

Introduction to Layered Architecture

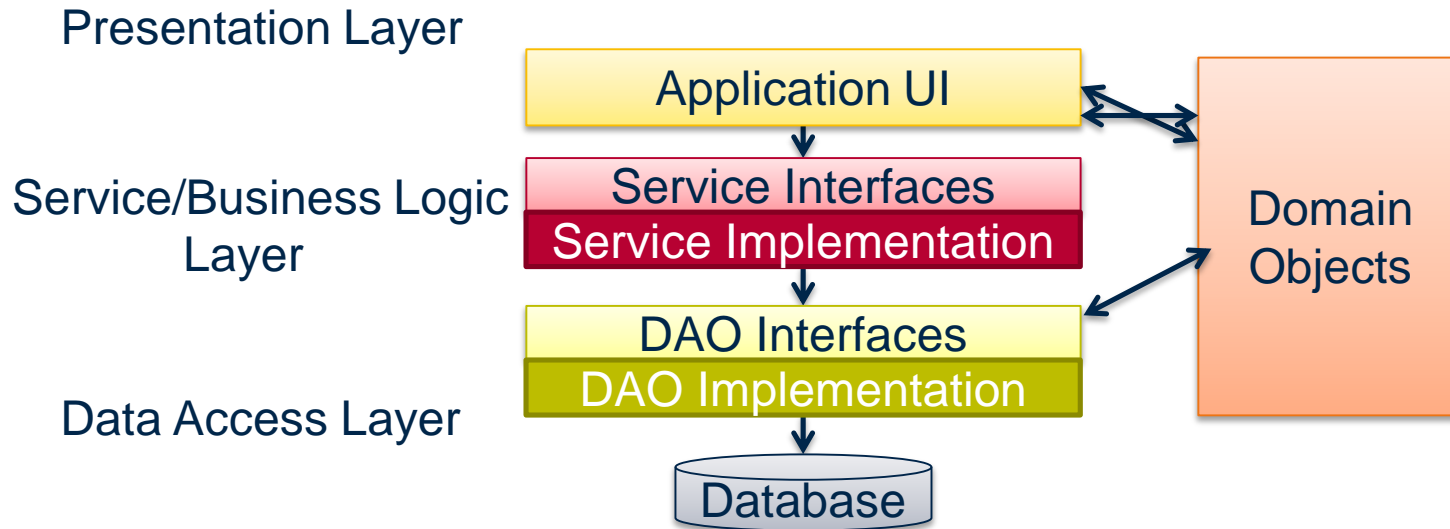
Lesson Objectives

- After completing this lesson, participants will be able to
 - Understand the concept of Layered Architecture
 - Implement layers in Java applications



What is Layered Architecture?

- Layered architecture is one of the architectural pattern based on call-and-return style
- In layered architecture, business rules, behavior, and data are obtained and manipulated, based on activity via the user interface.
- Layered architecture provides a clean separation between the business implementation, presentation and data-access logic.

