HW7

Requirements:

- Create a Java project named yourGroupId_OOP_HW7
- Read instructions and create classes needed. You are supposed to add 1 interface and 5 classes (*Analyzer*, *Employee, Manager, Company, WageAnalyzer, OvertimeAnalyzer*) to the project.
- All instance variables are private. Please use public methods to access private instance variables.

Description:

In a company, there are often many employees and supervisors. Both employees and supervisors have records of working hours and overtime hours to determine salary payment. To let HR understand the current situation of salary payment and overtime in the company, some companies will purchase relevant analysis software to assist. In this example, suppose that a software company provides two types of analyzers, named *WageAnalyzer*, and *OvertimeAnalyzer*, and provides an interface to connect the interaction between the company and the analyzers. This allows the buying company to more effectively understand the status of salary payment. Figure 1 describes the relationship between each class or interface.

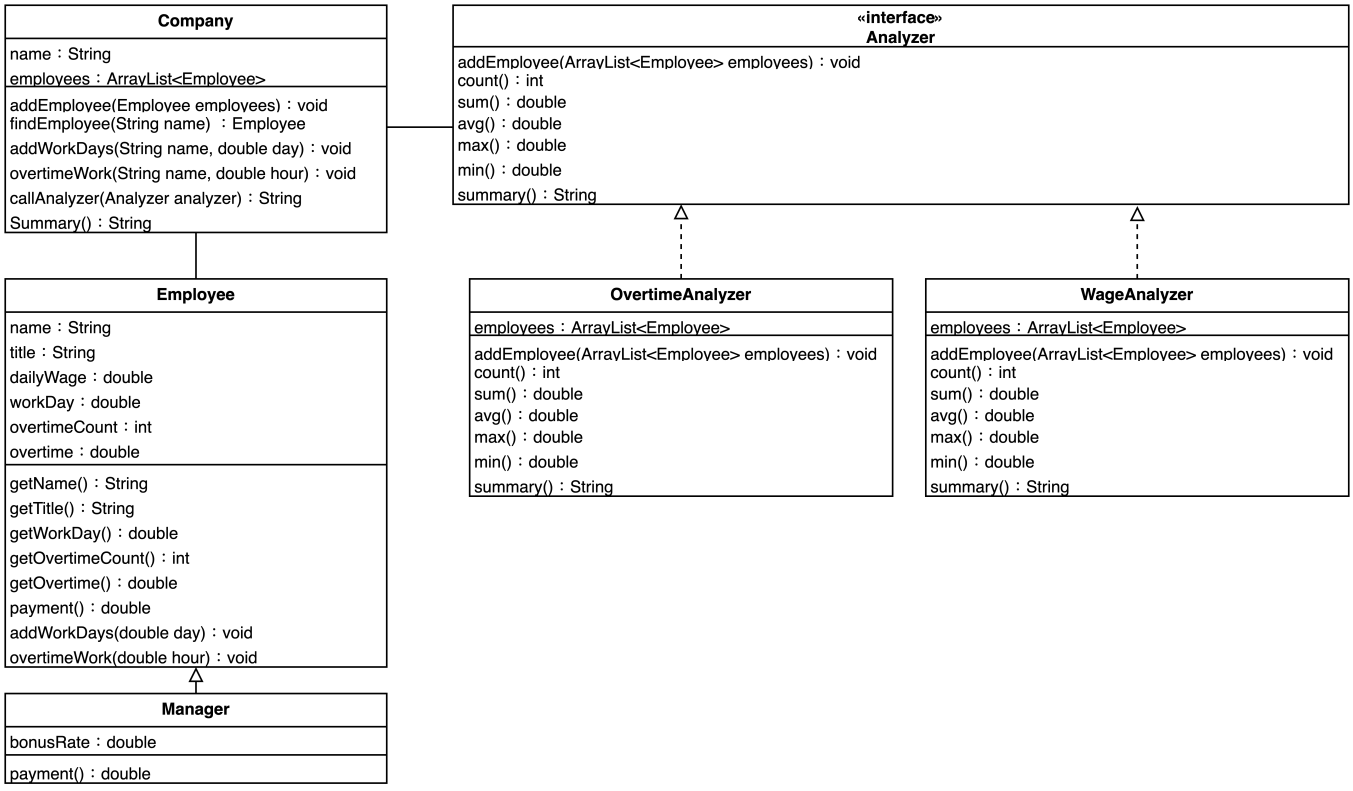

Figure 1. The UML diagram of the question

1. Create **Employee** class

Employee	
Modifier and type	**Method (or Variable) and description**
Instance variable	
String	name The name of the employee.
String	title The title of the employee.
double	dailyWage The basic daily wage of the employee.
double	workDay The actual work day(s).
int	overtimeCount The number the employee works overtime.
double	overtime Total hours of working overtime.
Constructor	
Employee (String name, String title, double dailyWage) Enable to instantiate the object of *Employee* with given *name, title, dailyWage*.	
Instance methods	
-	5 getter for 5 attributes (*getName(), getTitle (), getWorkDay(), getOvertimeCount(), getOvertime()*).
double	payment() According to the following formula to calculate the wage and return the result. ***(Wage = dailyWage * workDay + overtime * 150)***
void	addWorkDays(double day) Add workday(s) by given day(s).
void	overtimeWork(double hour) Add overtime hour(s) by given hour(s), and increase *overtimeCount* by 1.

2. Create **Manager** class

Manager	
Modifier and type	**Method (or Variable) and description**
Instance variable	
double	bonusRate The manager can get extra wage (rate).
Constructor	
Manager (String name, String title, double dailyWage, double bonusRate) Enable to instantiate the object of *Manager* with given *name, title, dailyWage, and bonusRate*.	

Instance methods	
double	payment() According to the following formula to calculate and return the employee's wage. ***(Wage = (dailyWage * workDay + overtime * 150) * bonusRate).***

