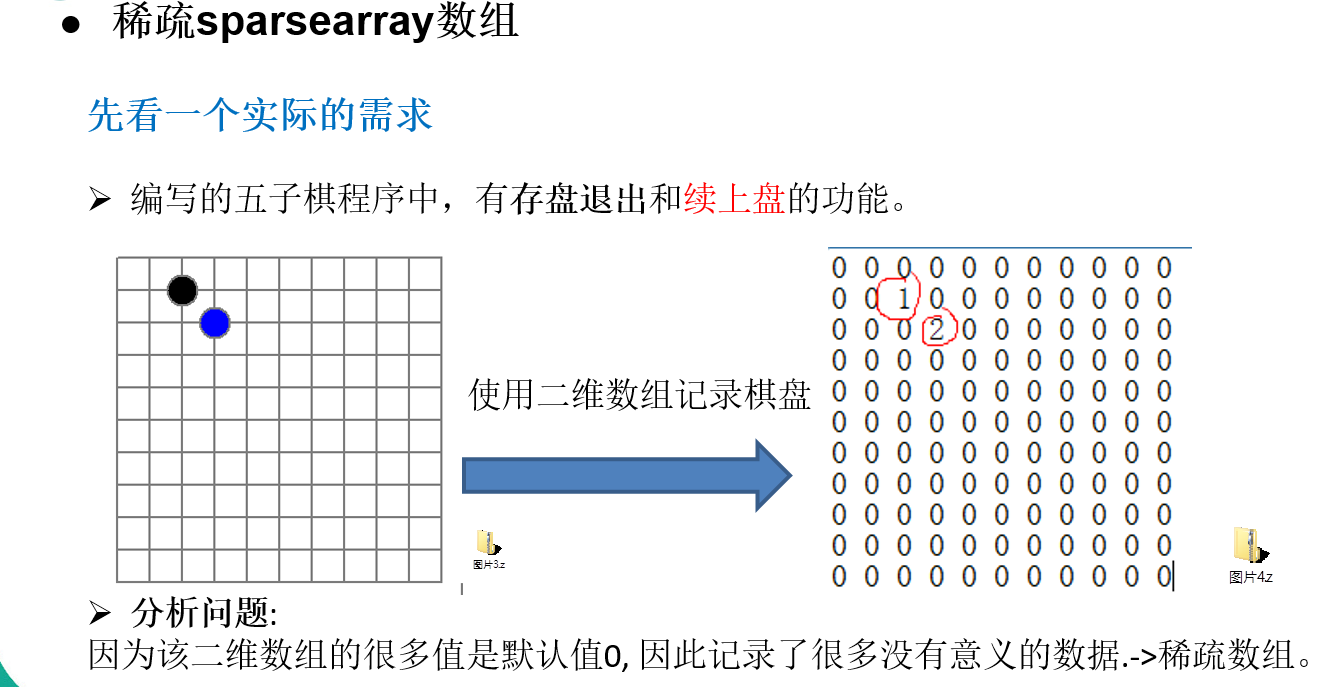
# 一 稀疏数组与二维数组

## 1 应用场景



## 2 基本介绍

当一个数组中大部分元素为０，或者为同一个值的数组时，可以使用稀疏数组来保存该数组。

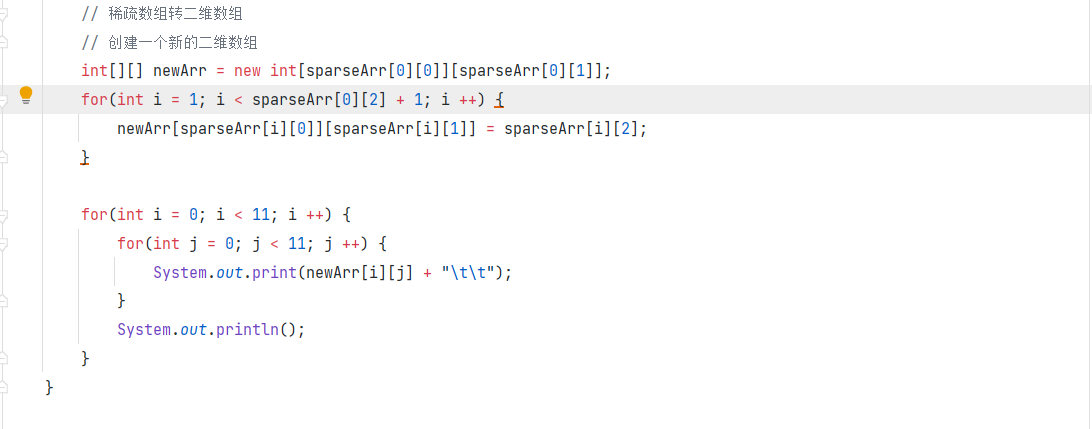
