### JAVA第五阶段—DAY06-JAVA作业答案

1. 自己实现队列结构，试写出相应的判断队列空、判断队列满、出队算法和入队算法。

public class MyQueue {

private Object[] queArray;

// 队列总大小

private int maxSize;

// 前端

private int front;

// 后端

private int rear;

// 队列中元素的实际数目

private int nItems;

public MyQueue(int s) {

maxSize = s;

queArray = new Object[maxSize];

front = 0;

rear = -1;

nItems = 0;

}

// 队列中新增数据

public void insert(int value) {

if (isFull()) {

System.out.println("队列已满！！！");

} else {

// 如果队列尾部指向顶了，那么循环回来，执行队列的第一个元素

if(rear+1 == maxSize){

rear = -1;

}

// 队尾指针加1，然后在队尾指针处插入新的数据

queArray[++rear] = value;

nItems++;

}

}

//移除

public Object remove(){

if(!isEmpty()) {

Object temp = queArray[front++];

nItems--;

if(front == maxSize)

front = 0;

return temp;

}

return null;

}

// 查看队列头数据

public Object peekFront() {

return queArray[front];

}

// 判断队列是否满了

public boolean isFull() {

return (nItems == maxSize);

}

// 判断队列是否为空

public boolean isEmpty() {

return (nItems == 0);

}

}

1. 设计一个单链表，从小到大存储元素[1,3,5,7,9]。另一个单链表，从小到大存储元素[2,4,6,8,10]。设计第3个链表，将前面2个链表的内容就归并，按照大小顺序排列，结果为[1,2,3,4,5,6,7,8,9,10]

**public class** Demo {  
  
 **public static void** main(String[] args) {  
 LinkedList<Integer> list1 = **new** LinkedList<>();  
 list1.addLast(1);  
 list1.addLast(3);  
 list1.addLast(5);  
 list1.addLast(7);  
 list1.addLast(9);  
  
 LinkedList<Integer> list2 = **new** LinkedList<>();  
 list2.addLast(2);  
 list2.addLast(4);  
 list2.addLast(6);  
 list2.addLast(8);  
 list2.addLast(10);  
  
 LinkedList<Integer> list3 = **new** LinkedList<>();  
 **while** (**true**){  
 **if**(list1.isEmpty() && !list2.isEmpty()){  
 list3.addLast(list2.getFirst());  
 list2.removeFirst();  
 **continue**;  
 }  
  
 **if**(!list1.isEmpty() && list2.isEmpty()){  
 list3.addLast(list1.getFirst());  
 list1.removeFirst();  
 **continue** ;  
 }  
  
 **if**(list1.isEmpty() && list2.isEmpty()){  
 **break**;  
 }  
  
  
 **int** data1 = list1.getFirst();  
 **int** data2 = list2.getFirst();  
  
 **if**(data1 < data2){  
 list3.addLast(data1);  
 list1.removeFirst();  
 }**else**{  
 list3.addLast(data2);  
 list2.removeFirst();  
 }  
 }  
  
 System.***out***.println(list3);  
 }  
}