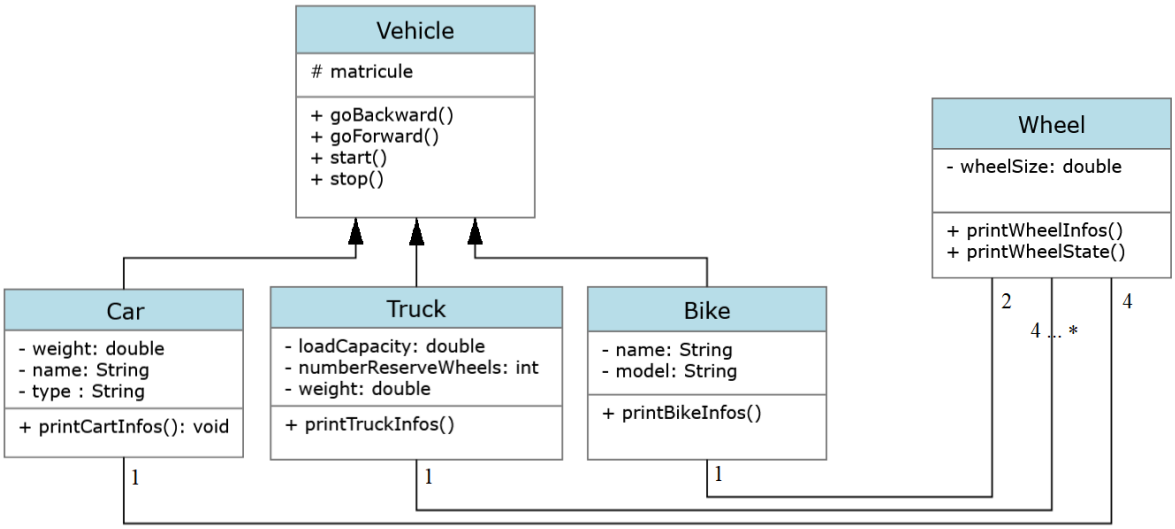
TP 06

Part One: (inheritance, interface, encapsulation) ★★★★★

Task 01➔ Answer the following questions (not in your head)

1. *What are the difference between an abstract class and an interface class*
2. *Can a Java class extend multiple interfaces? Can it extend multiple abstract classes? Explain why or why not.*
3. *How do abstract classes and interfaces differ in their implementation details and restrictions on method declarations?*

The following tasks need to be completed based on the given UML class diagram:



Task 02➔ Give java implementation all classes and their methods, including the constructors.

Task 03➔ Use encapsulation to ensure attributes can only be added and removed through methods, not directly.

Task 04➔ Create a Main class that contain a main method to execute the program. Your program should able to create objects instance of **Car**, **Truck** and **Bike**. Then, prints their information.

Task 05➔ Change the Vehicle class to a java interface class and establish the required changes to your code, including the implementation of the Vehicle class methods by the childclasses.

Note: do not remove Vehicle class; just rename it to Vehicle2, for the Tp consultation purposes.

Dr. Oussama SERHANE (o.serhane@esi-sba.dz)

Part Two: (inheritance, interface, encapsulation, abstraction) ★★★★★

We aimed to develop software for the Esi-Sba institution that can effectively manage the various properties in their facilities. This software will be capable of handling electronic materials like computers, printers, etc., as well as accessories such as tables, chairs, and blackboards. To achieve this goal, we have created a class diagram for the software (see next page). For the sake of simplicity, we have focused only on the relevant services and properties and have not considered all the buildings and services offered by the institution.

The objective of this TP part is two folds:

1) Know how to code interface and abstract class and know their utilities and the difference between them.

2) Made an initialization for the mini project of the upcoming days.

Note: you can reuse the code of Tp5 part Two.

Task 01 ➔ Edit the class code of TP05 part two to adapt with this diagram.

Task 02 ➔ Give a java code of all classes of the diagram including their constructor, association, methods.

Task 03 ➔ We did provide a Main.java class with these TP assets copy it to your code then test it.

Task 04 ➔ Note that esi-sba contains two departments, and in the main, we have created only the second cycle. Add your department.

