**package** com.hongzheng;

**import** java.util.Scanner;

**public** **class** CircleArray {

**public** **static** **void** main(String[] args) {

Circle circle = **new** Circle(4);

**boolean** loop = **true**;

**char** key = ' ';

Scanner sc = **new** Scanner(System.***in***);

**while** (loop) {

System.***out***.println("s(show):显示队列");

System.***out***.println("a(add):添加数据");

System.***out***.println("g(get):取队首");

System.***out***.println("h(head):显示队首");

System.***out***.println("e(exit):退出程序");

key = sc.next().charAt(0);

**switch** (key) {

**case** 's':

circle.showQueue();

**break**;

**case** 'a':

System.***out***.println("输入需要添加的数据");

**int** n = sc.nextInt();

circle.addQueue(n);

**break**;

**case** 'g':

System.***out***.println(circle.getQueue());

**break**;

**case** 'h':

System.***out***.println(circle.headQueue());

;

**break**;

**case** 'e':

loop = **false**;

sc.close();

**break**;

**default**:

**break**;

}

}

}

}

**class** Circle {

**private** **int** maxSize;

**private** **int** front; // 默认初始值为0，指向队首

**private** **int** rear; // 默认初始值为0，指向队尾的后一个

**private** **int**[] arr;

// 构造器

**public** Circle(**int** maxSize) {

**this**.maxSize = maxSize;

arr = **new** **int**[maxSize];

}

// 判断为满

**public** **boolean** isFull() {

**return** (rear + 1) % maxSize == front;

}

// 判断是否为空

**public** **boolean** isEmpty() {

**return** rear == front;

}

// 显示队列

**public** **void** showQueue() {

**if** (isEmpty()) {

**throw** **new** RuntimeException("队列为空~~~");

}

**for** (**int** i = front; i < front + size(); i++) {

System.***out***.printf("arr[%d]=%d\n", i % maxSize, arr[i % maxSize]);

}

}

// 取队首

**public** **int** getQueue() {

**if** (isEmpty()) {

**throw** **new** RuntimeException("队列为空~~~");

}

**int** value = arr[front];

front = (front + 1) % maxSize;

**return** value;

}

// 显示队首

**public** **int** headQueue() {

**if** (isEmpty()) {

**throw** **new** RuntimeException("队列为空~~~");

}

**return** arr[front];

}

// 加数据

**public** **void** addQueue(**int** n) {

**if** (isFull()) {

System.***out***.println("队列已满~~~");

**return**;

}

arr[rear] = n;

rear = (rear + 1) % maxSize;

}

// 判断队列中有多少个数据

**public** **int** size() {

**return** (rear - front + maxSize) % maxSize;

}

}

解析：

1. front指向队首,初始值为0
2. rear指向队尾的后一个元素，初始值为0
3. 判断队满的条件为(rear+1)%maxSize == front
4. 判断队列为空的条件为 rear == front
5. 队列长度为 (rear-front+maxSize)%maxSize

图解：

