Producerconsumer:

class Q

{

int n;

boolean valueSet = false;

synchronized int get()

{

while(!valueSet)

try

{

wait();

}

catch(InterruptedException e)

{

System.out.println("InterruptedException caught");

}

System.out.println("Got: " + n);

valueSet = false;

notify();

return n;

}

synchronized void put(int n)

{

while(valueSet)

try

{

wait();

}

catch(InterruptedException e)

{

System.out.println("InterruptedException caught");

}

this.n = n;

valueSet = true;

System.out.println("Put: " + n);

notify();

}

}

class Producer implements Runnable

{

Q q;

Producer(Q q)

{

this.q = q;

new Thread(this, "Producer").start();

}

public void run()

{

int i = 0;

while(true)

{

q.put(i++);

}

}

}

class Consumer implements Runnable

{

Q q;

Consumer(Q q)

{

this.q = q;

new Thread(this, "Consumer").start();

}

public void run()

{

while(true)

{

q.get();

}

}

}

class pc

{

public static void main(String args[])

{

Q q = new Q();

new Producer(q);

new Consumer(q);

System.out.println("Press Control-C to stop.");

}

}