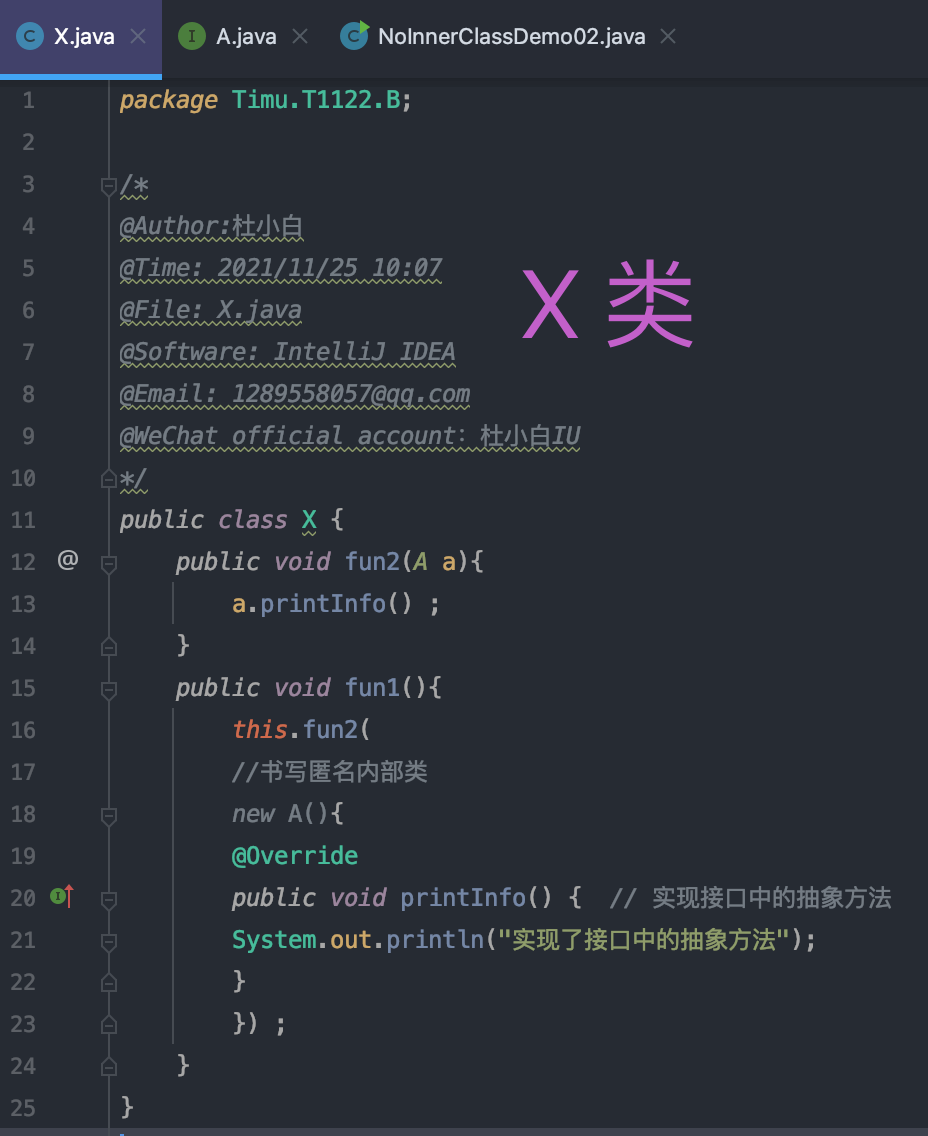
public static void main(String args[]){

// 【代码2】实例化X类的对象，并调用fun1()方法

new X().fun1();

 }

};



4、异常类的使用

背景：车站检查危险品的设备，如果发现危险品会发出警告。编程模拟设备发现危险品。根据如下描述编写各类的代码。

（1）编写一个Exception的子类DangerException。存为DangerException.java文件



（2）编写一个行李类Goods.java



（3）编写一个Machine类存为Machine.java文件，该类的方法checkBag（Goods g）---当发现参数g是危险品时（其isDanger属性为true）将抛出DangerException异常对象。否则就输出“检查通过”

