Interface

Interface

- An interface is a classlike construct that contains only constants and abstract methods.
- In many ways, an interface is similar to an abstract class
- No constructors

Interface

- To distinguish an interface from a class, Java uses the following syntax to declare an interface:

```
public interface InterfaceName {
    constant declarations;
    method signatures;
}
```

Example:

```
public interface Edible {
  public abstract String howToEat();
}
```

Interface

- Like an abstract class, you cannot create an instance from an interface using the new operator.
- You can use an interface as a data type for a variable, as the result of casting, and so on.

- All data fields are *public final static* and all methods are *public abstract* in an interface. For this reason, these modifiers can be omitted:

Interface

```
public interface T1 {
  public static final int K = 1;

  public abstract void p();
}
```

Equivalent

```
public interface T1 {
  int K = 1;

  void p();
}
```

Example

```
public interface Shape {
   public double getArea();    // calculate area
    public double getVolume();  // calculate volume
    public String getName();    // return shape name
     }
```

Example

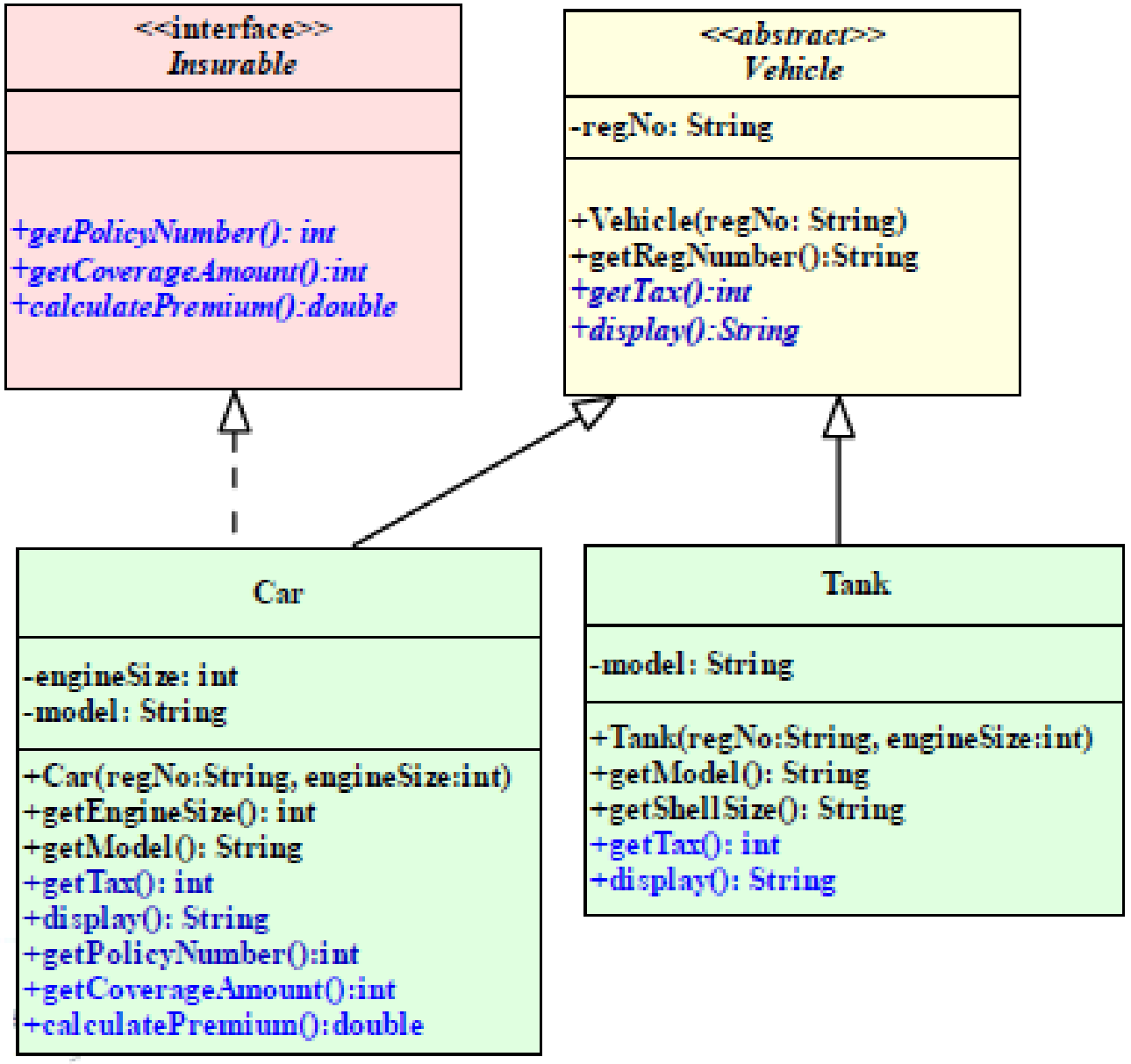
```
public class Point implements Shape {
        private int x;
        private int y;
        public Point()  {}
        public Point( int x, int y )  {  this.x = x; this.y = y;}
        public double getArea()   {   return 0.0;  }
       public double getVolume() {  return 0.0; }
       public String getName()  {  return "Point";  }
```

Implementing multiple interfaces

- public class C extends Class1 implements Int1, Int2{

}

Example


<<interface>>
Insurable
+getPolicyNumber(): int
+getCoverageAmount():int
+calculatePremium():double
<<abstract>>
Vehicle
-regNo: String
+Vehicle(regNo: String)
+getRegNumber():String
+getTax():int
+display():String
Car
-engineSize: int
-model: String
+Car(regNo:String, engineSize:int)
+getEngineSize(): int
+getModel(): String
+getTax(): int
+display(): String
+getPolicyNumber():int
+getCoverageAmount():int
+calculatePremium():double
Tank
-model: String
+Tank(regNo:String, engineSize:int)
+getModel(): String
+getShellSize(): String
+getTax(): int
+display(): String

Example

```
public class Car extends Vehicle implements
        Insurable, Drivable, Sellable  {


                .....

}
```