

```

/*
 * 数据结构-队列的实现
 */
public class Queue {
    private int [] array;//存放号码
    private int size;//“当前”有效元素的个数
    private int maxSize;//“当前”允许存放最多的元素个数
    private int value;
    //总是存放“当前”最后一个元素的值

    //构造器： constructor
    /*
     * 1.完成对象初始化的代码段
     * 2.名字和类名相同
     * 3.没有返回值
     * 4.在创建对象的时候由系统自动调用，
     * 程序员无权直接调用。
     */
    public Queue()
    {
        array=new int[20];
        maxSize=20;
    }

    //按照要求初始化队列的长度
    public Queue(int size)
    {
        array=new int[size];
        maxSize=size;
    }
    //（1）检测内存是否够用
    //（2）如果不够，
    //动态开辟内存：根据内存数量的需求动态确定内存空间的大小。
    public void testArray()
    {
        if(size>=maxSize)
        {
            int [] newArray=new int[maxSize+20];
            System.arraycopy(array,
                            0,
                            newArray,
                            0,
                            array.length);
            array=newArray;
        }
    }
}

```

```

        maxSize+=20;
    }
}

//（1）取号
public void add()
{
    //保证内存够用
    testArray();
    array[size]=value+1;
    size++;
    value++;
}

//（2）叫号
public void remove()
{
    //（1）队列为空的处理
    if(size==0)
        return;
    //（2）删除
    for(int i=0;i<size-1;i++)
        array[i]=array[i+1];
    array[size-1]=0;
    size--;
}

//（3）返回当前排队人数
public int size()
{
    return size;
}

//（4）输出排队细节
public void print()
{
    for(int i=0;i<size;i++)
    {
        System.out.print(array[i]+" ");
    }
    System.out.println();
    System.out.println("-----");
}

```

```
public static void main(String[] args) {  
    Queue queue=new Queue();  
    //（1）测试 add  
    queue.add();  
    queue.add();  
    queue.print();  
    //（2）测试 remove  
    queue.remove();  
    queue.print();  
    //（3）测试队列被删空  
    queue.remove();  
    queue.print();  
    //（4）测试队列被删空之后再增加  
    queue.add();  
    queue.add();  
    queue.print();  
    //（5）测试动态开辟内存  
    for(int i=0;i<30;i++)  
        queue.add();  
    queue.print();  
}  
}
```