

TP 05

Part One: (ArrayList) ★★★★★☆

Task 01 ➔ Write a Java program to create a new ArrayList (of type String), then ask a user to enter some element, and print out the collection after each entry.

Task 02 ➔ Add to the past program a Search method that look in the in ArrayList and return the element that match

Task 03 ➔ Add to the past program a Search method Search in ArrayList and return a sub list (in case of same element is duplicated)



In rest of the following tasks, our objective is to create a java program that digitizes the student list and their TP notes:

Task 04 ➔ Create a student class that contain the following attributes: student id (int and auto generated in the constructor), name, student TP note. This class contain only one method that print student information.

Task 05 ➔ Create a separate Main class that contain a main method to run our program and a static ArrayList of type Student.

Task 06 ➔ In the Main class, add the following functionalities and methods:

- Allow a user to enter a student information, create a new student object and then add to the ArrayList.
- Allow a user to search a Student by its name.
- Allow a user to search a set Students that have a similar names.
- Allow a user to enter a student TP note.
- Allow a user get average of all classe.

Note: Solution should have these two files:  Main.java  Student.java

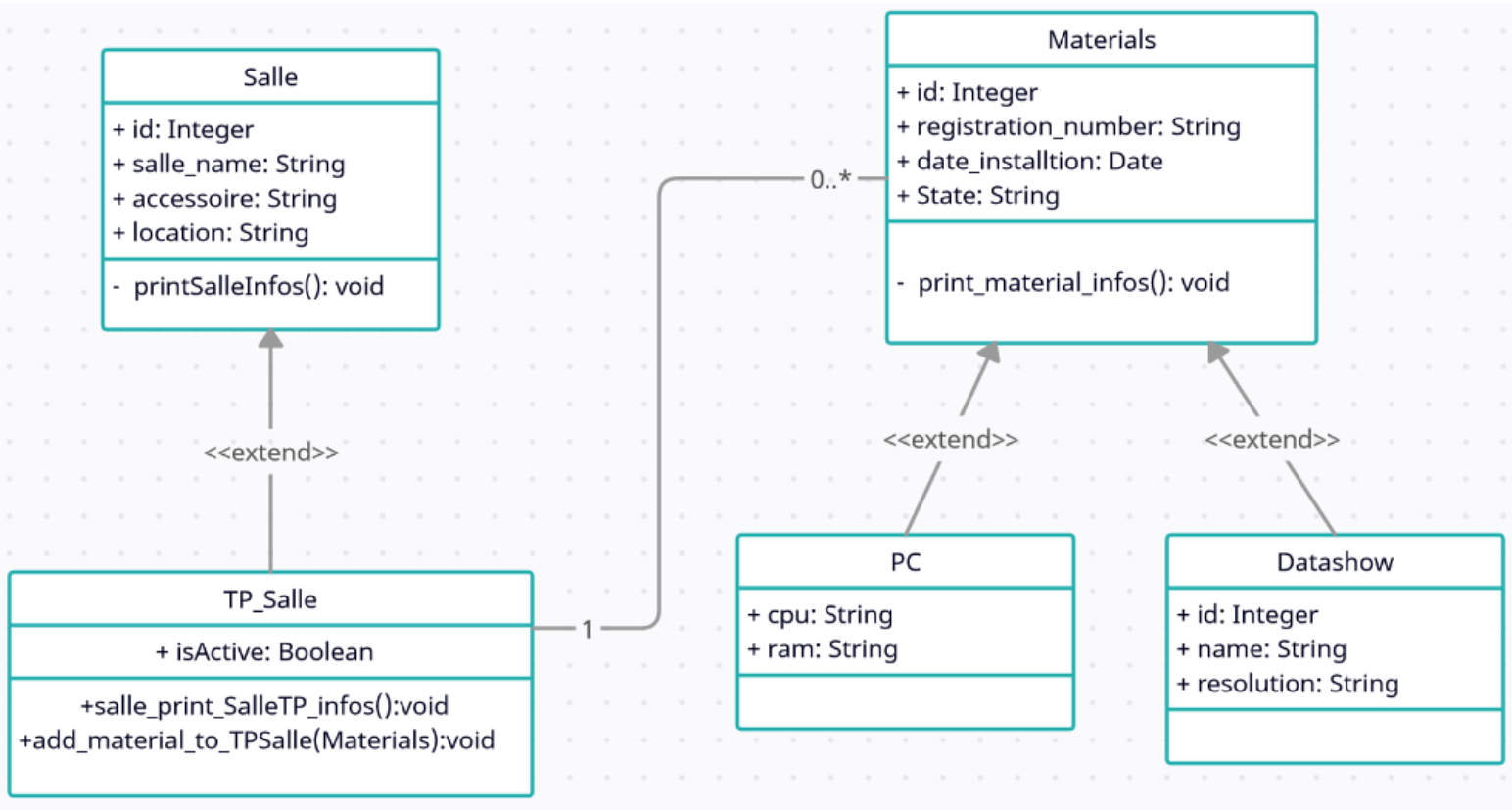
A Part of the Main class sample:

```
1 import java.util.ArrayList;
2 import java.util.Scanner;
3
4
5 public class Main {
6     static ArrayList<Student> student_list = new ArrayList();
7
8     public static void add_student() {...20 lines }
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29     public static void show_all_student() {...12 lines }
30
31
32
33
34
35
36
37
38
39
40
41
42     public static void search_one() {...14 lines }
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57     public static void search_many() {...17 lines }
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75     public static void compute_average_notes() {...10 lines }
76
77
78
79
80
81
82
83
84
85
86     public static void main(String [] args){
```

run:	Program output
Enter (add) to add a new student	
Enter (show) to show all students	
Enter (search) to search student by its name	
Enter (searchAll) to search students	
Enter (avg) To compute and display the class average	
add	
Enter Student name:	amine
Enter Student TP note:	15
The student has been added to the list	
do you want to add a new student (y/n)?	n
	Student Id:249046391
	Student Name:amine
	Student TP note:15.0
Enter (add) to add a new student	
Enter (show) to show all students	
Enter (search) to search student by its name	
Enter (searchAll) to search students	
Enter (avg) To compute and display the class average	

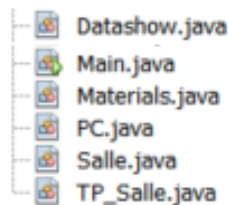
Part Two: Inheritance ★★★★★

Task 01 ➔ Using NetBeans, give the java code of the following diagram:



Task 02 <<Optional>> ➔ propose a test for example: create TP_Salle with a set of materials.

The result should be the following classes (files):



Note 1: respect organization of files.

Note 2: during the TP evaluation, any other ide except Netbeans will not be considered.