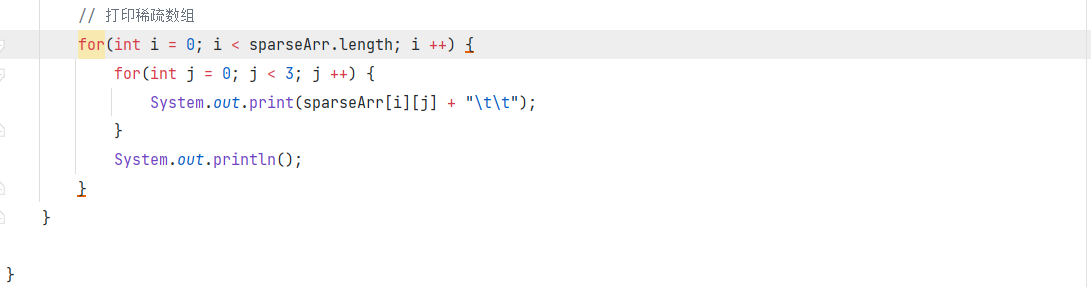
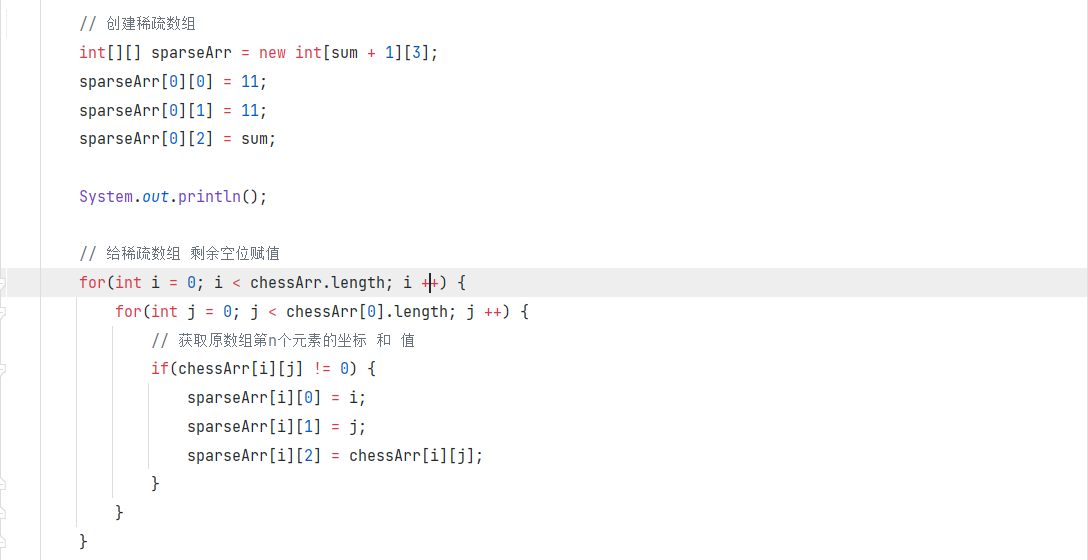
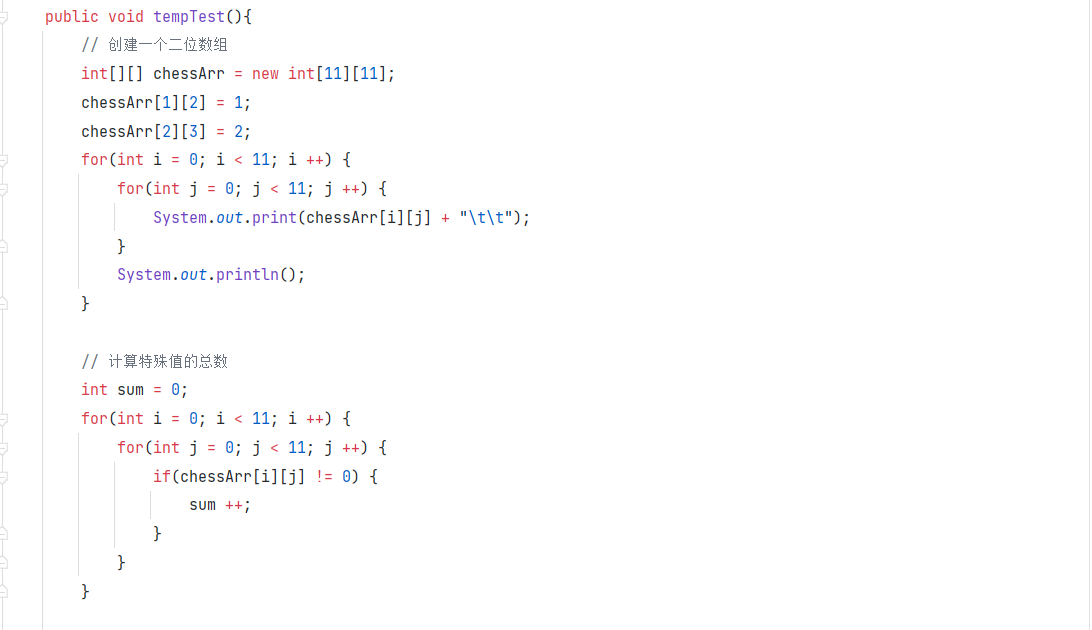
## 3 实现步骤

