

Platform-based Programming

#5 Interface

2019년 2학기

Program Output

- ❖ Make a Java program that manage Circle and Line

```
Enter Operation String! addC
10 20 10
[[10, 20] 10 314]
Enter Operation String! addL
10 10 20 40
[[10, 10] [20, 40] 32]
Enter Operation String! list
[[[10, 20] 10 314], [[10, 10] [20, 40] 32]]
Enter Operation String! sortA
Enter Operation String! list
[[[10, 10] [20, 40] 32], [[10, 20] 10 314]]
Enter Operation String! clear
Enter Operation String! LIST
[]
```

Problem Output

```
Enter Operation String! Addl
10 10 20 30
[[10, 10] [20, 30] 22]
Enter Operation String! Addc
10 20 30
[[10, 20] 30 2827]
Enter Operation String! List
[[[10, 10] [20, 30] 22], [[10, 20] 30 2827]]
Enter Operation String! Sortd
Enter Operation String! List
[[[10, 20] 30 2827], [[10, 10] [20, 30] 22]]
Enter Operation String! quit
Bye
```

Program Skeleton

```
public class SortInterfaceTest {  
  
    private static Scanner scanner = new Scanner(System.in);  
    private static List<MyComparable> comparableList = new ArrayList<>() ;  
  
    public static void main(String[] args) {  
        while ( true ) {  
            final OperationKind op = getOperation(scanner) ;  
            if ( op == OperationKind.QUIT ) {  
                System.out.println("Bye") ;  
                break;  
            }  
            if ( op == OperationKind.INVALID ) {  
                System.out.println("Invalid Operation!") ;  
                continue ;  
            }  
        }  
    }  
}
```

```

switch ( op ) {
    case ADDL : {
        final Line newLine = createLine(scanner) ;
        comparableList.add(newLine);
        System.out.println(newLine) ;    [[10, 10] [20, 40] 32]
        break ;
    }
    case ADDC : {
        final Circle newCircle = createCircle(scanner) ;
        comparableList.add(newCircle);
        System.out.println(newCircle) ;  [[10, 20] 10 314]
        break ;
    }
    case SORTA:
        sortList(comparableList, SortKind.ASCENDING) ;
        break ;
    case SORTD:
        sortList(comparableList, SortKind.DECENDING) ;
        break ;
    case CLEAR:
        comparableList.clear() ;
        break ;
    case LIST:
        System.out.println(comparableList) ;
        break ;
    case default: break;    [[[10, 10] [20, 40] 32], [[10, 20] 10 314]]
}
}
}

```

// Point.java

```
public class Point {  
    private final int x, y ;  
    ...  
}
```

// MyComparable.java

```
public interface MyComparable {  
    public int compareTo(final MyComparable other) ;  
    public long getSize() ;  
}
```

// Line.java

```
public class Line implements MyComparable {  
    private final Point point1, point2 ;  
    ...  
}
```

// Circle.java

```
public class Circle implements MyComparable {  
    private final Point center ;  
    private final int radius ;  
    ...  
}
```