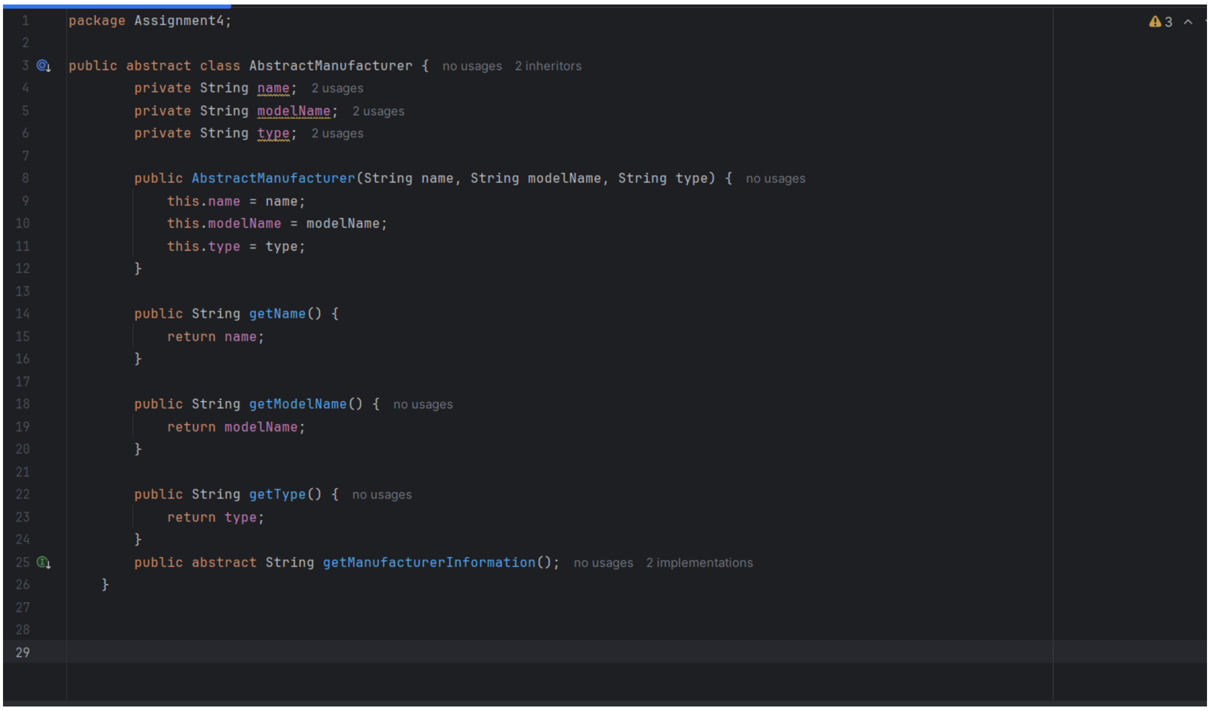
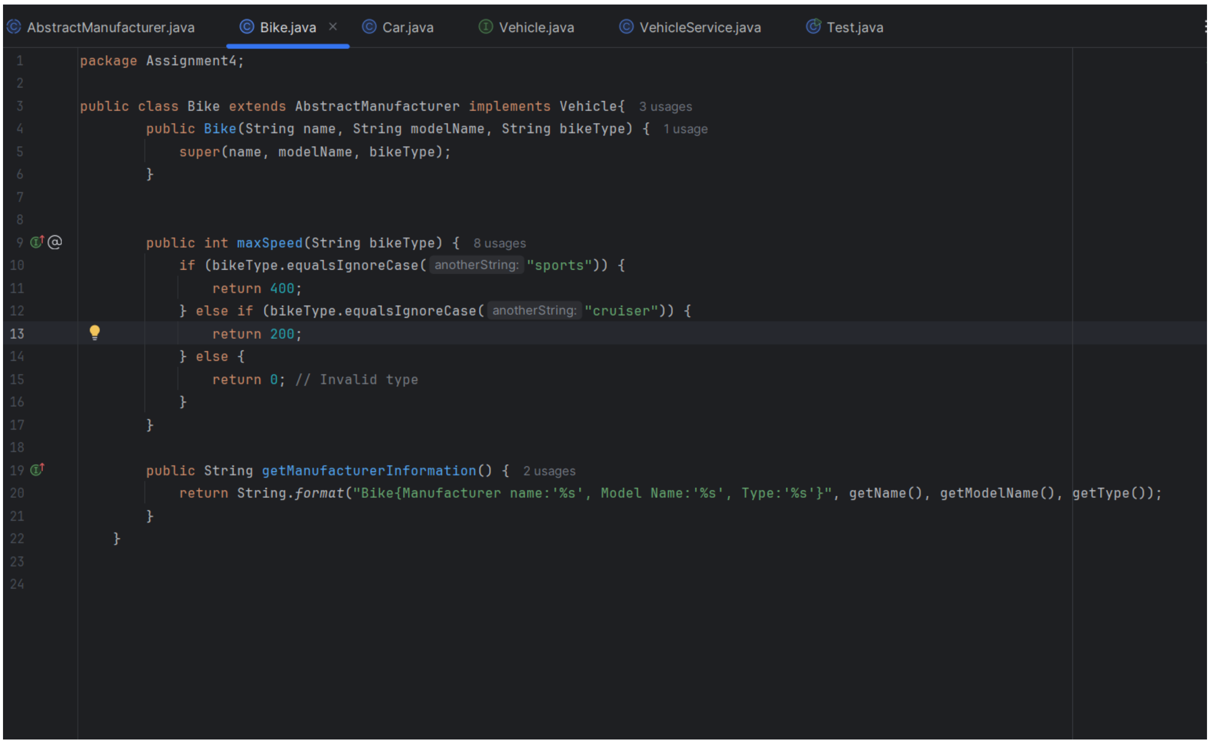
Assignment 4

