Question

In this task, you are required to model a company structure where Alphabet is the parent company, and Google, YouTube, and DeepMind are its child companies. The employee database is shared between all companies. When an employee joins any of the child companies, they are added to both the child company's list and the parent company's list. If an employee join parent company, they get added only to the parent company.

Task :

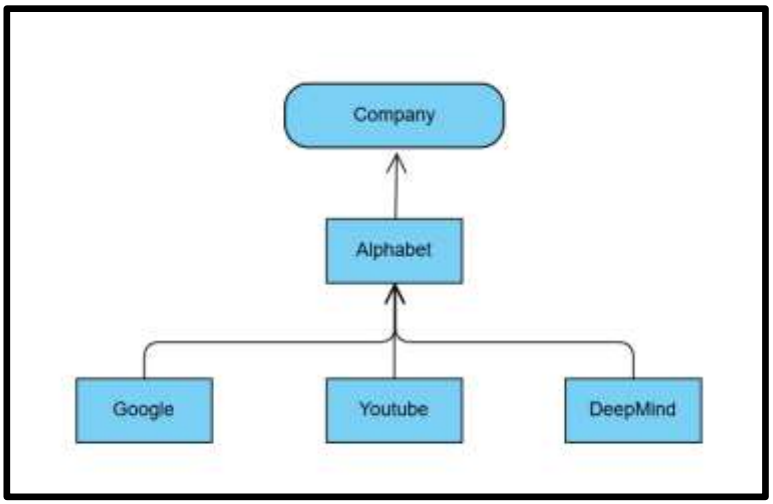
1. Abstract Class `**Company**`:
Implement an abstract class `**Company**` that contains a `**private String[4][10]**` employee database, where each row represents a company's employee list(Also the company ID):
   - Row 0: Alphabet employees
   - Row 1: Google employees
   - Row 2: YouTube employees
   - Row 3: DeepMind employees

   The `**Company**` class should also have Abstract methods:
   - `addEmployee(String employeeName)` to add an employee to the company's list.
   - `getEmployees()` to retrieve the list of employees for a specific company.
   - Getter and setter methods for `companyId`.

2. Concrete Classes:
   Create concrete classes for each company all extending `Company`. Each class should:
   - Implement the abstract methods from the `Company` class.
   - Ensure employees added to a child are also added to the parent company
   - Prevent duplicate entries for the same employee within a company's list.

3. Main Class:
   - Create instances of `Alphabet`, `Google`, `YouTube`, and `DeepMind`.
   - Add employees to the respective companies:
   - Employee "A" joins Alphabet.
   - Employees "B" and "A" join Google.
   - Employees "B" and "C" join YouTube.
   - Employee "D" joins DeepMind.
   - After adding the employees, print out the employee lists for each company.

Example:
If the following employees are added:
1. `Alphabet.addEmployee("A")`
2. `Google.addEmployee("A")`
3. `Google.addEmployee("B")`
4. `YouTube.addEmployee("B")`
5. `YouTube.addEmployee("C")`
6. `DeepMind.addEmployee("D")`



The output should be:

Employee list of Alphabet - A, B, C, D
Employee list of Google - A, B
Employee list of YouTube - B, C
Employee list of DeepMind - D

EVALUATION GUIDELINES:

1. Syntax and Code Structure

- Correct Java syntax and no errors.
- Proper naming conventions

2. Use of Access Modifiers

- Proper use of private, protected, and public to control access.
- Sensitive data like employee database is well-encapsulated and should not be directly accessible by any child company.

3. OOP Principles

- Correct use of abstraction, with abstract methods implemented in child classes.
- Inheritance used to share functionality between parent and child companies.
- Encapsulation through getter and setter methods.

4. Functionality

- Program meets all task requirements and adds employees correctly.
- Prevents duplicate employee entries and prints correct output.
- Complies properly and don’t have any errors

SUBMISSION GUIDELINES:

Submit **.java** file on BB. **Email Submissions will not be accepted. The hard deadline to submit is 2:55 PM.**