**JAVA LAB**

**EXERCISE #1**   
Write a program by creating an 'Employee' class having the following methods and print the final salary.   
1 - 'getInfo()' which takes the hourly salary, number of hours of work per day of employee as parameter   
2 - 'computeBaseSalary()' which computes the base salary for 20 days of work   
2 - 'addSal()' which adds $10 to the base salary of the employee if it is less than $500.   
3 - 'addWork()' which adds $5 daily to salary of employee if the number of hours of work per day is more than 6 hours.

Employees:

|  |  |  |
| --- | --- | --- |
| Name | Hours/day | Salary/day |
| John Smith | 5 | 10 |
| Sandra Wang | 4 | 5 |
| Gio Rossi | 7 | 12 |

**EXERCISE #2**   
Write a program to print the perimeter and the area of a square, a rectangle and a circle.

Shapes:

|  |  |
| --- | --- |
| Type | Dimensions |
| square | l = 4 |
| rectangle | b = 2, h = 3 |
| circle | r = 1.5 |

**EXERCISE #3**

Write a program to manage the ranking points of different teams. Create the interface **Team** with the following method:

updatePoints()

getTeamName() getPoints()

Create two classes that implement the Team interface:

**FootballTeam** with attributes: TeamName: String  
Points: Integer  
Sponsor: String

**BasketballTeam**

TeamName: String Points: Integer

Define the following methods in the interface and where is necessary:

constructors of the classes;  
updatePoints: update the points of the team with the last match result, using these rules:

FOOTBALL TEAM:  
Win: add 3 points; Drawn: add 1 point; Lose: do nothing.

BASKETBALL TEAM: Win: add 2 points; Lose: do nothing.

getTeamName: return the name of the team; getPoints: return the points of the team.

The main method should:

1. Load teams parameters from teams.csv file; for each line define the instance for a team;
2. Print standing of Football teams (ordering by points);
3. Print standing of Basketball teams (ordering by points);
4. Load results of the last matches from results.csv file and update teams points based on loaded results;
5. Print new standing of Football teams (ordering by points);
6. Print new standing of Basketball teams (ordering by points).

**File teams.csv format (without header):**

Sport, Team Name, Points, Sponsor

**File results.csv format (without header):**

Team Name, Result

HINTS:  
To read CSV file use java.io.BufferedReader, java.io.InputStreamReader and the String split method  
To collect the instance of Football and Basketball teams use collection of type Team

**EXERCISE #4**

Using the file Books.csv write a program for our amazing library!

Build 2 entities: Author and Book

The Author has a name.

The Book has a title, a year of publication and an average rating.

An author owns a list of books.

In the csv there are many columns: ignore the useless ones.

Pay attention: several books are written by different authors, for example: "The Adventures of Huckleberry Finn" was written by Mark Twain, John Seelye, Guy Cardwell.

In this case, each author will have the same book in their list, so:

Mark Twain -> "The Adventures of Huckleberry Finn"

John Seelye -> "The Adventures of Huckleberry Finn"

Guy Cardwell -> "The Adventures of Huckleberry Finn"

Using the concepts you learned during the course (Composition, polymorphism, ...), create a Library that allows you to:

1. List all books written by an author using following criteria:

* alphabetical asc/desc,
* publication date asc/desc,
* rating asc/desc;

2. Given a book, know who is the author / are the authors;

3. How many books has an author written;

4. Given a Date, what are the books written in that year;

5. ... who is the most prolific author?

6. (Optional) If you think others functionality are usefull, try to add to the exercise.

I recommend simple, small and clean methods. Use javadoc, write junits and use a build automation tool (maven or gradle for example)

**EXERCISE #5**

**Car rental management**

Description.

You want to create a program in Java for the management of the vehicles of a **car rental**.

The data that must be registered for each **vehicle** are: an identification code, the brand, the model, the number of seats, the license plate.

There are about fifty vehicles to be managed.

When a vehicle is rented, it is necessary to register the rental data including the data of the **customer** who made the **rental** and the chosen vehicle.

Persistent data storage of each rental must be achieved by storing the data in a text file in CSV format.

Finally, the application must allow you to export all car rental vehicles and import new ones, through a text file in CSV format.

The features that must be available in the application are:

1. Addition of a new rental of a vehicle by a customer, with which the rental start and end dates, customer data and a reference to the rented vehicle are recorded; (done)

2. Return of the vehicle to the car rental; (done)

3. Display of the list of all vehicles belonging to the car rental fleet. (Done)

4. View the list of all vehicles available for rental in a specified period of time. ??

5. Adding a new vehicle to the car rental fleet. (done)

6. Cancellation of a vehicle from the car rental fleet. (Done)

7. Importing vehicles from a CSV file, with which it is possible to add new vehicles to the car rental fleet. (Done)

8. Export of all car rental vehicles in a CSV file. (Done)