3. Create **Company** class

Company	
Modifier and type	**Method (or Variable) and description**
Instance variable	
String	name The company's name.
ArrayList <Employee>	employees An ArrayList contains all employees in the company.
Constructor	
Company(String name) Enable to instantiate the object of *Company* with given *name* and initialize the *Employees ArrayList*.	
Instance methods	
void	addEmployee(Employee employee) Add the employee to the *Employee ArrayList*.
Employee	findEmployee(String name) Find the employee object in employees by the employee's name. If found, return the employee object. Otherwise, return null.
void	addWorkDays(String name, double day) Add an employee's workday(s) by given employee's name and day(s). You must determine whether there is such an employee in the company and if there is one, execute the corresponding action. Otherwise, print *"The employee is not found."*
void	overtimeWork(String name, double hour) Add an employee's overtime hour(s) by given employee name and hour(s). You must determine whether there is such an employee and if there is one, execute the corresponding action. Otherwise, print *"The employee is not found."*
String	callAnalyzer(Analyzer analyzer) Invoke the *addEmployee()* of the analyzer to add analysis data and returns the summary result.
String	summary() Return a formatted String text about each employee's name, work day, overtime count, overtime hour(total), wage, and title. The return value as following example.

Example:

```
Company:      TSMC
      Name  Work Day Overtime Count Overtime Hour(Total)      Wage       Title
       Bob      6.00              2                 8.00   8400.00       staff
```

4. Create **Analyzer** interface

<<Interface>> **Analyzer**	
Modifier and type	**Method (or Variable) and description**
Abstract methods	
void	addEmployee(ArrayList<Employee> employees) Abstract method
int	count() Abstract method
double	sum() Abstract method
double	avg() Abstract method
double	max() Abstract method
double	min() Abstract method
String	summary() Abstract method

5. Create **WageAnalyzer** class

WageAnalyzer	
Modifier and type	**Method (or Variable) and description**
Instance variable	
ArrayList<Employee>	employees An ArrayList contains all employees.
Constructor	
WageAnalyzer () Initialize the *employees ArrayList*.	
Instance methods	
void	addEmployee(ArrayList<Employee> employees) Add all employees to the *employees ArrayList*.
int	count() Return the number of employees.
double	sum() Return the sum of all wage.
double	avg() Return the average of the wage.
double	max() Return the max value of the wage.

double	min() Return the min value of the wage.

