```
package com.sxt.syn;
/**
 *  线程安全：在并发时保证数据的正确性，效率尽可能高
 * synchronized
 * 1.同步方法
 * 2.同步块
 * @author DELL
 *
 */

public class SynTest5 {
   public static void main(String[] args) {
             Account account=new Account(100,"结婚基金");
             SafeDrawing you=new SafeDrawing(account,80,"可悲的你");
             SafeDrawing wife=new SafeDrawing(account,90,"happy的她");
             you.start();
                 wife.start();
}

}

//账户已经在UnsafeTest2存在

//取款
class SafeDrawing extends Thread{
        Account account;//取钱账户
        int drawingMoney;//取的钱数
        int packetTotal;//口袋钱数

        public SafeDrawing(Account account,int drawingMoney,String name) {
                super(name);
                this.account=account;
                this.drawingMoney=drawingMoney;
        }
        @Override
        public void run() {
                test();
    }

        //目标不对锁定失败，这里不是锁this应该锁定account
        public synchronized void test(){
                if(account.money-drawingMoney<0) {
                        return;
                }
                try {
                        Thread.sleep(200);
                } catch (InterruptedException e) {
                        e.printStackTrace();
                }
                account.money-=drawingMoney;
                packetTotal+=drawingMoney;
            System.err.println(this.getName()+"-->账户余额为"+account.money);
            System.err.println(this.getName()+"-->口袋的钱为"+packetTotal);

        }
}
```