QueueElement.java:

```
package javaHomework4;

public class QueueElement {
	private int priority;
	private String data;
	public QueueElement(){
		priority = 0 ;
		data=new String("");
}
	public QueueElement(int p, String d){
		priority = p ;
		data = d ;
	}
	public void setPriority(int p) {
		priority = p ;
	}
	public void setData(String d) {
		data = d ;
	}
	public int getPriority() {
		return priority;
	}
	public String getData() {
		return data;
	}

}
```

PriorityQueue.java:

```
package javaHomework4;

public class PriorityQueue {
	private static final int MAX_BUFFER_SIZE = 100 ;
	private QueueElement[] buffer;
	private int queueLength ;
	public PriorityQueue(){
		queueLength = 0 ;
		buffer= new QueueElement[MAX_BUFFER_SIZE];
}
	public void initialize(){
		queueLength = 0 ; //初始化队列的下标
```

```
}
    public boolean isFull(){
          return queueLength == MAX_BUFFER_SIZE ;
   }

    public boolean isEmpty() {
          return queueLength == 0 ;
}
    public void insert(QueueElement d){
                // 确保队列未满
                if (isFull()){
                      System.out.println("queue overflow") ;
                      System.exit(1) ;
                   }
                //查找正确的插入位置
                int i = 0 ;
               //寻找插入点
               while ((i < queueLength) && (buffer[i].getPriority() <= d.getPriority())) {
                      i++ ;
               }
                for (int j = queueLength;j>i;j--){//i 是插入点
                      buffer[j] = buffer[j - 1];
                //插入
                }
                buffer[i] = new QueueElement(d.getPriority(), d.getData());

                if(queueLength==0) {
                     buffer[i]=d;//若队列中无内容则直接插入
                }
                // 调整队列长度
                queueLength++;

               }
     public QueueElement remove(){
                //检查是否是空队列
                if (isEmpty())
                      {
                      System.out.println("queue underflow") ;
                      System.exit(1) ;
                      }
                //存储队首元素
                QueueElement temp = buffer[0] ;
                for (int i = 0 ; i < queueLength - 1 ; i++) // 从第二个元素其依次向前移
                      buffer[i] = buffer[i + 1] ;
```

```
                //调整长度
                queueLength-- ;
                //返回队首元素
                return temp ;
                }
        public QueueElement first() {
                if (isEmpty()){
                    System.out.println("queue underflow") ;
                    System.exit(1) ;
                 }
                return new QueueElement(buffer[0].getPriority(), buffer[0].getData()) ;
                //元素 copy,返回新的对象的引用。避免了更改 queue 中元素的值
                }
        public int length(){
                return queueLength ;
                }
}
```

UsePriority.java:

```
package javaHomework4;

public class UsePriority {
    public static void main(String[] args) {
        // 创建 PriorityQueue 实例
        PriorityQueue queue = new PriorityQueue();
        // 插入 QueueElement 对象
        queue.insert(new QueueElement(2, "Mow the lawn"));
        queue.insert(new QueueElement(1, "Complete the programming assignment"));
        queue.insert(new QueueElement(4, "Read the next chapter of the book on Java"));
        queue.insert(new QueueElement(3, "Look at recipes for Saturday's party"));
        queue.insert(new QueueElement(2, "Cleam room"));
        System.out.println("队首元素为: "+queue.first().getPriority()+" "+queue.first().getData());
        System.out.println("队列长度为: "+queue.length());
        QueueElement temp=queue.remove();
        System.out.println("被移除的元素为:"+temp.getPriority()+" "+temp.getData());
        System.out.println("移除一次后的队首元素为: "+queue.first().getPriority()+" "+queue.first().getData());
        queue.remove();
        System.out.println("移除两次后的队首元素为"+queue.first().getPriority()+" "+queue.first().getData());
        System.out.println("移除两次后的队列长度为: "+queue.length());
```

```
    }
}
```

运行结果：

QueueElement.java | PriorityQueue.java | UsePriority.java

```
1  package javaHomework4;
2
3  public class UsePriority {
4      public static void main(String[] args) {
5          // 创建 PriorityQueue 实例
6          PriorityQueue queue = new PriorityQueue();
7
8          // 插入 QueueElement 对象
9          queue.insert(new QueueElement(2, "Mow the lawn"));
10         queue.insert(new QueueElement(1, "Complete the programming assignment"));
11         queue.insert(new QueueElement(4, "Read the next chapter of the book on Java"));
12         queue.insert(new QueueElement(3, "Look at recipes for Saturday's party"));
13         queue.insert(new QueueElement(2, "Cleam room"));
14         System.out.println("队首元素为："+queue.first().getPriority()+" "+queue.first().getData());
15         System.out.println("队列长度为："+queue.length());
16         QueueElement temp=queue.remove();
17         System.out.println("被移除的元素为："+temp.getPriority()+" "+temp.getData());
18         System.out.println("移除一次后的队首元素为："+queue.first().getPriority()+" "+queue.first().getData());
19         queue.remove();
20         System.out.println("移除两次后的队首元素为"+queue.first().getPriority()+" "+queue.first().getData());
21         System.out.println("移除两次后的队列长度为："+queue.length());
22     }
23 }
24
```

控制台 | 问题 | Debug Shell

```
<已终止> UsePriority [Java 应用程序] E:\Java\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.4.v20240802-155
队首元素为：1 Complete the programming assignment
队列长度为：5
被移除的元素为：1 Complete the programming assignment
移除一次后的队首元素为：2 Mow the lawn
移除两次后的队首元素为2 Cleam room
移除两次后的队列长度为：3
```