Lab 06

Lab-06 Objectives:

The objective of this Lab is to understand the relationship between classes, abstract classes, and interfaces in Java, and how they can be used in combination to write more flexible and extensible code.

Part 01-Lab Setup (average of estimated time to complete this part 30 min):

1. Create a new Java project name it "**ClassAbstractInterfaceTP06**".
2. Create three packages named "**classes**", "**abstractclasses**", and "**interfaces**".
3. In the "**classes**" package, create a class named "Person" that has attributes and methods as in (See Annex Person class).
4. Implement the code inside "**printDetails**" method to print the Name and the age of the employee
5. In the "**abstractclasses**" package, create an abstract class named "**Employee**" that extends the "**Person**" that has attributes and methods as in (See Annex Employee class).
6. In the "**interfaces**" package, create an interface named "**Manager**" with the following method signature:

```java
public interface Manager {
    public void approveLeave(String employeeName);
}
```

7. In the "**classes**" package, create a class named "**SalesPerson**" that extends the "**Employee**" abstract class and implements the "**Manager**" interface, SalesPerson class has attributes and methods as in (See Annex SalesPerson class).
8. Add the body of "**callculateBonus**" and "**approveLeave**" methods as following:

```java
    // Override method to calculate bonus
public double calculateBonus() {
    return salesAmount * 0.1;
}
    // Method to approve leave
public void approveLeave(String employeeName) {
    System.out.println("Leave approved for employee " + employeeName);
}
```

Part 02-Lab Execution (average of estimated time to complete this part 20 min):

1. In the "**main**" method, create an instance of the "**SalesPerson**" class and set its properties:

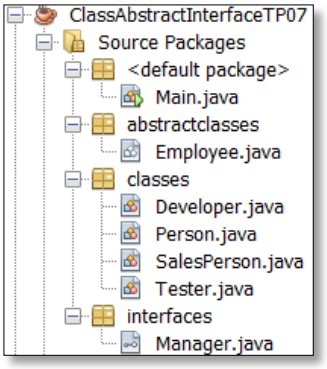
   `("Amine", 30, 101, 5000.0, 10000.0) and then print its details information`

2. Call the "**calculateBonus**" method on the "**salesPerson**" object to calculate its bonus and print it.
3. Call the "**approveLeave**" method on the "**salesPerson**" object to approve leave for an employee.
4. Create a new class named "**Developer**" in the "**classes**" package that also extends the "**Employee**" abstract class interface and has attributes and methods as in (See Annex Developer class).
5. Add the body of "**callculateBonus**" method as following:

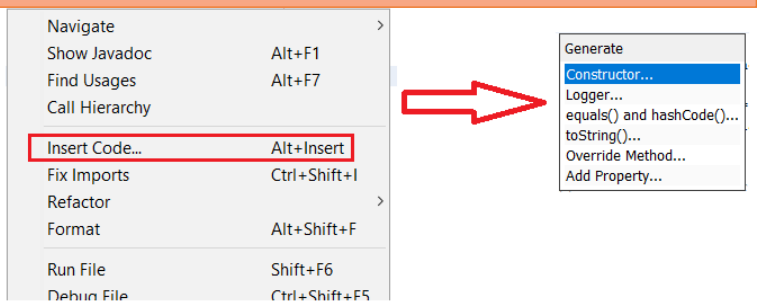
```
// Override method to calculate bonus
public double calculateBonus() {
    return salary * (experience * 0.01);     }
```

6. Create a new class named "**Tester**" in the "**classes**" package that implements the "**Manager**" interface and has attributes and methods as in (See Annex Tester class).
7. Test your code by creating instances of the "**Developer**" and "Tester" classes and calling their respective methods.
8. Finally, Give the final UML diagram of all classes and the relationship between them.

The final result Classes:



Important: use the build-in Netbeans option to insert constructor and getter and setters. Do not write all codes. As showing the following:



Lab 06 Conclusion:

In this Lab, you learned how to use classes, abstract classes, and interfaces in Java to write more flexible and extensible code. You also learned how to create and use abstract classes, interfaces, and how to implement them in your code. By using these concepts, you can write code that is easier to maintain, extend, and reuse.

Part 03 Create your own exercise: Use an LLM (like ChatGPT) to generate a custom Java exercise involving abstract classes and key OOP concepts.

