Problem 1: Shape Interface
Objective: Implement an interface Shape with a method double area(). Create two classes, Circle and Rectangle, that implement this interface, demonstrating polymorphism.

Problem 2: Vehicle Behavior
Objective: Design an interface Vehicle with methods void start(), void stop(), and String getFuelType(). Implement this interface in classes Car and ElectricScooter, showcasing different fuel types and start/stop behaviors.

Problem 3: Sortable Collection
Objective: Create an interface Sortable with a method void sort(). Implement this interface in a NumberCollection class that sorts an array of numbers.

Problem 4: Multi-Interface Implementation
Objective: Define two interfaces, Flyable with a method void fly() and Swimmable with a method void swim(). Create a class Duck that implements both interfaces, demonstrating multiple inheritance through interfaces.

Problem 5: Default Method in Interface
Objective: Add a default method void displayType() to the Shape interface that prints "Unknown Shape". Override this method in the Circle class to print "Circle".

Problem 6: Functional Interface for Operations
Objective: Define a functional interface Operation with a method int operate(int a, int b). Implement this interface using lambda expressions to perform addition, subtraction, and multiplication.

Problem 7: Extend Interface
Objective: Create an interface AdvancedVehicle that extends the Vehicle interface, adding a method void autopilot(). Implement this extended interface in a class SelfDrivingCar.

Problem 8: Interface for Callbacks
Objective: Design an interface Callback with a method void onCallback(String result). Implement this interface in a class AsyncTask that simulates asynchronous operations, calling back with a result.

Problem 9: Animal Hierarchy with Interfaces
Objective: Define an interface Animal with methods String sound() and String type(). Implement this interface in classes Dog and Cat, returning appropriate sounds and types.

Problem 10: Interface with Static Method
Objective: Add a static method static String getVersion() to the Vehicle interface that returns the version of the interface. Demonstrate how to call this method without an instance of the implementing classes.