* 1. 记录数组一共有几行几列，有多少个不同的值。
  2. 把具有不同值的元素的行列及值记录在一个小规模的数组中，从而缩小程序的规模。

## 举例说明



## 5 代码实现



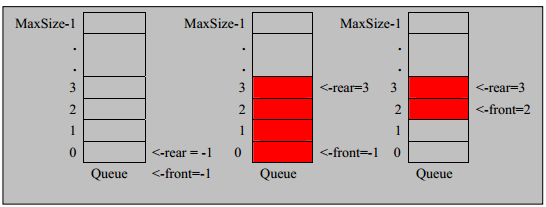
# 二 数组模拟队列

## 1 应用场景

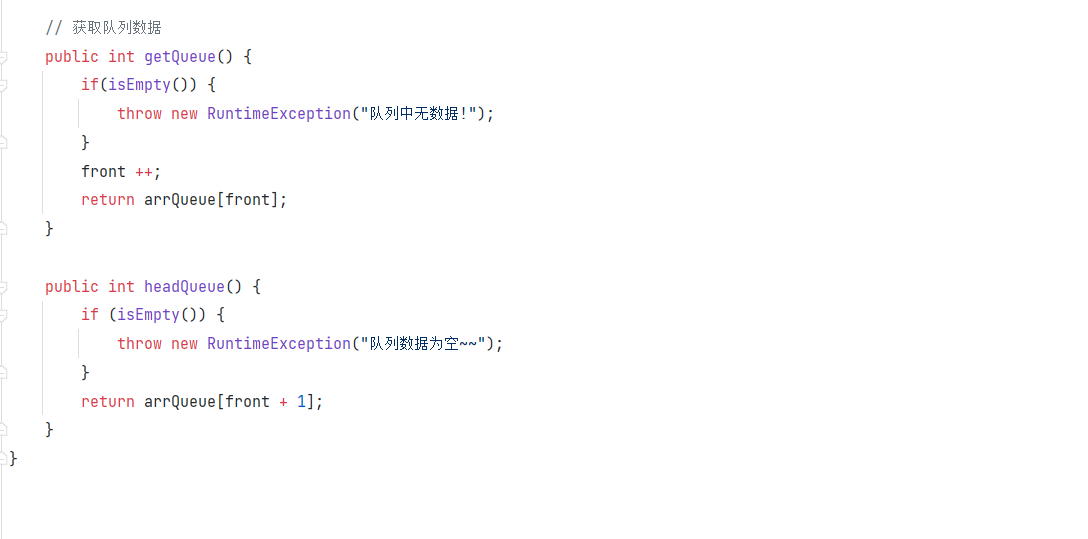


## 2 基本介绍

1. 队列是一个有序列表，可以用**数组**或是**链表**来实现。
2. 遵循**先入先出**的原则。即：先存入队列的数据，要先取出。后存入的要后取出
3. 示意图：(使用数组模拟队列示意图)



