

Object-Oriented programming and design

Laboratory work #5 (3 pts)

Interfaces

Deadline: week 10

#1

- a. When to use an Interface vs when to use an abstract class. For each “when” provide extended example(s) (with class/interface codes).
- b. Suppose you have an interface `Moveable`. Think of some interface that can extend it. Implement this two interfaces.

#2

Extend `Employee` and `Manager` classes created in lab#3.

- Replace field `year` by the field `hireDate` of type `java.util.Date`
- Your classes should implement `Comparable` interface. (`Employee1 > Employee2` if its salary is more than the salary of `Employee2`, the same for managers, but if their salaries are equal, compare by `bonus`).

#3

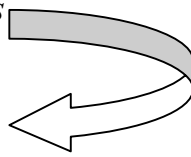
A collection represents a group of objects, known as its elements. Some collections allow duplicate elements and others do not. Some are ordered and others unordered. Create an Interface `MyCollection` which is maximum general (abstract) collection possible.

#4

You need to write a class `MinMax` with a method `minmax` that takes an array of integers as a parameter and returns min and max simultaneously (using one method and one call).

Hint: use inner class

```
public class MinMax {  
    static class ??? {  
  
    }  
    static ??? minmax(int values[]){  
  
        return ???;  
    }  
}
```



Test class:

```
int a[] = {0, 8 , -3, 20};  
MinMax m = new MinMax();  
// Do something to find min and max using instance m of  
class MinMax
```