

Group Exercise #2

After Java OOPs:

Must complete tasks as a group

PART 1: CODING

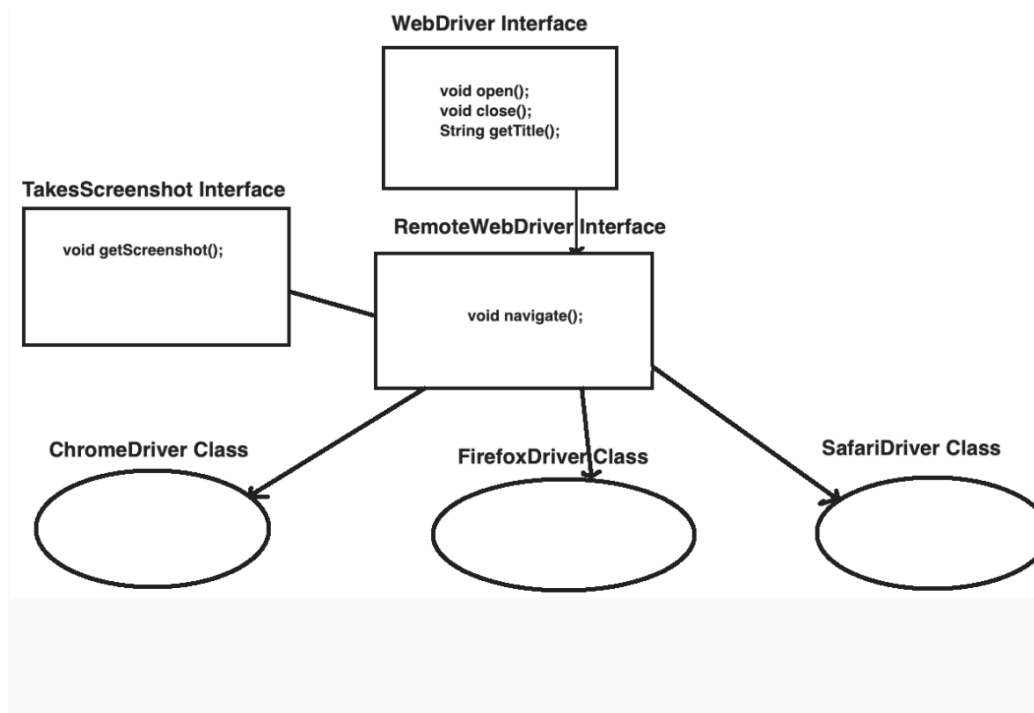
1. Create an Interface 'Shape' with undefined methods as calculateArea and calculatePerimeter. Create 2 classes Circle & Square that implements functionality defined in the Shape Interface. Test your code.
2. We have to calculate the average of marks obtained in three subjects by student A and by student B. Create class 'Marks' with an abstract method 'getPercentage' that will be returning the average percentage of marks. Provide implementation of abstract method in classes 'A' and 'B'. The constructor of student A takes the marks in three subjects as its parameters and the marks in four subjects as its parameters for student B. Test your code
3. Create a Class Car that would have the following fields: carPrice and color and method calculateSalePrice() which should be returning

a price of the car.

Create 2 sub classes: Sedan and Truck. The Truck class has a field as weight and has its own implementation of calculateSalePrice() method in which returned price is calculated as following: if weight>2000 then returned price car should include 10% discount, otherwise 20% discount.

The Sedan class has field as length and also does it is own implementation of calculateSalePrice(): if length of sedan is >20 feet then returned car price should include 5% discount, otherwise 10% discount

4. Provide Implementation for the diagram below. Then create a test class in which you need to create Objects of ChromeDriver, FirefoxDrive and SafariDriver classes and see which methods available to them.



2. PART 2

First have a discussion about each topic and when all team members are clear then Record a flipgrid video as a group:
<https://flip.com/8b308dba>

Questions:

1. What types of variables we have in Java. Define all of them and explain the differences

2. What is the constructor? Use of a constructor in the class? Difference between constructor and a method? Can we overload a constructor?
3. What is the usage of super and this? Super vs this? Super() vs super.? this() vs this.
4. What is Inheritance? Benefits? Types? Explain each type
5. What is Overloading? How can it be achieved? What can we overload and what we cannot?
6. What is Overriding? How can it be achieved? What can we override and what we cannot?
7. What is Polymorphism? Types of Polymorphism? Differences between Overloading & Overriding?
8. What is the final keyword, usage and purpose?

9. What is Abstraction? How can we achieve an Abstraction in Java? What is Interface? What is Abstract Class? Interface vs Abstract Class?