ASSIGNMENT 4b

Implement a class named *Company* that stores information about the employees of a company. The element used for storing the employees is a class that implements the *Collection* interface. Every employee has a name and a unique personal ID (i.e. “cod numeric personal”). The employees of the company can be permanent and hourly payed employees. Permanent employees are paid a flat salary that is set via the constructor of the class and stored inside the class. The employees that are hourly paid have an attribute for storing the worked hours within a month as well as an attribute for storing the rate corresponding to an worked hour. The rate is set via the constructor and the number of the worked hours is set by a suitable method. The sum earned by an employee during the current month is returned by the method *public double getSalary()*.

The class *Company* provides the following services:

(a) a method for adding a new employee. The unique parameter of this method is a reference to the employee that is going to be kept inside; if an employee with the same name and personal ID like the one referenced by the parameter is already stored, the method that called this service is informed that the employee is already stored.

(b) a suitable method for printing the employees belonging to the company. For each employees its name, personal ID and the earned amount of money will be printed.

(c) a method *public boolean isInCompany(Strategy strategy),* where *Strategy* is an interface that contains the method *public boolean isCondition(Employee employee)*. The method *isInCompany* returns true if the company contains at least an employee for which the method *isCondition* defined in the strategy returns true. In the method an iteration through all the existing employees is performed and if at least an employee satisfying the condition exists (the method *isCondition* is called for the current employee), the method *isInCompany* returns true; otherwise, the method return false.

Requirements:

1. Implement according to the mentioned characteristics the *Company* class together with the classes and subclasses mentioned. Besides the mentioned characteristics, you are allowed to add additional services (if needed).
2. Define two classes that implement the Strategy interface as it follows:

a. a class that has a constructor with a String parameter denoting a name that is stored internally. In this case the method *public boolean isCondition(Employee employee)* returns true if the name kept inside is the same as the name of the employee referenced by the parameter *employee*.

b. a class for which the method *public boolean isCondition(Employee employee)* returns true if the amount earned by the employee during the current month is 1000.

3. Instantiate within a main method a company. Call all the methods provided by the instantiated object.
4. Explain which are the advantages of polymorphic calls in this application.