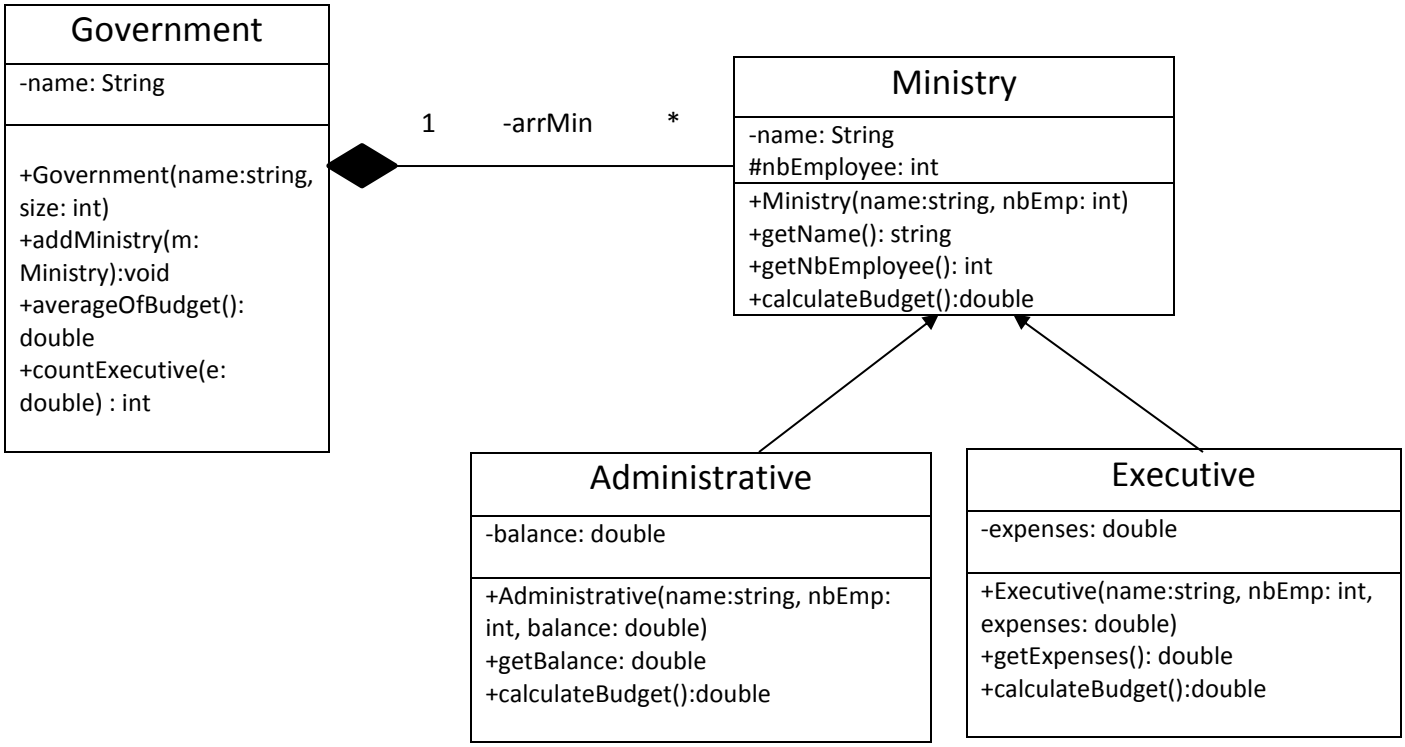
Lab (Polymorphism / Method overriding)



Ministry class:

- Attributes:
  - ***name***: the name of the ministry.
  - ***nbEmployee:*** number of employees in the ministry
- Methods:
  - ***Ministry(name: String, nbEmp: int)***: constructor
  - ***getName ()*:** this method returns the name of the ministry.
  - ***getNbEmployee ()*:** this method returns the number of employees of the ministry.
  - ***calculateBudget ()***: this method returns zero.

Executive class

- Attributes:
  - ***expenses***: the expenses of the ministry.
- Methods:
  - ***Executive(name: String, nbEmp : int, expenses : double)***: constructor.
  - ***getExpenses()***: this method returns the amount of expenses.
  - ***calculateBudget ()***: this method returns:

    *Budget = expenses + nbEmployee * 1.5*

Administrative class:

- Attributes:
  - ***balance***: the basic salary allocated for the role.
- Methods:
  - ***Administrative (name: String , nbEmp : int, balance : double):*** constructor.
  - ***getBalance ():*** This method returns the balance of the administrative ministry.
  - ***calculateBudget ()***: this method returns:

    *Budget = nbEmployee * 10000 - balance*.

Government class

- Attributes:
  - ***name***: the name of the Government.
- Methods:
  - ***Goverment (name: String, size: int)***: constructor.
  - ***addMinistry(m: Ministry*)*:*** this method adds the *Ministry* ***m*** to the Government.
  - ***averageOfBudget ():*** this method calculates and returns the average budget of all ministries of the government.
  - ***countExecutive(e: double):*** this method returns the number of **Executive** ministries with **expenses** greater than **e**.

Exercise: Write all the classes shown in the UML. In the test class create an object of Government and add Executive and Administrative class objects to it. Call the averageOfBudget() and countExecutive() methods on this government object.