

TCP1201 Object-Oriented Programming and Data Structures

Lab06 Abstract Classes and Interfaces

Exercise 1: Abstract Class

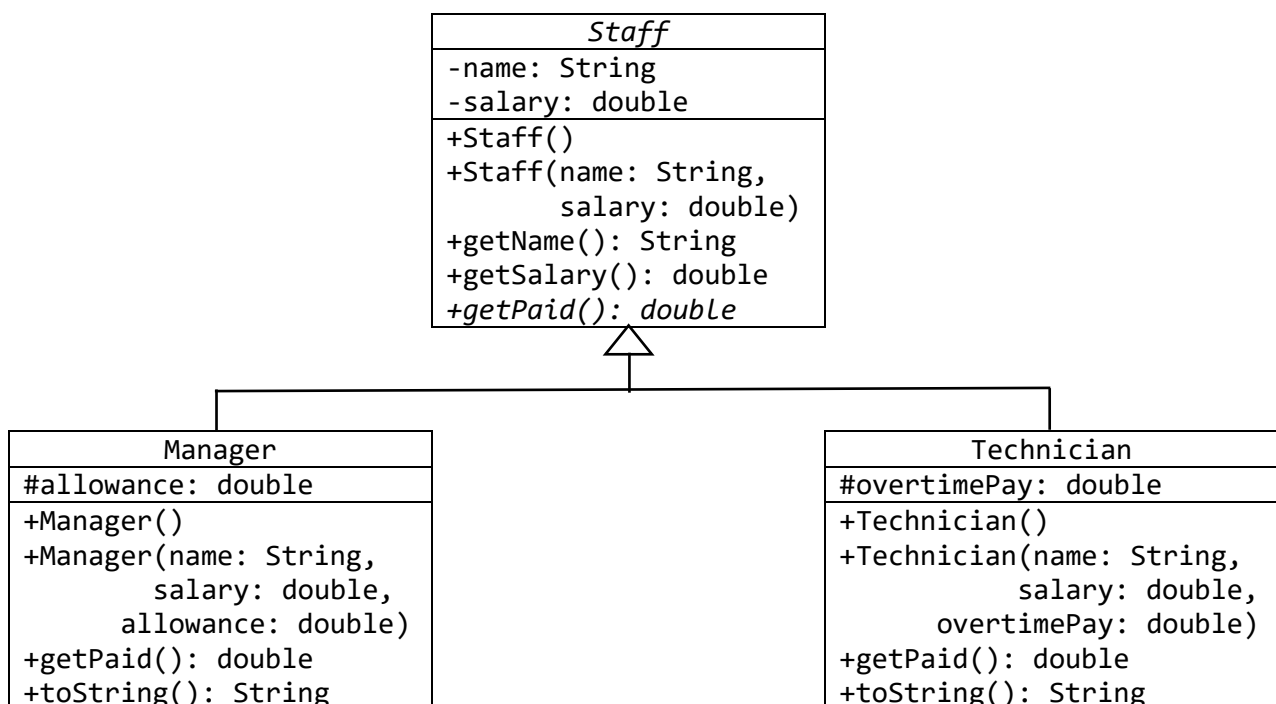
The following UML Class Diagram is provided. Implement all the classes in the Class Diagram.

The `getPaid()` methods return the total paid as follows:

1. For Manager, returns the sum of salary and allowance.
2. For Technician, returns the sum of salary and overtimePay.

The `toString()` methods return the following String:

1. For Manager, returns name, salary, allowance, and total paid in a String.
2. For Technician, returns name, salary, overtimePay, and total paid in a String.



If the 3 classes are implemented correctly, the first statement in the following main method would generate a compile error. Study the error. What generates the error?

```
public static void main (String[] args) {
    Staff s = new Staff ("Alex", 1000);
    Staff m = new Manager ("Siti", 4000, 1500);
    System.out.println (m);
    Staff t = new Technician ("Ali", 2000, 1200);
    System.out.println (t);
}
```

Comment out the first statement, re-compile and run the program. It should produce the following output.

Sample run:

Manager: name = Siti, salary = 4000.0, allowance = 1500.0, paid = 5500.0

Technician: name = Ali, salary = 2000.0, overtimePay = 1200.0, paid = 3200.0

Exercise 2: Comparable Interface

Update the Staff inheritance hierarchy in Exercise 1 so that we can use `java.util.Arrays.sort` method to sort all objects in the inheritance hierarchy. A correct implementation makes the following main method produce the output as shown in sample run. Note that the staffs are sorted by paid in ascending order.

```
public static void main (String[] args) {
    Staff[] staffArray = { new Technician ("Ali", 2000, 1200),
                           new Manager ("Siti", 4000, 800),
                           new Technician ("Abu", 3000, 2000) };

    // 1. Add code to sort and and print staffArray.

    // Create an ArrayList from staffArray.
    ArrayList<Staff> staffList = new ArrayList<>(Arrays.asList(staffArray));

    // 2. Add code to sort and print staffList.

}
```

Sample run:

Array:

Technician: name = Ali, salary = 2000.0, overtimePay = 1200.0, paid = 3200.0

Manager: name = Siti, salary = 4000.0, allowance = 800.0, paid = 4800.0

Technician: name = Abu, salary = 3000.0, overtimePay = 2000.0, paid = 5000.0

ArrayList:

Technician: name = Ali, salary = 2000.0, overtimePay = 1200.0, paid = 3200.0

Manager: name = Siti, salary = 4000.0, allowance = 800.0, paid = 4800.0

Technician: name = Abu, salary = 3000.0, overtimePay = 2000.0, paid = 5000.0