数组模拟线性表

注意 ：在用数组模拟线性队列的时候，需要注意以下事项，在抛异常之后的语句不再执行

①front 为队首的前一个指针；

②rear 为队列中最后一个元素的指针；

③当front==rear , rear==-1 时队列为空 ；

④当rear == maxSize-1 时队满；

**package** DataStructure;

**import** java.util.Scanner;

**public** **class** ArrayQueueDemo {

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

// 测试一把

// 创建一个队列

ArrayQueue queue = **new** ArrayQueue(3);

**char** key = ' '; // 接收用户输入

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

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

// 显示菜单

**while** (loop) {

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

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

System.***out***.println("a(add)：添加队列");

System.***out***.println("g(get)：取出队列");

System.***out***.println("h(head)：查看队列头");

key = sc.next().charAt(0); // 接收一个字符

**switch** (key) {

**case** 's':

queue.showQueue();

**break**;

**case** 'a':

System.***out***.println("请输入一个数据");

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

queue.addQueue(value);

**break**;

**case** 'g':

**int** res = queue.getQueue();

System.***out***.printf("取出的数据是%d\n", res);

**break**;

**case** 'h':

**try** {

**int** rs = queue.headQueue();

System.***out***.printf("取出的对列头为%d\n", rs);

} **catch** (Exception e) {

// **TODO**: handle exception

}

**break**;

**case** 'e':

sc.close();

loop = **false**;

System.***out***.println("程序退出~~~");

**break**;

**default**:

**break**;

}

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

}

}

}

// 使用数组模拟队列编写一个ArrayQueue类

**class** ArrayQueue {

**private** **int** maxSize; // 表示数组中的最大容量

**private** **int** front; // 对列头

**private** **int** rear; // 队列尾

**private** **int**[] arr; // 该数组用于存放数据，模拟队列

// 创建队列的构造器

**public** ArrayQueue(**int** arrMaxSize) {

maxSize = arrMaxSize;

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

front = -1; // front指向数列的前一个位置

rear = -1; // rear指向队列尾部，那个具体的值

}

// 判断队列是否为满

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

**return** rear == maxSize - 1;

}

// 判断队列是否为空

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

**return** rear == front;

}

// 添加一个数据

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

// 先判断队列是否满

**if** (isFull()) {

System.***out***.println("队列满，不可以加入~~~");

**return**;

}

rear++;

arr[rear] = n;

}

// 获取队列数据，出队列

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

// 判断队列是否为空

**if** (isEmpty()) {

System.***out***.println("队列为空~~~");

**throw** **new** RuntimeException("队列为空，不可取数据");

// throw 后面的语句不可到达

}

front++;

**return** arr[front];

}

// 显示所有数据

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

**if** (isEmpty()) {

System.***out***.println("队列为空~~~");

}

**for** (**int** i = 0; i < arr.length; i++) {

System.***out***.print(arr[i] + " ");

}

}

// 显示队列头，不是取出数据，仅仅是显示

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

// 判断

**if** (isEmpty()) {

System.***out***.println("队列空");

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

}

**return** arr[front + 1];

}

}