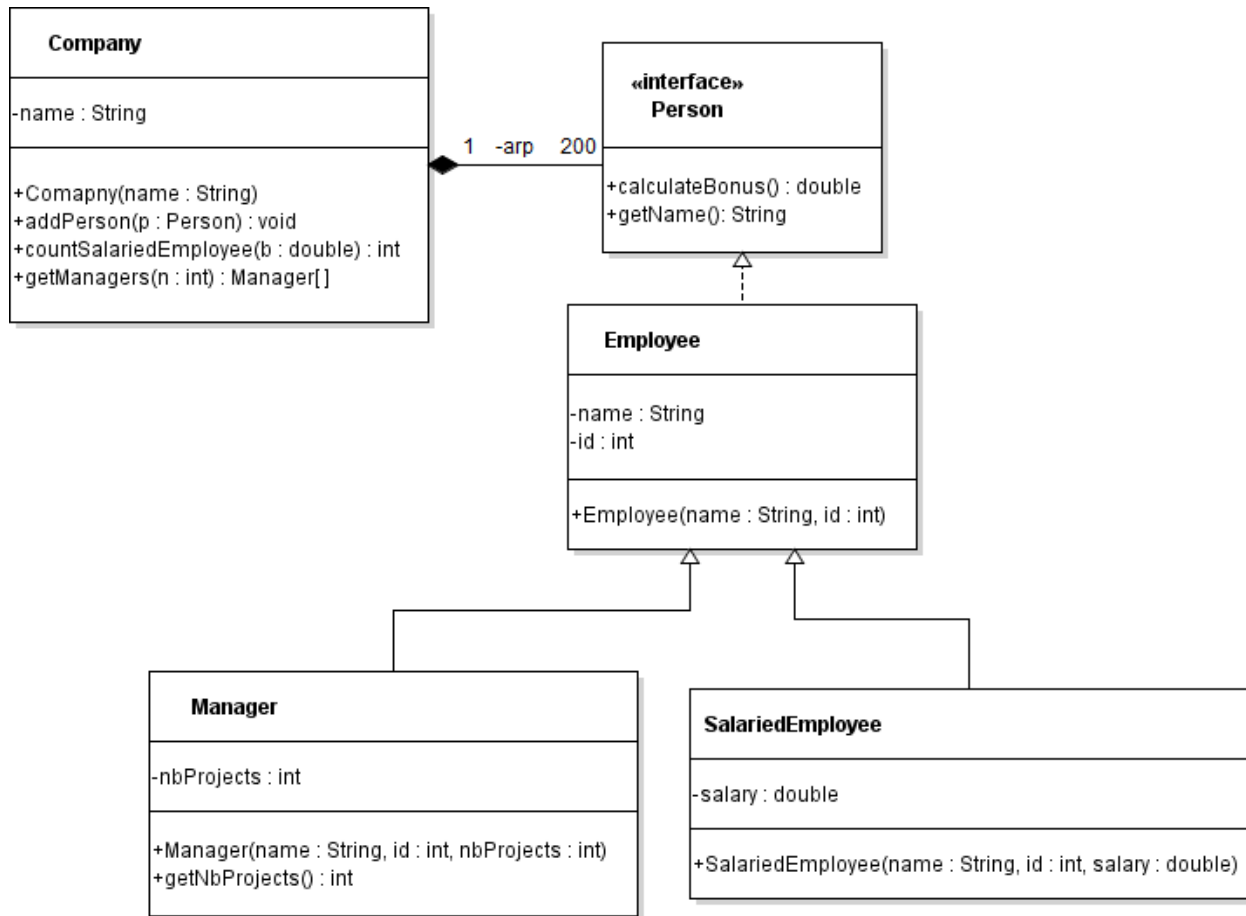


Exercise



Interface Person:

- METHODS:
 - *calculateBonus* (): calculated as:
 - for *Manager* : $\text{Bonus} = \text{nbProjects} * 10,000$
 - for *SalariedEmployee*: $\text{Bonus} = \text{salary} * 2$
 - *getName* (): returns the name of the employee.

Employee class

- METHODS:
 - *Employee* (name: String, id : int): constructor.

Manager class

- METHODS:
 - *Manager* (name: String, id : int, nbProjects : int): constructor.
 - *getNbProjects*(): getter for attribute nbProjects.

SalariedEmployee class

○ METHODS:

- ***SalariedEmployee (name: String, gpa : double)***: constructor.