IMPLEMENTING POLYMORPHISM USING ABSTRACT CLASS AND INTERFACE:

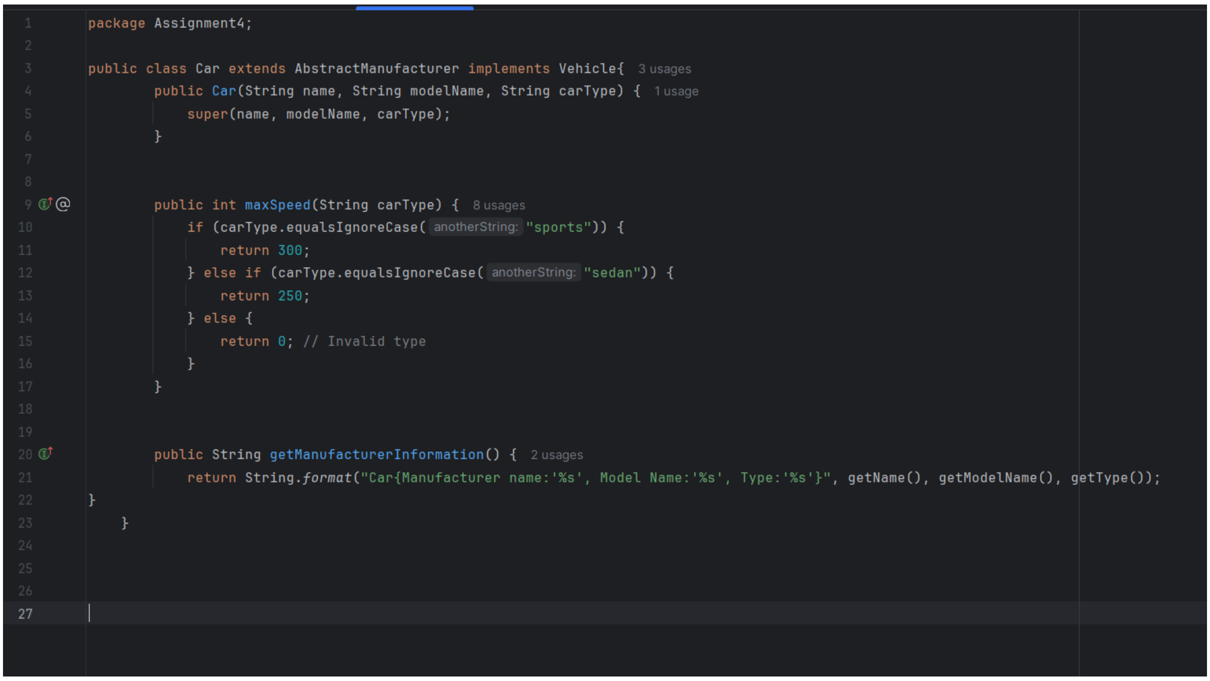
CODE:



