Designing for Testability and Integration Testing

1. Inspect all the classes in folder `large` and draw the design diagram for this system using the Hexagonal Architecture and/or the dependency inversion principle.

domain classes: FinalPriceCalculator, FinalPriceCalculatorFactory
port interfaces: PriceRule
adapter classes: PriceOfItems, DeliveryPrice, ExtraChargeForElectronics

2. Given the design diagram generated at the previous step, how would you tests the different components of the system (unit, integration, system tests)?

domain classes: unit (using test doubles) and system testing
adapter classes: unit tests

3. What type of class is `InvoiceDao.java` located inside the `sql` folder, using the Hexagonal Architecture and/or the dependency inversion principle?

adapter class: InvoiceDao

4. How would you test `InvoiceDao.java` (unit, integration, system tests)?

integration tests

5. Take a look at `SqlIntegrationTestBase.java` and `InvoiceDaoIntegrationTest.java`? What type of testing does each perform? How are they connected and why?

The two classes are in an inheritance relation, and the integration testing is split among them (adheres to the principle of separation of concerns). The setup and cleanup of the database is done inside the parent class and the actual integration tests are done in the child class.

6. What takes place in `openConnectionAndCleanup()`?

a hsqldb database is created in memory and 2 empty tables are added (if don't already exist or are not empty)

7. What takes place in `close()`?

the database is closed; Note that the tables created may persist

8. What takes place in `save()`?

Integration tests focused on saving to the database.

9. What takes place in `atLeast()`?

Integration tests focused on filtering data in the database.