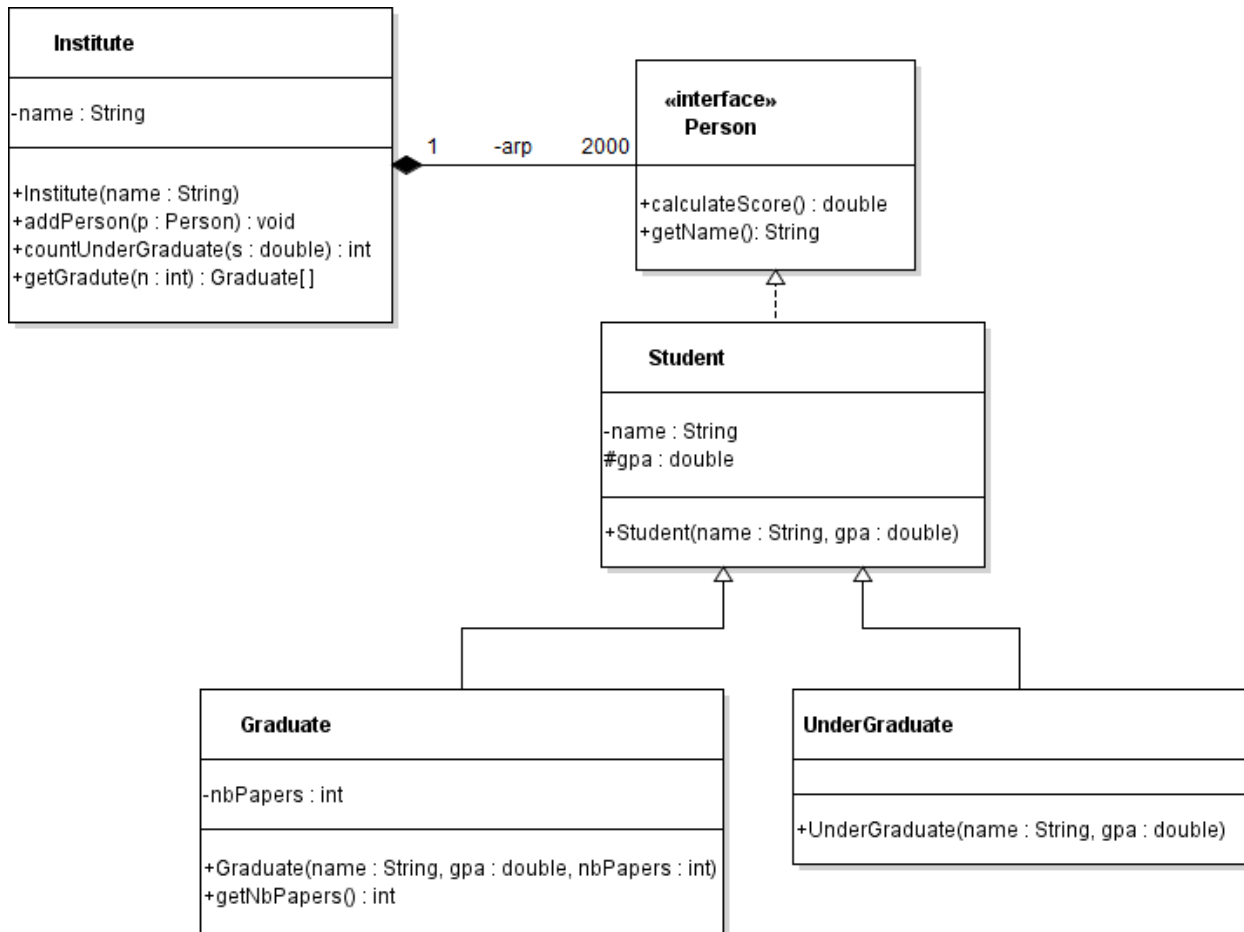
Company class

○ METHODS:

- ***Company (name: String)***: constructor.
- ***addPerson(p: Person)***: add a person to the company.
- ***countSalariedEmployee(b : double)***: count the number of **SalariedEmployee** in the company with bonus greater than b.
- ***getManagers(n : int)***: this method will return an array containing all the managers with number of projects greater than n.

QUESTION: Translate into Java code the class *Manager and Company*.

Exercise



Interface Person:

- METHODS:
 - *calculateScore* (): calculated as:
 - for *Graduate* : $\text{Score} = \text{nbPapers} * \text{gpa}$
 - for *UnderGraduate*: $\text{Score} = \text{gpa} * 3 + 5$
 - *getName* (): returns the name of the student.

Student class

- METHODS:
 - *Student* (name: String, gpa : double): constructor.

Graduate class

- METHODS:
 - *Graduate* (name: String, gpa : double, nbPpapers : int): constructor.
 - *getNbPapers*(): getter for attribute nbPapers.

UnderGraduate class

○ METHODS:

- ***UnderGraduate (name: String, gpa : double)***: constructor.

