实验十二： 多线程编程

## 实验目的

1、 了解线程的调度和执行过程，掌握Java 语言中多线程编程的基本方法

2、 掌握Runnable 接口实现多线程的方法

3、 掌握Thread 类实现多线程的用法

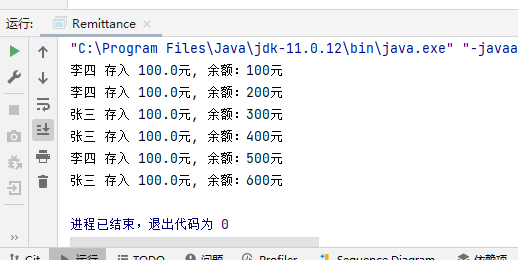
## 实验内容

采用多线程机制模拟汇款业务。定义一银行类可接受顾客的汇款，其属性count用于存储账户金额。现有两名顾客，每人分三次、每次100元将钱汇入count。每进行一次汇款，便输出汇款人和账户总额。

## 实验代码：

package com.shf.demo12;  
  
public class Remittance *{* public static void main*(*String*[]* args*) {* Customer c1 = new Customer*(*"张三"*)*;  
 Customer c2 = new Customer*(*"李四"*)*;  
  
 c1.start*()*;  
 c2.start*()*;  
 *}  
}*class Customer extends Thread*{* private String name;  
 public Customer*(*String name*){* this.name = name;  
 *}* @Override  
 public void run*() {  
// 存入3次 每次100* for *(*int i = 0; i < 3; i++*) {* Bank.*save(*name,100*)*;  
 *}  
 }  
}*class Bank*{  
// 余额* private static int *balance* = 0;  
 *//用synchronized关键字设定该方法为线程同步的* public synchronized static void save*(*String name,double money*) {* try *{  
 //利用sleep（）方法令当前正在执行的线程进入休眠状态，且参数类型必须为int* Thread.*sleep((*int*) (*Math.*random()* \* 2000*))*;  
 *}* catch *(*InterruptedException e*) {* e.printStackTrace*()*;  
 *}  
 balance*+=money;  
 System.*out*.print*(*name+" 存入 "+money+"元, "*)*;  
 System.*out*.println*(*"余额："+*balance*+"元"*)*;  
 *}  
}*

## 实验截图



## 实验小结

通过本次实验我学会java中的多线程操作，一个进程中可以并发多个线程，每条线程并行执行不同的任务。进程的声明周期①新建状态②就绪状态③运行状态④阻塞状态⑤死亡状态。我们可以通过实现 Runnable 接口；通过继承 Thread 类本身；通过 Callable 和 Future 创建线程，三种方式创建线程。. 采用实现 Runnable、Callable 接口的方式创建多线程时，线程类只是实现了 Runnable 接口或 Callable 接口，还可以继承其他类。使用继承 Thread 类的方式创建多线程时，编写简单，如果需要访问当前线程，则无需使用 Thread.currentThread() 方法，直接使用 this 即可获得当前线程。