

자료구조 (Data Structure) 실습 과제

# Priority Queue

2024학년도 1학기

Instructor: Prof. Young-guk Ha  
Dept. of Computer Science & Engineering



## 실습 1-1. 다음의 Entry 인터페이스를 이용하여 MyEntry 클래스를 구현하고, Sorted List를 기반으로 MyPQ 클래스를 완성하시오

```
// Interface Entry
public interface Entry {
    public void setKey(Object k);
    public void setValue(Object v);
    public Object getKey();
    public Object getValue();
}

// Class MyPQ
import java.util.ArrayList; // Java ArrayList 이용
import java.util.Comparator; // Java Comparator 이용

public class MyPQ {
    // Constructors
    MyPQ(Comparator comp) { ... }
    MyPQ(int initialCapacity, Comparator comp) { ... }

    // Implement the following methods
    public int size() { ... }
    public boolean isEmpty() { ... }
    public MyEntry insert(Object k, Object v) { ... }
    public MyEntry removeMin() { ... }
    public MyEntry min() { ... }
}
```

※ 다음 페이지 계속

## 실습 1-2. Comparator 인터페이스를 이용하여 Integer를 비교하는 IntComparator 클래스를 구현하고, main 메소드를 실행하여 결과를 출력하시오

※ 실습 1-1과 1-2는 하나의 프로젝트로 작성할 것

// Class IntComparator (두개의 Integer 오브젝트를 비교)

```
import java.util.Comparator;
```

```
public class IntComparator implements Comparator {
```

```
    public int compare(Object o1, Object o2) { ... }
```

```
    // compare 메소드만 구현 (실습자료 뒤에 [별첨1] 참조)
```

```
}
```

// main 메소드 실행 결과 확인

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

```
    IntComparator c = new IntComparator();
```

```
    MyPQ pq = new MyPQ(c);
```

```
    pq.insert(new Integer(30), null);
```

```
    pq.insert(new Integer(10), null);
```

```
    pq.insert(new Integer(20), null);
```


```
    System.out.println((Integer)pq.removeMin().getKey());
```

```
    System.out.println((Integer)pq.removeMin().getKey());
```

```
    System.out.println((Integer)pq.removeMin().getKey());
```

```
}
```

화면  
출력



```
10  
20  
30
```

## 실습 2. MyPQ 클래스를 이용하여 다음 문제를 해결하는 Java 프로그램을 작성하시오

1) 2차원 평면상에 아래 그림과 같은 점들이 있을 때, 이 점들을 원점에서부터 가까운 순서대로 출력하는 프로그램을 작성하시오 (PointComparator 구현)

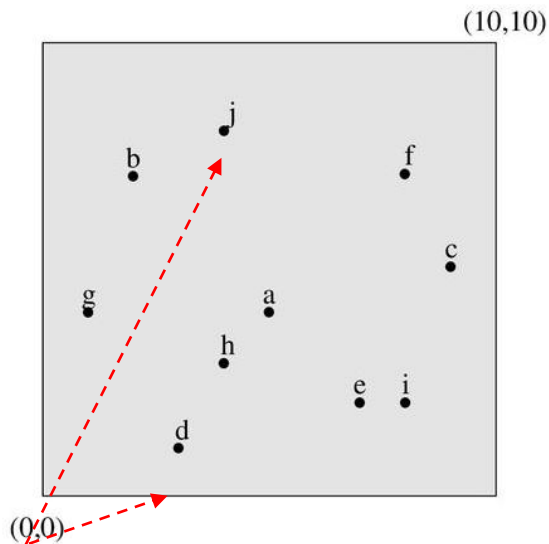
✓ 2차원 평면상의 거리: Euclidean Distance from  $(0, 0)$  to  $p(x, y) = \sqrt{p.x^2 + p.y^2}$

✓ 2차원 평면상의 점은 java.awt.Point 클래스를 사용하여 구현함 (실습자료 뒤에 [별첨2] 참조)

2) 2차원 평면상에 아래 그림과 같은 점들이 있을 때, 이 점들을 원점에서부터 멀리 있는 순서(위의 1번 문제와 반대 순서)대로 출력할 수 있도록 새로운 PointComparator2를 구현하고, 이를 이용하여 화면에 결과를 출력하시오

❖ 상기 1), 2)번 모두 하나의 Java 프로그램에서 출력하도록 할 것

점	위치
a	(5, 4)
b	(2, 7)
c	(9, 5)
d	(3, 1)
e	(7, 2)
f	(9, 7)
g	(1, 4)
h	(4, 3)
i	(8, 2)
j	(4, 8)



화면  
출력

[가까운 순서]

d (3, 1)

g (1, 4)

...

[멀리 있는 순서]

f (9, 7)

c (9, 5)

...

## [별첨1] Interface java.util.Comparator Reference

Method Summary	
int	<b><u>compare</u></b> ( <a href="#">Object</a> o1, <a href="#">Object</a> o2) Compares its two arguments for order.
boolean	<b><u>equals</u></b> ( <a href="#">Object</a> obj) Indicates whether some other object is "equal to" this Comparator.

## [별첨2] Class java.awt.Point Reference

Constructor Summary	
<a href="#">Point()</a>	Constructs and initializes a point at the origin (0, 0) of the coordinate space.
<a href="#">Point(int x, int y)</a>	Constructs and initializes a point at the specified (x, y) location in the coordinate space.
<a href="#">Point(Point p)</a>	Constructs and initializes a point with the same location as the specified <code>Point</code> object.
Method Summary	
boolean	<a href="#">equals(Object obj)</a> Determines whether two points are equal.
<a href="#">Point</a>	<a href="#">getLocation()</a> Returns the location of this point.
double	<a href="#">getX()</a> Returns the X coordinate of the point in double precision.
double	<a href="#">getY()</a> Returns the Y coordinate of the point in double precision.
int	<a href="#">hashCode()</a> Returns the hashCode for this point.
void	<a href="#">move(int x, int y)</a> Moves this point to the specified location in the (x, y) coordinate plane.
void	<a href="#">setLocation(double x, double y)</a> Sets the location of this point to the specified float coordinates.
void	<a href="#">setLocation(int x, int y)</a> Changes the point to have the specified location.
void	<a href="#">setLocation(Point p)</a> Sets the location of the point to the specified location.
<a href="#">String</a>	<a href="#">toString()</a> Returns a representation of this point and its location in the (x, y) coordinate space as a string.
void	<a href="#">translate(int x, int y)</a> Translates this point, at location (x, y), by dx along the x axis and dy along the y axis so that it now represents the point (x + dx, y + dy).