



Employee class:

- **Attributes:**
 - *id*: ID of the employee
 - *name*: name of the employee
- **Methods:**
 - *Employee(int id, String name)*: constructor
 - *calculateSalary()*: calculates monthly salary as following:
 - *for Programmers*: hours * 55 * 30
 - *for ProjectManagers*: nbProjects * 500 * 30
 - *display()*: displays the id, name, and salary of the employee

Programmer class:

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

ProjectManager class:

- **Attributes:**
 - *nbProjects*: the number of projects the manager is working on
- **Methods:**

AuditDepartment class:

- **Attributes:**
 - ***company***: the name of the company the department belongs to
- **Methods:**
 - ***AuditDepartment(String company, int size)***: constructor
 - Throws a ***NegativeArraySizeException*** when created with a negative size
 - ***addEmployee(Employee e)***: adds Employee *e* in the department
 - ***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 lower, double upper)***: returns an array of employees having salaries between ***lower*** and ***upper*** bounds inclusive
 - Throws an ***IllegalArgumentException*** when ***from*** is larger than ***to***
 - ***display()***: displays the information of all employees in the department

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

Exercise 2: Create a main class that tests the functionalities of the previous classes. The main class 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 6
 - Enter your option:

In each method call that could throw 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. There are five exceptions to be caught overall.