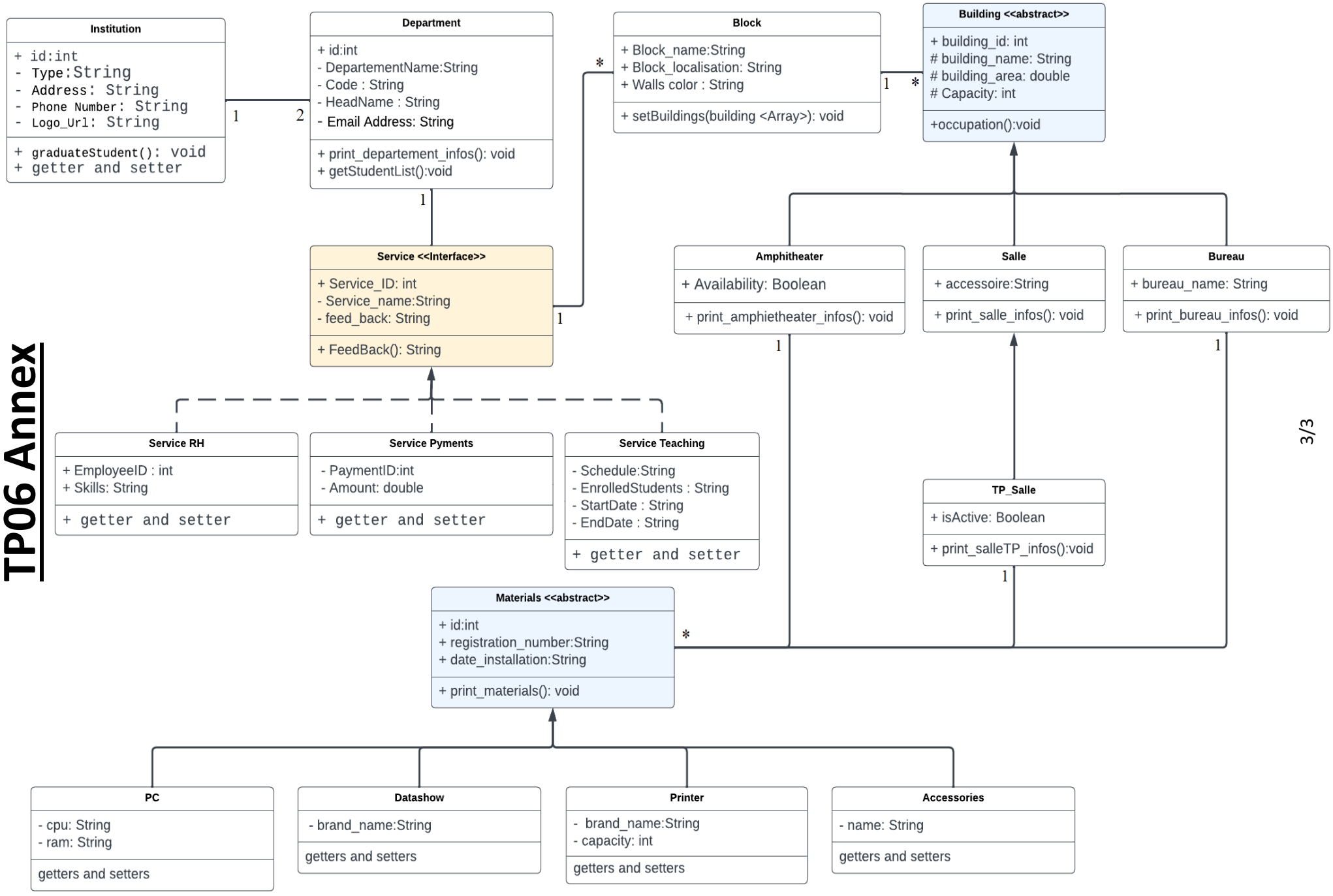
Task 05 ➔ Add a function that print all the information of the ESI-sba institution.

Note 1: respect organization of files.

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

Dr. Oussama SERHANE (o.serhane@esi-sba.dz)

TP06 Annex


Institution
+ id:int
- Type:String
- Address: String
- Phone Number: String
- Logo_Url: String
+ graduateStudent(): void
+ getter and setter
1
2
Department
+ id:int
- DepartementName:String
- Code : String
- HeadName : String
- Email Address: String
+ print_departement_infos(): void
+ getStudentList():void
1
*
Block
+ Block_name:String
+ Block_localisation: String
+ Walls color : String
+ setBuildings(building <Array>): void
1
*
Building <<abstract>>
+ building_id: int
# building_name: String
# building_area: double
# Capacity: int
+occupation():void
Service <<Interface>>
+ Service_ID: int
- Service_name:String
- feed_back: String
+ FeedBack(): String
1
Amphitheater
+ Availability: Boolean
+ print_amphietheater_infos(): void
1
Salle
+ accessoire:String
+ print_salle_infos(): void
Bureau
+ bureau_name: String
+ print_bureau_infos(): void
1
Service RH
+ EmployeeID : int
+ Skills: String
+ getter and setter
Service Pyments
- PaymentID:int
- Amount: double
+ getter and setter
Service Teaching
- Schedule:String
- EnrolledStudents : String
- StartDate : String
- EndDate : String
+ getter and setter
TP_Salle
+ isActive: Boolean
+ print_salleTP_infos():void
1
Materials <<abstract>>
+ id:int
+ registration_number:String
+ date_installation:String
+ print_materials(): void
*
PC
- cpu: String
- ram: String
getters and setters
Datashow
- brand_name:String
getters and setters
Printer
- brand_name:String
- capacity: int
getters and setters
Accessories
- name: String
getters and setters