第一周

1、用 add first 或 add last 这套新的 API 改写 Deque 的代码

import java.util.LinkedList;

public class Deque {

public static void main(String[] args) {

LinkedList<String> deque = new LinkedList<String>();

deque.addFirst("a");

deque.addFirst("b");

deque.addFirst("c");

System.out.println(deque.toString());

String str = deque.peek();

System.out.println(str);

System.out.println(deque.toString());

while (deque.size() > 0) {

System.out.println(deque.removeLast());

}

System.out.println(deque.toString());

}

}

2、分析 Queue 和 Priority Queue 的源码

优先级队列：PriorityQueue

构造方法

[PriorityQueue](https://docs.oracle.com/javase/10/docs/api/java/util/PriorityQueue.html#%3Cinit%3E())()

PriorityQueue​(int initialCapacity)

PriorityQueue​(int initialCapacity, Comparator<? super E> comparator)

PriorityQueue​(Collection<? extends E> c)

PriorityQueue​(Comparator<? super E> comparator)

PriorityQueue​(PriorityQueue<? extends E> c)

PriorityQueue​(SortedSet<? extends E> c)

常用功能函数：

add(E e) 添加元素

clear() 清空元素

contains() 包含

offer(E e) 添加元素

peek() 读取元素，不删除

poll() 读取元素，删除

remove(Object o) 删除指定元素

size() 返回长度

实例：

给程序员写的代码量排序

Programmer类

public class Programmer implements Comparable{

private int codeNum;

private String name;

public Programmer(int codeNum, String name) {

this.codeNum = codeNum;

this.name = name;

}

public int getCodeNum() {

return codeNum;

}

public void setCodeNum() {

this.codeNum = codeNum;

}

public String getName() {

return name;

}

public void setName() {

this.name = name;

}

public String toString(){

return "姓名：" + name + "-" + codeNum + "分";

}

@Override

public int compareTo(Object o) {

Programmer current = (Programmer) o;

if(current.getCodeNum() > this.getCodeNum()){

return 1;

}else if(current.getCodeNum()==this.getCodeNum()){

return 0;

}

return -1;

}

}

优先级队列排序

public class PriorityQueueTest {

public static void main(String[] args) {

final PriorityQueue<Programmer> queue=new PriorityQueue<>();

Programmer programmer1 = new Programmer(1000, "zhangsan");

Programmer programmer2 = new Programmer(2000, "lisi");

Programmer programmer3 = new Programmer(3000, "wangwu");

Programmer programmer4 = new Programmer(4000, "zhaoliu");

queue.add(programmer1);

queue.add(programmer2);

queue.add(programmer3);

queue.add(programmer4);

for (Programmer programmer: queue) {

System.out.println(programmer.toString());

}

System.out.println("-----------");

while (!queue.isEmpty()) {

System.out.println(queue.poll());

}

}

}

运行结果：

姓名：zhaoliu-4000分

姓名：wangwu-3000分

姓名：lisi-2000分

姓名：zhangsan-1000分

-----------

姓名：zhaoliu-4000分

姓名：wangwu-3000分

姓名：lisi-2000分

姓名：zhangsan-1000分

Process finished with exit code 0

3、感受

第一周，开始学习习惯建立，不足的地方，是题目还没有练习多次，这里应该加强下。