```
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,目标更明确
 * @author DELL
 *
 */

public class SynBlockTest6 {
   public static void main(String[] args) {
            Account account=new Account(100,"结婚基金");
            SynDrawing you=new SynDrawing(account,80,"可悲的你");
            SynDrawing wife=new SynDrawing(account,90,"happy的她");
            you.start();
                wife.start();
}

}

//账户已经在UnsafeTest2存在

//模拟存款 线程安全
class SynDrawing extends Thread{
        Account account;//取钱账户
        int drawingMoney;//取的钱数
        int packetTotal;//口袋钱数

        public SynDrawing(Account account,int drawingMoney,String name) {
                super(name);
                this.account=account;
                this.drawingMoney=drawingMoney;
        }
        @Override
```

```
        public void run() {
                test();
    }

        //目标锁定account
        public  void test(){
                //提高性能
                if(account.money<=0) {
                        return;
                }
                //同步块
                synchronized(account) {
                        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);
                }



        }
}
```