String	summary()
	Return a formatted String text about the analysis of the wage of the company, including the number of employees, total wage, average wage, minimum wage, and maximum wage. The return value as following example.
	Example:
	`Total employees:  3`
	`Total wage: 29775.00`
	`Average wage:  9925.00`
	`Max wage: 11925.00`
	`Min wage:  8400.00`

Test the wage analyzer object.

6. Create **OvertimeAnalyzer** class

OvertimeAnalyzer	
Modifier and type	**Method (or Variable) and description**
Instance variable	
ArrayList<Employee>	employees An ArrayList contains all employees who work overtime.
Constructor	
OvertimeAnalyzer () Initialize the *employees ArrayList*.	
Instance methods	
void	addEmployee(ArrayList<Employee> employees) Add all employees who work overtime to the *employees ArrayList*.
int	count() Return the number of employees who work overtime.
double	sum() Return the sum of all overtime hours.
double	avg() Return the average of the overtime hours.
double	max() Return the max value of the overtime hours.
double	min() Return the min value of the overtime hours.
String	summary() Return a formatted String text about the analysis of the working overtime of the company, including the number of working overtime employee, total hours, average hours, minimum hours, and maximum hours. The return value as following example.

Example:

```
     Total employees:        2
         Total hours:    13.00
       Average hours:     6.50
           Max hours:     8.00
           Min hours:     5.00
```

Tester

```
import java.util.ArrayList;

public class Tester {

	public static void main(String[] args) {
		// TODO Auto-generated method stub

		Company companyA = new Company("TSMC");

		companyA.addEmployee(new Employee("Bob", "staff", 1200));
		companyA.addEmployee(new Employee("Lily", "staff", 1350));
		companyA.addEmployee(new Manager("Andy", "supervisor", 1200, 1.5));

		companyA.addWorkDays("Bob", 6);
		companyA.overtimeWork("Bob", 5);
		companyA.overtimeWork("Bob", 3);

		companyA.addWorkDays("Lily", 7);

		companyA.addWorkDays("Andy", 6);
		companyA.overtimeWork("Andy", 5);

		Analyzer wageAnalyzer = new WageAnalyzer();
		Analyzer overtimeAnalyzer = new OvertimeAnalyzer();

		System.out.println("<<summarizeWage>>");
		System.out.println(companyA.summary());

		System.out.println();

		System.out.println("<<Wage analyzer>>");
		System.out.println(companyA.callAnalyzer(wageAnalyzer));

		System.out.println();

		System.out.println("<<Overtime analyzer>>");
		System.out.println(companyA.callAnalyzer(overtimeAnalyzer));
	}
}
```

Output	Format
<<summarizeWage>> Company: TSMC Name Work Day Overtime Count Overtime Hour(Total) Wage Title Bob 6.00 2 8.00 8400.00 staff Lily 7.00 0 0.00 9450.00 staff Andy 6.00 1 5.00 11925.00 supervisor <<Wage analyzer>> Total employees: 3 Total wage: 29775.00 Average wage: 9925.00 Max wage: 11925.00 Min wage: 8400.00 <<Overtime analyzer>> Total employees: 2 Total hours: 13.00 Average hours: 6.50 Max hours: 8.00 Min hours: 5.00	10 10 10 10 15 21 10 12 10 10 15 21 10 12 21 9 21 9

Submission: Submit your project as **".zip file"** via Moodle. No other submissions will be graded.

Reminder: Please zip the whole project.

Deadline: 2021/03/22 (Mon56) or 2021/03/23 (for Tue23)