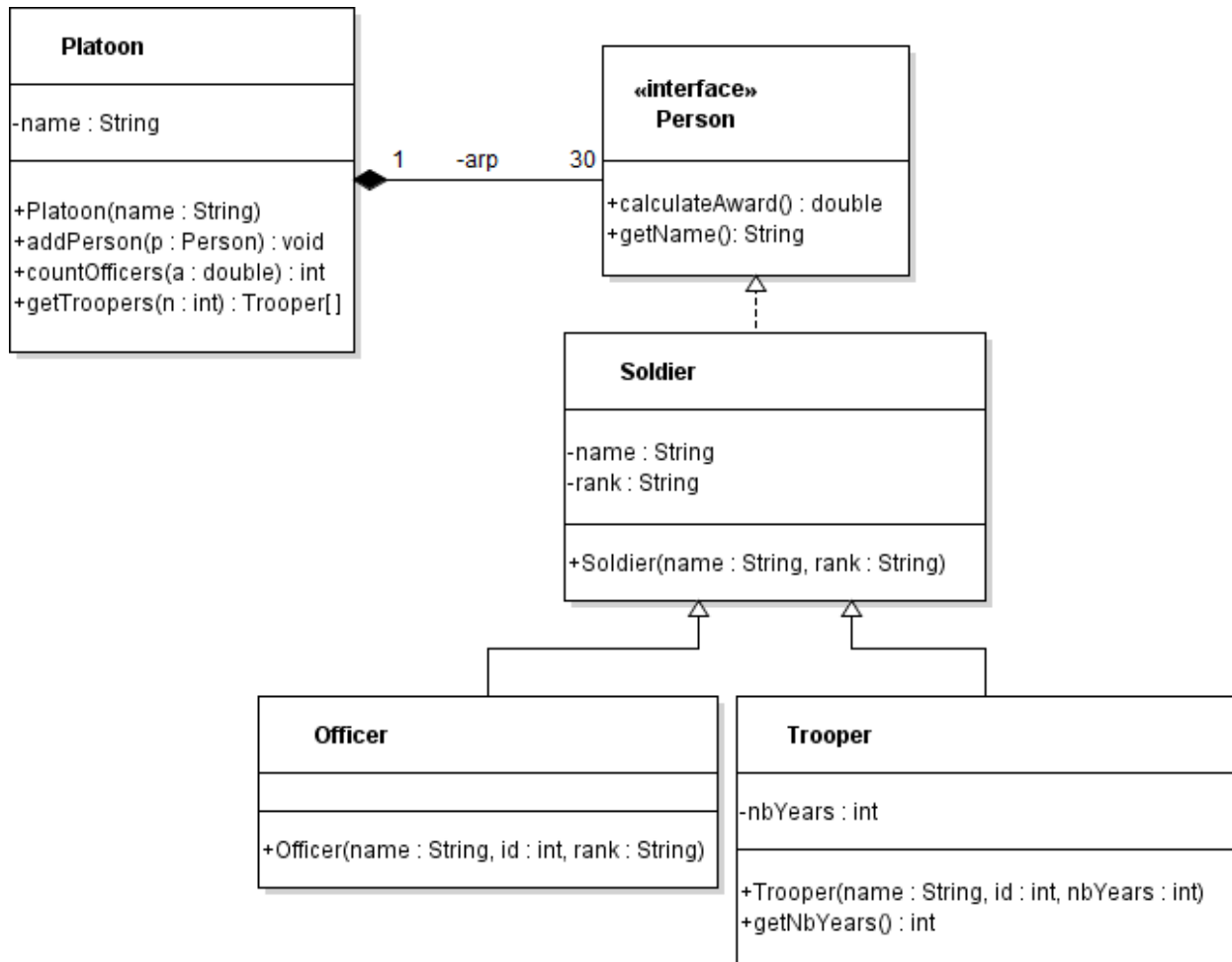
Institute class

○ METHODS:

- ***Institute(name: String)***: constructor.
- ***addPerson(p: Person)***: add a person to the institute.
- ***countUnderGraduate (s : double)***: count the number of **UnderGraduate** in the institute with **score** grater or equal to s.
- ***getGraduate (n : int)***: this method will return an array containing all the Graduate with number of papers greater than n.

QUESTION: Translate into Java code the class ***Graduate and Institute.***

Exercise



Interface Person:

○ METHODS:

- **calculateAward ()**: calculated as:
 - **for Officer** : Award = 50,000
 - **for Trooper**: Award = nbYears * 2000
- **getName ()**: returns the name of the Soldier.

Soldier class

○ METHODS:

- **Soldier (name: String, rank : String)**: constructor.

Officer class

○ METHODS:

- **Officer (name: String, rank : String)**: constructor.

Trooper class

○ METHODS:

- ***Trooper (name: String, rank : String, nbYears : int)***: constructor.
- ***getNbYears()***: getter for attribute nbYears.

Platoon class

○ METHODS:

- ***Platoon (name: String)***: constructor.
- ***addPerson(p: Person)***: add a person to the Platoon.
- ***countOfficers(a : double)***: count the number of **Officers** in the Platoon with Award greater than a.
- ***getTroopers(n : int)***: this method will return an array containing all the Troopers with nbYears greater than or equal to n.

QUESTION: Translate into Java code the class ***Trooper and Platoon.***