What You'll Do:

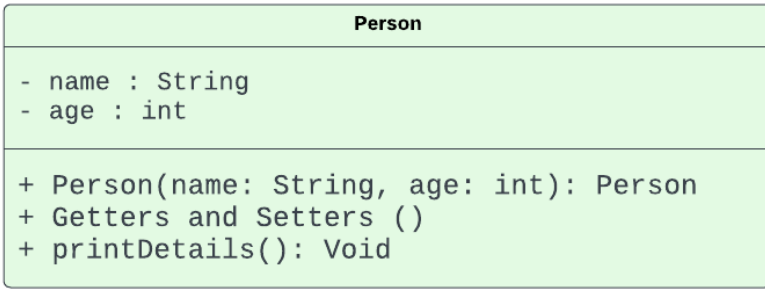
You will use a large language model (LLM) like ChatGPT to generate a Java programming exercise. The exercise must require the use of:

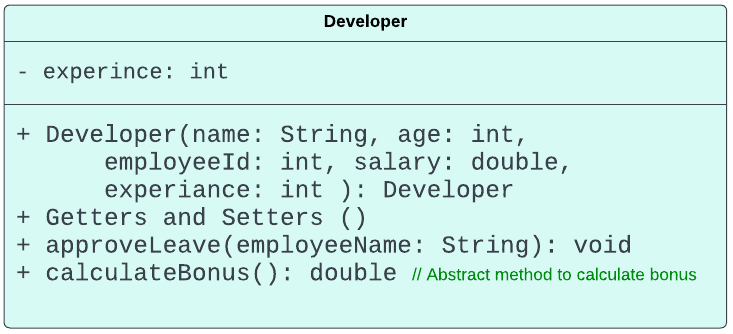
- ✓ Abstract class(es)
- ✓ At least **two levels of abstraction** (e.g., abstract class → concrete class → subclass)
- ✓ Encapsulation (use of `private` fields and getters/setters)
- ✓ Polymorphism (objects used via abstract class reference)
- ✓ Create Uml diagram

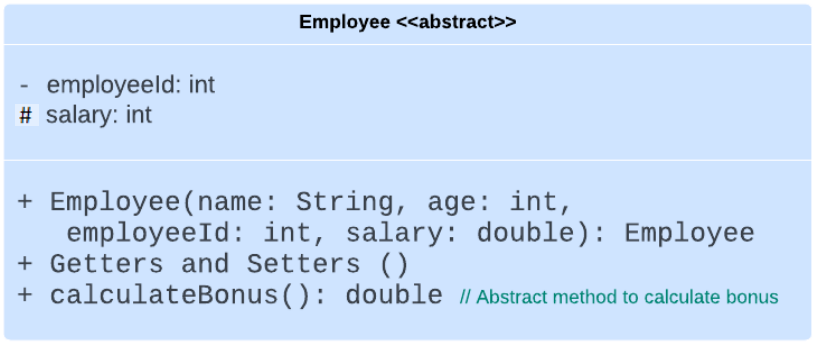
Then, **you will implement the solution** and presented during Lab session.

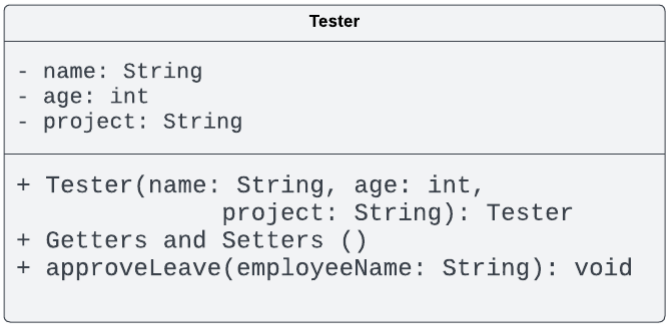
Dr. Oussama SERHANE (o.serhane@esi-sba.dz)

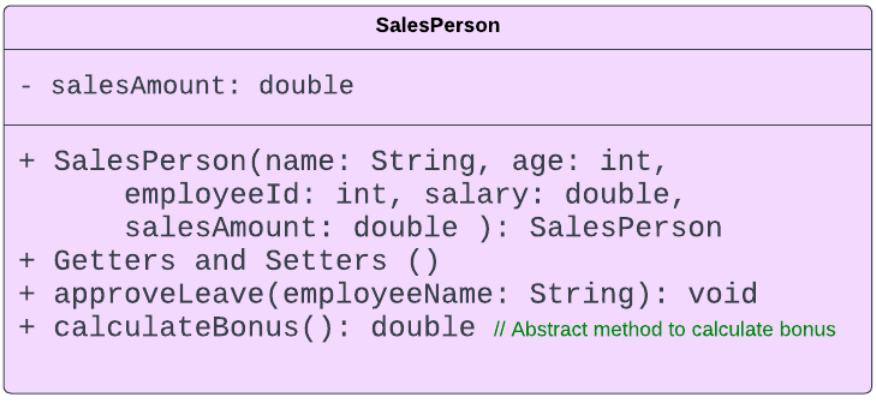
Lab 06-Annex











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