队列的链式存储结构不用带头结点的单链表来实现，为了方便实现入队和出队操作，需要引进两个指针，一个指向队头的front和一个指向队尾的rear

内部结构：一个指向队头的指针+一个指向队尾的指针

功能及实现：

1. public void clear() { //清空队列，直接将指针赋值为null即可

rear=front=null;

}

2. public boolean isEmpty() { //判断队列是否为空

return front==null; //用队头队尾判断都可以

}

3. public int length() { //队列长度

Node p = new Node();

p=front; //从队头开始遍历

int counter=0; //节点计数器

while(p!=null){

counter++;

p=p.next;

}

return counter;

}

4. public Object peek() { //取队首元素，因为是用没有头节点的队列来存储的

return front.val; //直接返队头的数据就行

}

5. public void offer(Object x) { //入队，每次都在队尾入队

Node p = new Node(x);

if(front!=null) { //判断队列是否为空

rear.next=p; //不空加入队列

rear=p;//修改队尾指针

}

else //第一次添加把头尾节点连在一起

front=rear=p;//这句代码很重要，把首尾连在一起

}

注意：有的顺序队列最后一个rear指针指向的单元并不放入数据，所以就是先存数据，然后改指针到下一个空的单元，rear指针始终指向空的数据单元。链队则是真的指向队尾元素，不为空。

6. public Object poll() { //出队，每次都在队头出队

if(front!=null){ //队不为空，出队

Node p = front; //保存出队的元素

front=front.next; //修改队头指针

return p.val; //返回刚出队的元素数据

}

else //队空则返回null

return null;

}