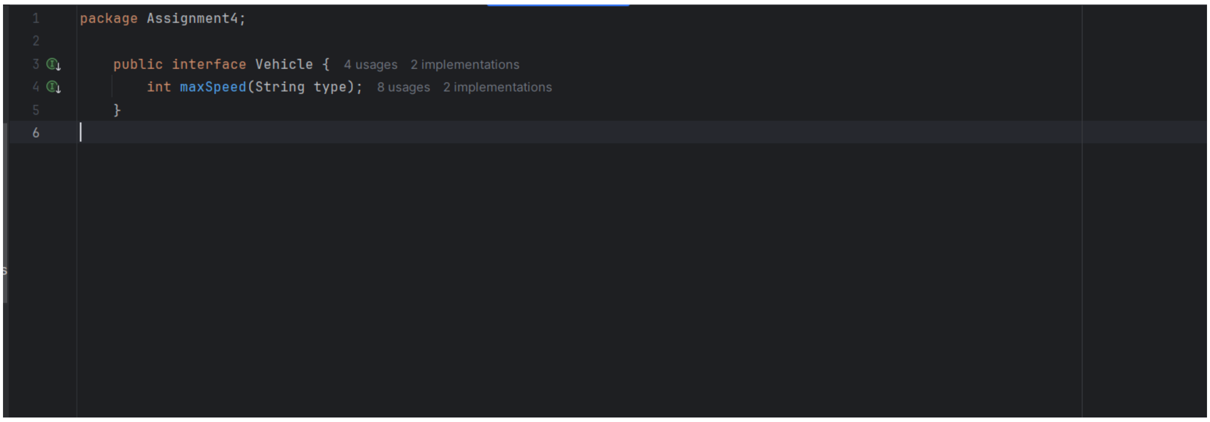
```java
package Assignment4;

public abstract class AbstractManufacturer {
        private String name;
        private String modelName;
        private String type;

        public AbstractManufacturer(String name, String modelName, String type) {
            this.name = name;
            this.modelName = modelName;
            this.type = type;
        }

        public String getName() {
            return name;
        }

        public String getModelName() {
            return modelName;
        }

        public String getType() {
            return type;
        }
        public abstract String getManufacturerInformation();
    }


```



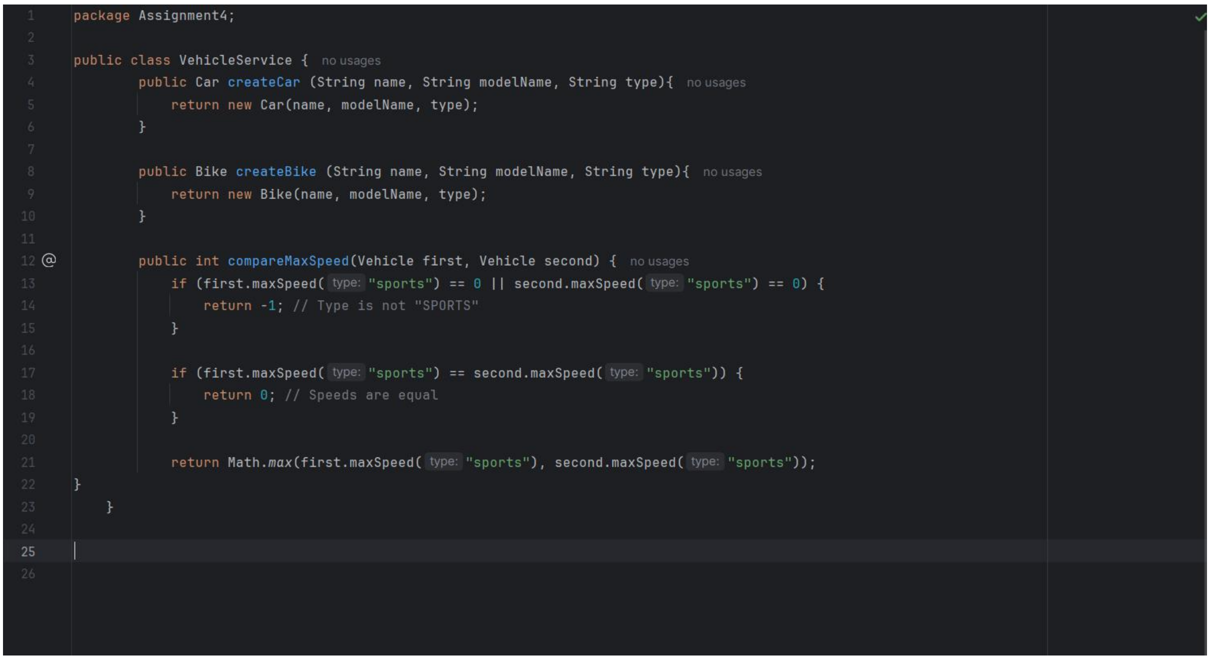
```java
package Assignment4;

public class Bike extends AbstractManufacturer implements Vehicle{
        public Bike(String name, String modelName, String bikeType) {
            super(name, modelName, bikeType);
        }


        public int maxSpeed(String bikeType) {
            if (bikeType.equalsIgnoreCase("sports")) {
                return 400;
            } else if (bikeType.equalsIgnoreCase("cruiser")) {
                return 200;
            } else {
                return 0; // Invalid type
            }
        }

        public String getManufacturerInformation() {
            return String.format("Bike{Manufacturer name:'%s', Model Name:'%s', Type:'%s'}", getName(), getModelName(), getType());
        }
    }
```