## 3 代码实现



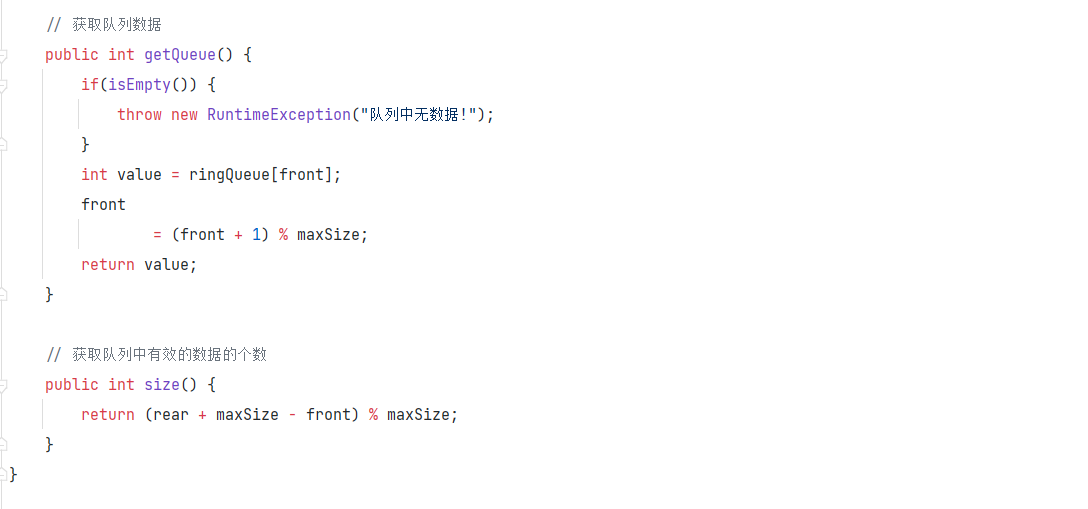
# 三 数组模拟环形队列

## 1 基本介绍

对前面的数组模拟队列的优化，充分利用数组. 因此将数组看做是一个环形的。(通过取模的方式来实现即可)