（4）主类Check.java，完成功能如下：

a.实例化一个Machine对象

b.初始化一个String数组变量name，String name[]={“苹果”, “炸药”, “西服”, “硫酸”, “手表”, “硫磺”}

c.初始化一个Goods型数组变量goods，Goods[] goods=new Goods[name.length];

d.循环设置数组中各个元素的isDanger属性和name属性。

for(….){

goods[i]=new Goods();

goods[i].setName(name[i]);

if(i%2==0){

goods[i].setIsDanger(false);}

else{

goods[i].setIsDanger(true);}

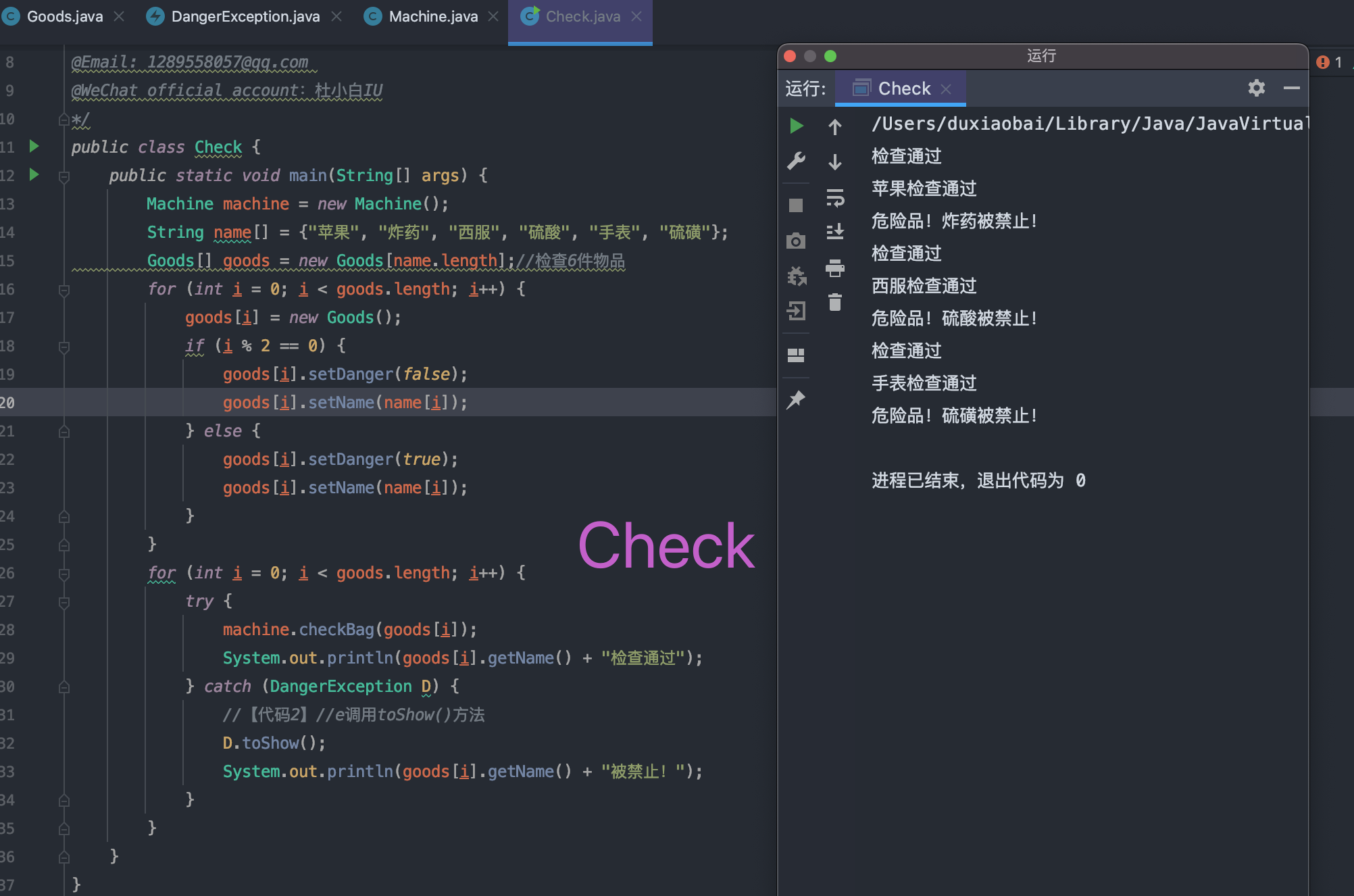
e.循环输出行李的名称

f.循环调用Machine对象的checkBag（Goods g）方法。



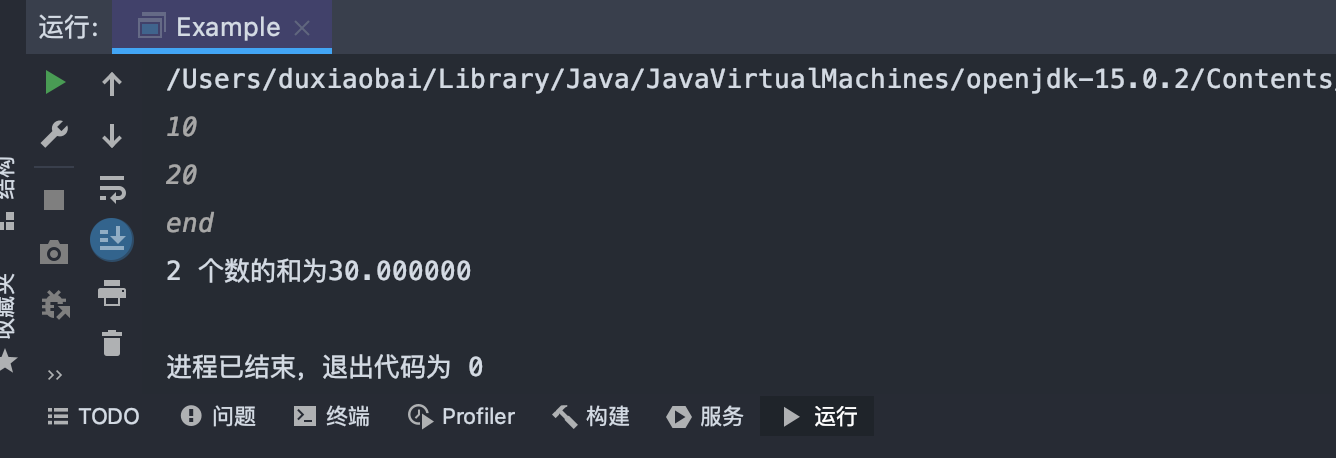


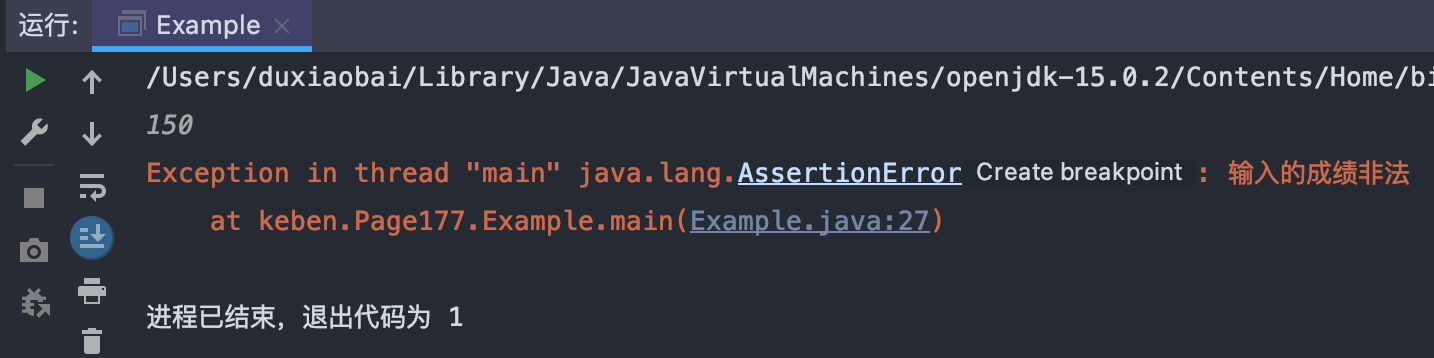


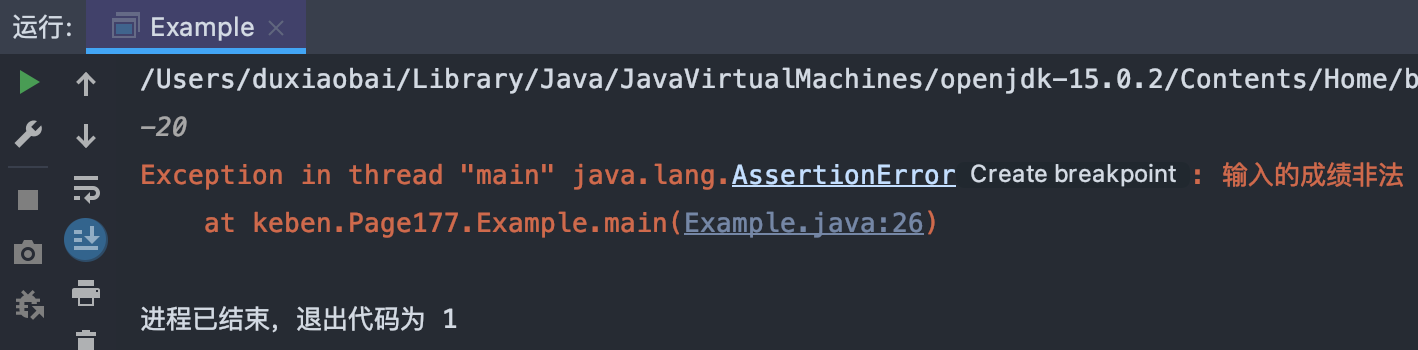


5.教材P177页第5题编程题。

运行结果：







**三、实验内容中遇到的问题、解决方法和体会**

1.Exception在程序中必须使用try...catch进行处理。

2.断言用来检测程序的执行结果。