class Share

{

private int s;

private boolean empty=true;

public synchronized int get()

{

while(empty==true)

{

try

{

wait();

}

catch(InterruptedException e)

{}

}

System.out.println("value consumed"+s);

empty=true;

notifyAll();

return s;

}

public synchronized void put(int s)

{

while(empty==false)

{

try

{

wait();

}

catch(InterruptedException e)

{}

}

System.out.println("value produced"+s);

this.s=s;

empty=false;

notifyAll();

}

}

class producer extends Thread

{

private Share shared;

public producer(Share s)

{

shared=s;

}

public void run()

{

for(int i=0;i<10;i++)

{

shared.put(i);

}

}

}

class consumer extends Thread

{

private Share shared;

private int value;

public consumer(Share s)

{

shared=s;

}

public void run()

{

for(int i=0;i<10;i++)

{

value=shared.get();

}

}

}

class producerconsumer

{

public static void main(String args[])

{

Share s=new Share();

consumer c=new consumer(s);

producer p=new producer(s);

p.start();

c.start();

}

}