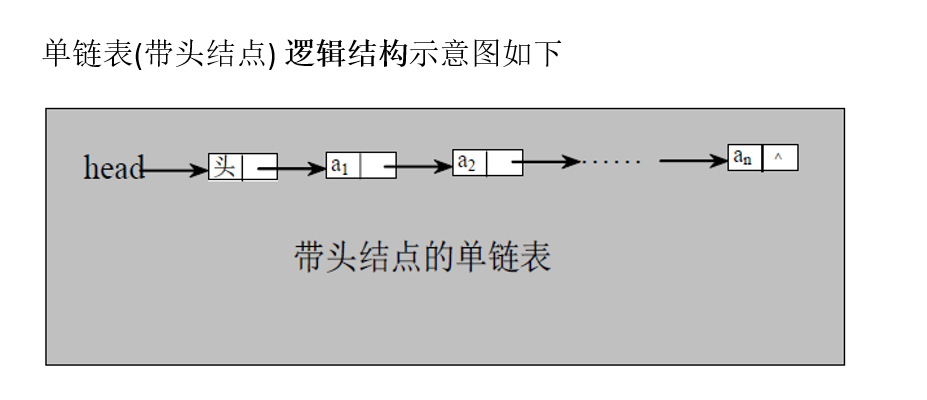
## 3 代码实现



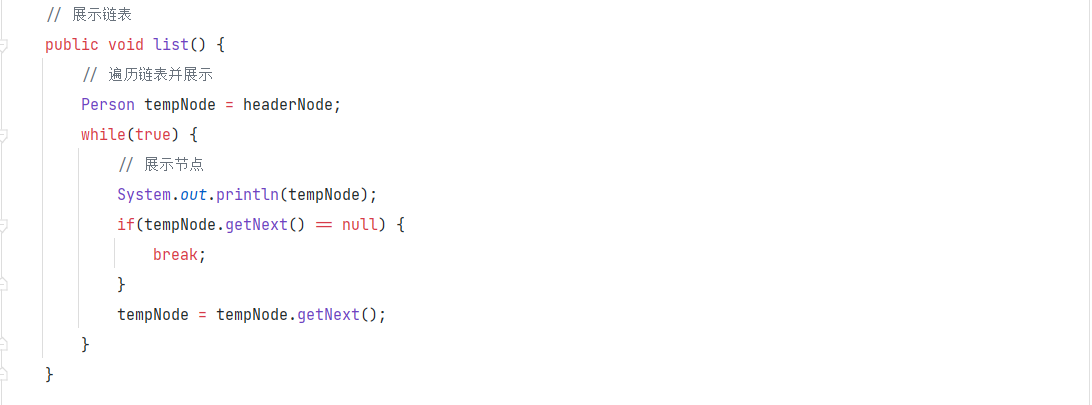
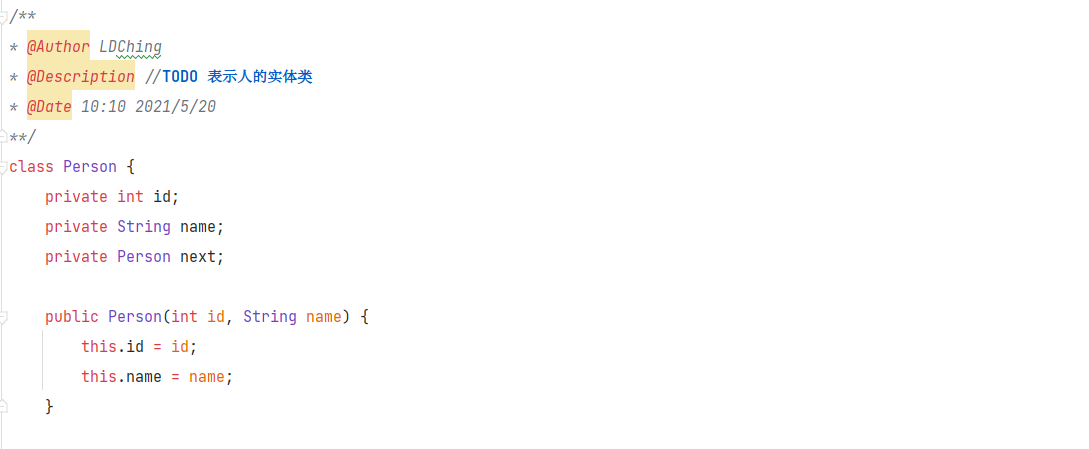


# 四 单链表

## 1 介绍



## 2 代码实现（不考虑排序）

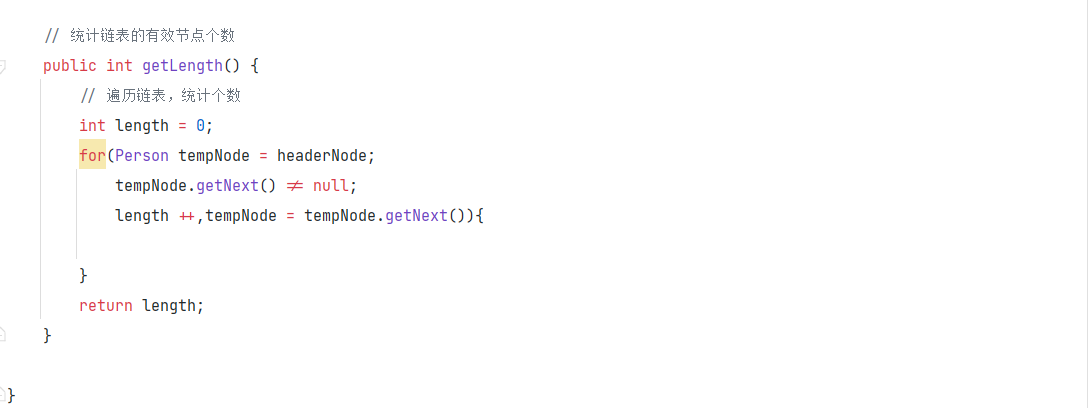
## 3 代码实现（考虑排序）





## 扩展

### 1 求单链表中有效节点的个数



### 2 求单链表中的倒数第k个节点



### 3 单链表的反转



### 4 从尾到头打印单链表



# 五 双向链表

## 1 介绍

和单链表类似，多了prev指针指向前一个节点，可以直接删除节点不需要先找到前一个节点再删除。

## 2 代码实现

默认在尾部增加节点