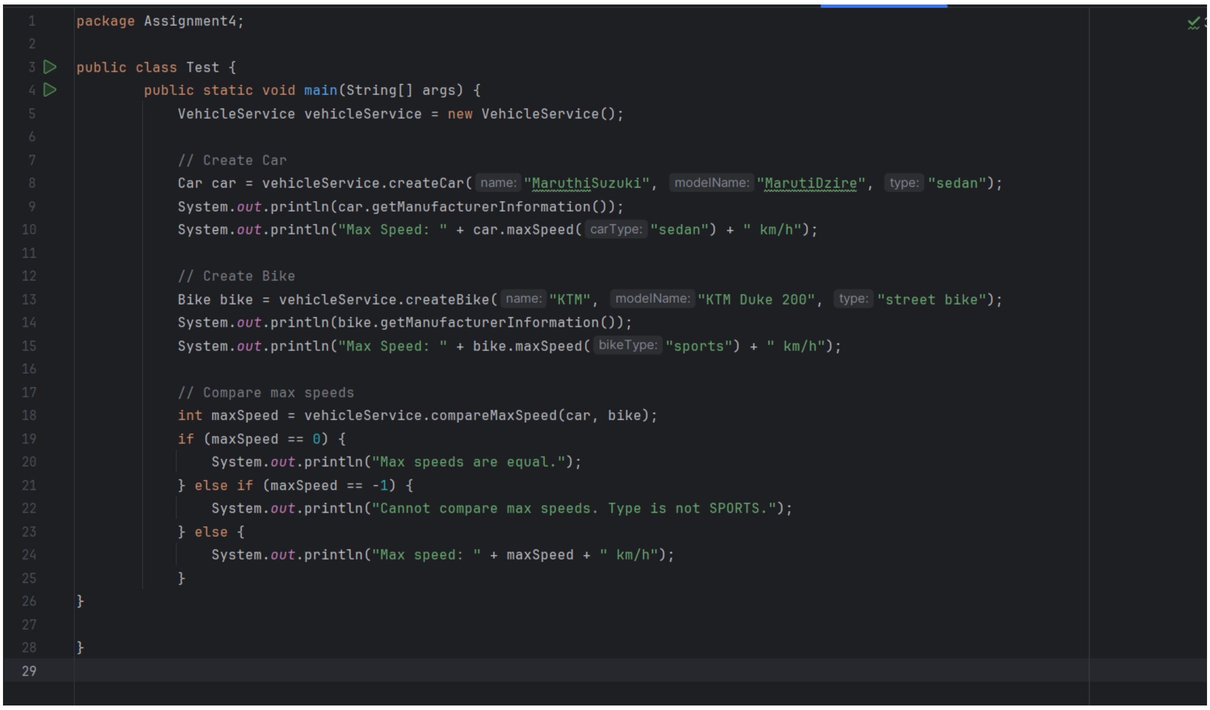
```
package Assignment4;

public class Car extends AbstractManufacturer implements Vehicle{  3 usages
        public Car(String name, String modelName, String carType) {  1 usage
            super(name, modelName, carType);
        }


        public int maxSpeed(String carType) {  8 usages
            if (carType.equalsIgnoreCase( anotherString: "sports")) {
                return 300;
            } else if (carType.equalsIgnoreCase( anotherString: "sedan")) {
                return 250;
            } else {
                return 0; // Invalid type
            }
        }


        public String getManufacturerInformation() {  2 usages
            return String.format("Car{Manufacturer name:'%s', Model Name:'%s', Type:'%s'}", getName(), getModelName(), getType());
}
    }




