

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package hw06;
```

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

```
public class HW06 {
```

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

```
        driver1();
        driver2();
        driver3();
```

```
    }
```

```
    private static void driver1() {
```

```
        Comparator comp = (Comparator<Integer>) new Comparator<Integer>() {
```

```
            @Override
```

```
            public int compare(Integer o1, Integer o2) {
                return (o1.compareTo(o2));
            }
```

```
        };
```

```
        GITPriorityQueue git = new GITPriorityQueue(comp);
```

```
        git.offer(new Integer("7"));
        git.offer(new Integer("6"));
        git.offer(new Integer("5"));
        git.offer(new Integer("4"));
        git.offer(new Integer("3"));
        git.offer(new Integer("2"));
        git.offer(new Integer("1"));
        git.offer(new Integer("0"));
```

```
        System.out.println(git.toString());
        System.out.println("remove() => " + git.remove());
        System.out.println(git.toString());
        System.out.println("element() => " + git.element().toString());
        System.out.println("-----");
    }
```

```
    private static void driver2() {
```

```
        Comparator comp = (Comparator<Double>) new Comparator<Double>() {
```

```
            @Override
```

```
            public int compare(Double o1, Double o2) {
                return (o1.compareTo(o2));
            }
```

```
        };
```

```
        GITPriorityQueue git = new GITPriorityQueue(comp);
```

```
        git.offer(new Double("7.2"));
        git.offer(new Double("7.1"));
        git.offer(new Double("7.6"));
        git.offer(new Double("4.8"));
        git.offer(new Double("4.2"));
        git.offer(new Double("4.3"));
        git.offer(new Double("2.2"));
```

```

        git.offer(new Double("2.1"));

        System.out.println(git.toString());
        System.out.println("remove() => " + git.remove());
        System.out.println(git.toString());
        System.out.println("element() => " + git.element().toString());
        System.out.println("-----");
    }

    private static void driver3() {
        Comparator comp = (Comparator<String>) new Comparator<String>() {

            @Override
            public int compare(String o1, String o2) {
                return (o1.compareTo(o2));
            }
        };

        GITPriorityQueue git = new GITPriorityQueue(comp);

        git.offer("GYTE");
        git.offer("Yusuf");
        git.offer("Sinan");
        git.offer("Akgül");

        System.out.println(git.toString());
        System.out.println("remove() => " + git.remove());
        System.out.println(git.toString());
        System.out.println("element() => " + git.element().toString());
    }
}

```