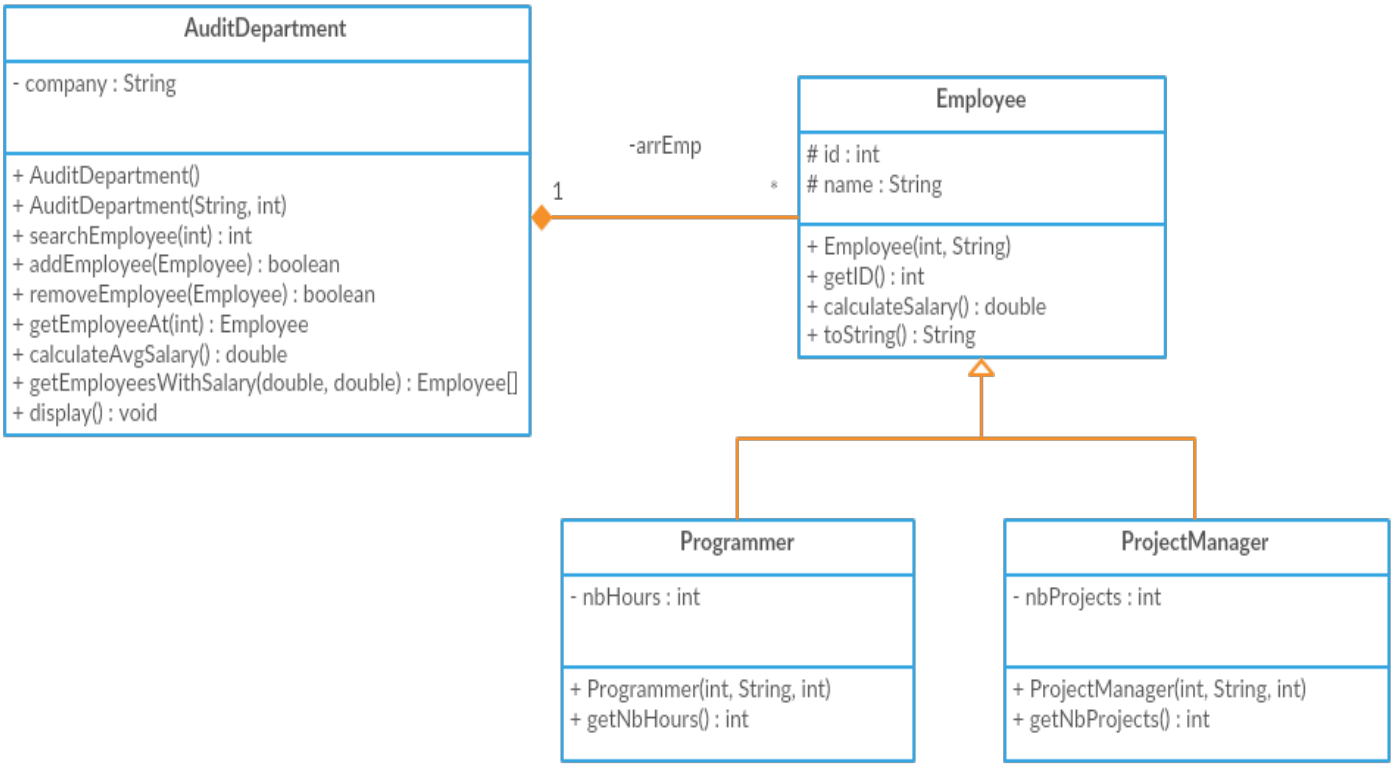
King Saud University
College of Computer and Information Sciences
Department of Computer Science
CSC113 – Computer Programming II – Exception Handling Lab – Fall 2019



Employee class:

- **Attributes:**
  - ***id:*** ID of the employee
  - ***name:*** name of the employee
- **Methods:**
  - ***Employee(int id, String name):*** constructor
  - ***getID():*** returns the id of the employee
  - ***calculateSalary():*** calculates monthly salary as follows:
    - ***for Programmers:*** 160SR an hour
    - ***for ProjectManagers:*** 30,000SR + 1,000SR for each project
  - ***toString():*** returns a string representation of Employee

Programmer class:

- **Attributes:**
  - ***nbHours:*** the number of daily work hours for the programmer
- **Methods:**
  - ***Programmer(int id, String name, int nbHours):*** constructor
  - ***getNbHours():*** returns the number of hours of the programmer

King Saud University
College of Computer and Information Sciences
Department of Computer Science
CSC113 – Computer Programming II – Exception Handling Lab – Fall 2019

ProjectManager class:

- **Attributes:**
  - ***nbProjects*:** the number of projects the manager is working on
- **Methods:**
  - ***ProjectManager(int id, String name, int NbProjects):*** constructor
  - ***getNbProjects():*** returns the number of projects the manager is working on

AuditDepartment class:

- **Attributes:**
  - ***company*:** the name of the company the department belongs to
- **Methods:**
  - ***AuditDepartment()*:** default constructor
  - ***AuditDepartment(String company, int size):*** constructor
    - Throws a ***NegativeArraySizeException*** when created with a negative size
  - ***searchEmployee(int id):*** searches for the employee using his id and returns his index in the array if found
  - ***addEmployee(Employee e):*** adds Employee ***e*** in the department if he isn't already added
  - ***removeEmployee(Employee e):*** removes employee ***e*** from the department if he exists
  - ***getEmployeeAt(int i):*** returns the Employee at index ***i***
    - Throws an ***ArrayIndexOutOfBoundsException*** when ***i*** is out of bounds
    - Throws a ***NullPointerException*** when employee at ***i*** is null
  - ***calculateAvgSalary():*** calculates the average salary for all employees
    - Throws an ***ArithmeticException*** when dividing by zero
  - ***getEmployeesWithSalary(double lowerBound, double upperBound):*** returns an array of employees having salaries between lower and upper bounds inclusive
    - Throws an ***IllegalArgumentException*** when the lower bound is larger than upper bound
  - ***dispay*():** displays the information of all employees in the department

Exercise 1: Translate into Java code classes *Employee, Programmer, ProjectManager,* and *AuditDepartment.*

Exercise 2: Write a main method that tests the functionalities of the previous classes. The main method should display an interactive menu for the user in the following way:

- Please enter the name of the company and the number of employees in the Audit Department:
  - To add an employee, enter 1
    - Enter 1 for a Programmer, or 2 for a ProjectManager
      - …Proceed to take the required info
  - To print the info of an employee, enter 2
    - Enter the index of the employee
  - To print the average salary, enter 3
  - To list the employees with certain salary range, enter 4
    - Enter the *lower* and *upper* bounds
  - To display the information of all employees, enter 5
  - To exit, enter -1
  - Enter your option:

In each method-call that throws an exception, handle that exception using ***try-catch*** statements and print a useful message when an exception is caught e.g. since we know that the ***AuditDepartment*** constructor will throw a ***NegativeArraySizeException*** when passed a negative size, we will surround that method call with a try statement, and catch the specified exception. In every data entry from the user, ***InputMismatchException*** should also be handled e.g. entering a string of characters when an integer input is expected.