**生产者：  往一个公共的盒子里面放苹果**

**消费者：从公共的盒子里面取苹果**

**盒子：盒子的容量不能超过5**

实现方法：   wait()

看盒子代码

        public class PublicBox {

           private int apple = 0;

           public synchronized void increace() {

               while (apple ==5) {

                   try {

                       wait();

                   } catch (InterruptedException e) {

                       e.printStackTrace();

                   }

               }

               apple++;

               System. out .println("生成苹果成功！" );

               notify();

           }

           public synchronized void decreace() {

               while (apple ==0) {

                   try {

                       wait();

                   } catch (InterruptedException e) {

                       e.printStackTrace();

                   }

               }

               apple--;

              System. out.println( "消费苹果成功！" );

               notify();

           }

           public static void main(String []args)

              {

                     PublicBox box= new PublicBox();

                     Consumer con= new Consumer(box);

                     Producer pro= new Producer(box);

                     Thread t1= new Thread(con);

                     Thread t2= new Thread(pro);

                     t1.start();

                     t2.start();

              }

       }

生产者代码（定义十次）：

public class Producer implements Runnable {

    private PublicBox box;

    public Producer(PublicBox box) {

        this .box = box;

    }

    @Override

    public void run() {

        for( int i=0;i<10;i++)

       {

               try {

                     System. out .println("pro  i:" +i);

                           Thread. sleep(30);

                     } catch (InterruptedException e) {

                            // TODO: handle exception

                           e.printStackTrace();

                     }

            box.increace();

       }

    }

}

消费者代码（同样十次）：

public class Consumer implements Runnable {

    private PublicBox box;

    public Consumer(PublicBox box) {

        this .box = box;

    }

    @Override

    public void run() {

        for( int i=0;i<10;i++)

       {

             try {

                     System. out .println("Con: i " +i);

                           Thread. sleep(3000);                // 这里设置跟上面30不同是为了 盒子中的苹果能够增加，不会生产一个马上被消费

                     } catch (InterruptedException e) {

                            // TODO: handle exception

                           e.printStackTrace();

                     }

            box.decreace();

        }

 }

}