In the original code, the following areas were identified for improvement:

DeliveryRequest Class:
Problem: Constructor lacked input validation for product and address fields.
Improvement: Added input validation and implemented equals and hashCode for proper usage in collections.

Shop Class:
Problem: Observer pattern implementation lacked checks for duplicate or null registrations.
Improvement: Added safeguards and improved notification logic.

TaxiDriver Class:
Problem: Direct System.out.println usage for logging was inflexible.
Improvement: Replaced with Java's Logger API for better modularity.

ShopTest Class:
Problem: Test cases lacked assertions and adequate coverage.
Improvement: Enhanced assertions and laid groundwork for future mock-based tests.


DeliveryRequest Class:
Changes:
Added validation in the constructor to ensure input fields are non-null and non-empty.
Implemented equals and hashCode for reliable operations in collections.

Shop Class:
Changes:
Improved observer registration to prevent duplicate and null entries.
Added a getDrivers method to expose driver list for testing purposes.

TaxiDriver Class:
Changes:
Replaced direct print statements with Java's Logger API.
Added validation to ensure driver names are valid strings.

ShopTest Class:
Changes:
Enhanced tests with assertions to verify driver registrations and notifications.
Prepared for future mock-based logging verification.

## Driver «interface»

- update(DeliveryRequest) : void

## TaxiDriver

- TaxiDriver(String)
- name : String
- logger : Logger
- update(DeliveryRequest) : void

## DeliveryRequest

- DeliveryRequest(String, String)
- product : String
- address : String
- getProduct() : String
- toString() : String
- equals(Object) : boolean
- hashCode() : int
- getAddress() : String

## Shop

- Shop()
- drivers : List<Driver>
- registerDriver(Driver) : void
- removeDriver(Driver) : void
- getDrivers() : List<Driver>
- notifyDrivers(DeliveryRequest) : void

## ShopTest

- ShopTest()
- testDeliveryNotification() : void