Manavendra Sen | 199302058 | IT 3A

OOP LAB 5

EXERCISE 2.4

Make a class Employee with attributes
–name:String
–salary: double.

This class supplies
(i) A parameterized constructor
(ii) Accessor and Mutator method(s) for every instance field and
(iii) toString() method which returns the values of instance fields by adding proper heading labels and spaces.

Make a class Manager that inherits from Employee and add an instance field name
– department:String.

This class also supplies parameterized constructor, accessor and mutator methods and a toString() method that returns the manager's name, department, and salary by adding proper labels and spaces.

```
class Employee {
 private String name;
 private double salary;

 // Parameterized constructor
 Employee(String name, double salary) {
   this.name = name;
   this.salary = salary;
 }

 public String getName() {
   return name;
 }

 public double getSalary() {
   return salary;
 }
```

```
 public void setName(String name) {
   this.name = name;
 }

 public void setSalary(double salary) {
   this.salary = salary;
 }

 // Overriding the toString method of object class
 @Override
 public String toString() {
   return ("Employee Name: " + this.getName() + "\n" + "Salary: " +
this.getSalary());
 }
}

// Subclass
class Manager extends Employee {
 private String department;

 Manager(String name, double salary, String department) {
   // Every time an object of a subclass will be created we need the values
from its super/parent class also, so we call the super method
   super(name, salary);
   this.department = department;
 }

 public String getDepartment() {
   return department;
 }

 public void setDepartment(String department) {
   this.department = department;
 }
```

```java
 @Override
 public String toString() {
   // We call the to string method of its super and concatenate to it
   return super.toString() + "\n" + "Department: " + this.getDepartment();
 }
}

class Driver {
 public static void main(String[] args) {
   Employee e1 = new Employee("Felix", 20000.0);
   Employee e2 = new Employee("Mark", 18000.0);
   Manager m1 = new Manager("Sean", 50000.0, "Video Production");
   Manager m2 = new Manager("Patrick", 25000.0, "Security");
   // Displaying the details in the console
   System.out.println("-- Employee Details --");
   System.out.println(e1 + "\n");
   System.out.println(e2 + "\n");
   System.out.println(m1 + "\n");
   System.out.println(m2 + "\n");
 }
}
```

```
mavn:Java/ $ javac LAB5_1.java          [0:35:36]
mavn:Java/ $ java Driver                [0:35:37]
-- Employee Details --
Employee Name: Felix
Salary: 20000.0

Employee Name: Mark
Salary: 18000.0

Employee Name: Sean
Salary: 50000.0
Department: Video Production

Employee Name: Patrick
Salary: 25000.0
Department: Security

mavn:Java/ $                            [0:35:44]
```

EXERCISE 1.2

Compile and Execute the following code by completing it as per commented specification given. Write the whole code.

```
class A {
 public int a = 100;
} // End of class A

class B extends A {
 public int a = 80;
} // End of class B

class C extends B {
 public int a = 60;
} // End of class C

class D extends C {
 public int a = 40;
} // End of class D

class E extends D {
 public int a = 10;

 public void show() {
   int a = 0;
   System.out.println(a);
   System.out.println(this.a);
   System.out.println(super.a);
   // Question: Write Java statements to display the values of
   // all a's used in this file on System.out
   // to access the public "a" of class A, B, C we can't use super
   // we may try to typecast the this reference ex -> (A) this.a ,
   System.out.println(((C) this).a);
   System.out.println(((B) this).a);
   System.out.println(((A) this).a);
 }
```

```java
}

class Main {
 public static void main(String args[]) {
   new E().show();
   A a1 = new E();
   D d1 = (D) a1;
 }
}
```

```
mavn:Java/ $ javac LAB5_3.java                    [15:50:32]
mavn:Java/ $ java Main                            [15:50:34]
0
10
40
60
80
100
mavn:Java/ $                                      [15:50:39]
```