```



```
package Assignment4;

    public interface Vehicle {  4 usages  2 implementations
        int maxSpeed(String type);  8 usages  2 implementations
    }

```

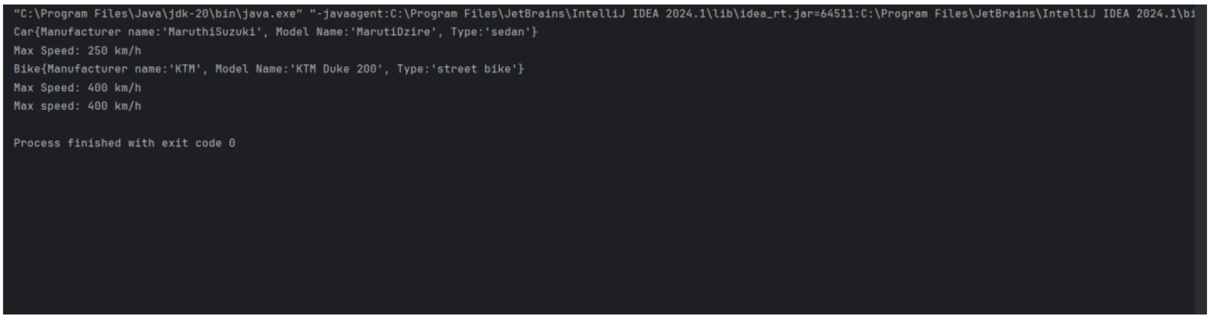


```
package Assignment4;

public class VehicleService {  no usages
        public Car createCar (String name, String modelName, String type){  no usages
            return new Car(name, modelName, type);
        }

        public Bike createBike (String name, String modelName, String type){  no usages
            return new Bike(name, modelName, type);
        }

        public int compareMaxSpeed(Vehicle first, Vehicle second) {  no usages
            if (first.maxSpeed( type: "sports") == 0 || second.maxSpeed( type: "sports") == 0) {
                return -1; // Type is not "SPORTS"
            }

            if (first.maxSpeed( type: "sports") == second.maxSpeed( type: "sports")) {
                return 0; // Speeds are equal
            }

            return Math.max(first.maxSpeed( type: "sports"), second.maxSpeed( type: "sports"));
}
    }


```



```java
package Assignment4;

public class Test {
        public static void main(String[] args) {
            VehicleService vehicleService = new VehicleService();

            // Create Car
            Car car = vehicleService.createCar( name: "MaruthiSuzuki",  modelName: "MarutiDzire",  type: "sedan");
            System.out.println(car.getManufacturerInformation());
            System.out.println("Max Speed: " + car.maxSpeed( carType: "sedan") + " km/h");

            // Create Bike
            Bike bike = vehicleService.createBike( name: "KTM",  modelName: "KTM Duke 200",  type: "street bike");
            System.out.println(bike.getManufacturerInformation());
            System.out.println("Max Speed: " + bike.maxSpeed( bikeType: "sports") + " km/h");

            // Compare max speeds
            int maxSpeed = vehicleService.compareMaxSpeed(car, bike);
            if (maxSpeed == 0) {
                System.out.println("Max speeds are equal.");
            } else if (maxSpeed == -1) {
                System.out.println("Cannot compare max speeds. Type is not SPORTS.");
            } else {
                System.out.println("Max speed: " + maxSpeed + " km/h");
            }
}

}
```

Output:



```
"C:\Program Files\Java\jdk-20\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.1\lib\idea_rt.jar=64511:C:\Program Files\JetBrains\IntelliJ IDEA 2024.1\bi
Car{Manufacturer name:'MaruthiSuzuki', Model Name:'MarutiDzire', Type:'sedan'}
Max Speed: 250 km/h
Bike{Manufacturer name:'KTM', Model Name:'KTM Duke 200', Type:'street bike'}
Max Speed: 400 km/h
Max speed: 400 km/h

Process finished with exit code 0
```