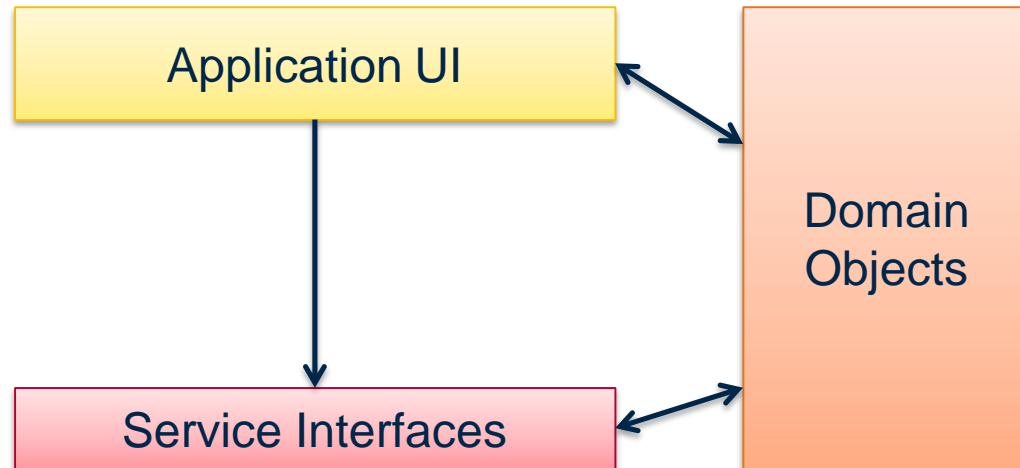


Presentation Layer

- Presentation layer consists of objects defined to accept user input and to display application outputs
- Exception handling is also an important responsibility of this layer.
- Presentation-layer simply request service/business layer for required functionality by sending and receiving domain objects

Presentation Layer

Service/Business Logic
Layer

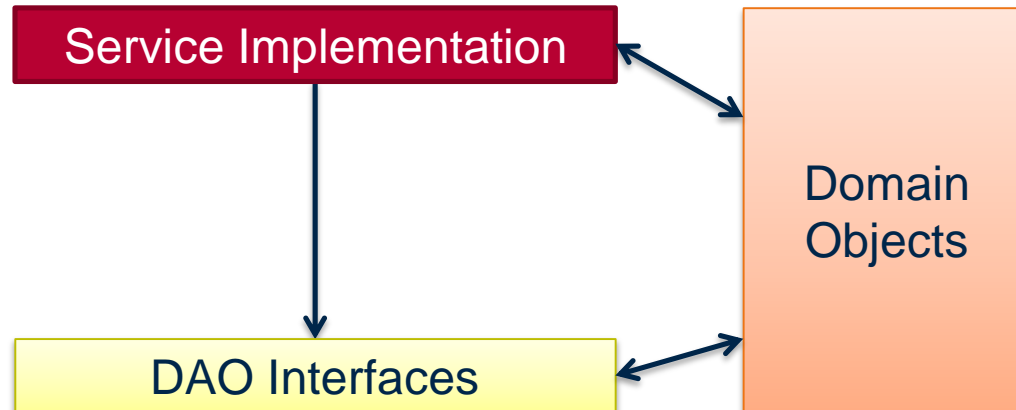


Business Logic/Service Layer

- Business logic layer is concerned with the retrieval, processing, transformation and management of application data
- This layer is responsible to implement business rules and policies
- It also ensures data consistency and validity
- Presentation layer passes data collected from UI to business layer and interact with business logic through abstract interfaces

Service/Business Logic
Layer

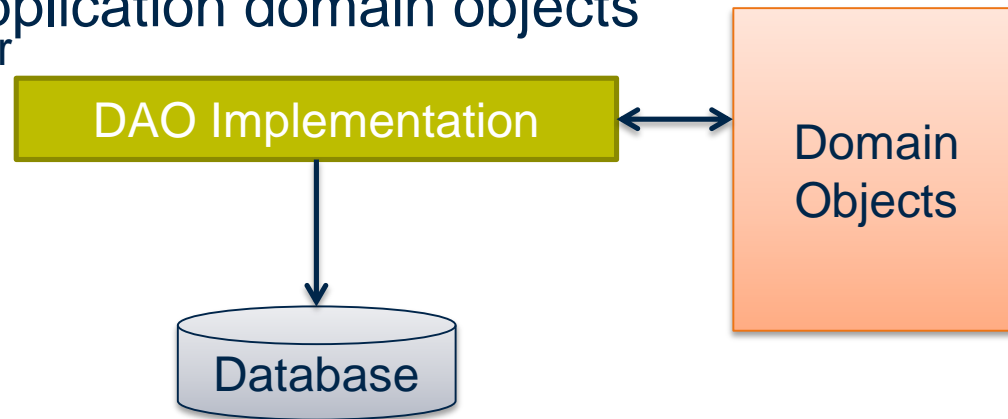
Data Access Layer



Data Access Layer

- This layer abstract the logic required to access the underlying data stores
- It centralize common data access functionality in order to make the application easier to configure and maintain.
- This layer is responsible for managing connections, generating queries, and mapping application domain objects to data source structures
- Business logic layer interacts to data access layer through abstract interfaces using application domain objects

Data Access Layer



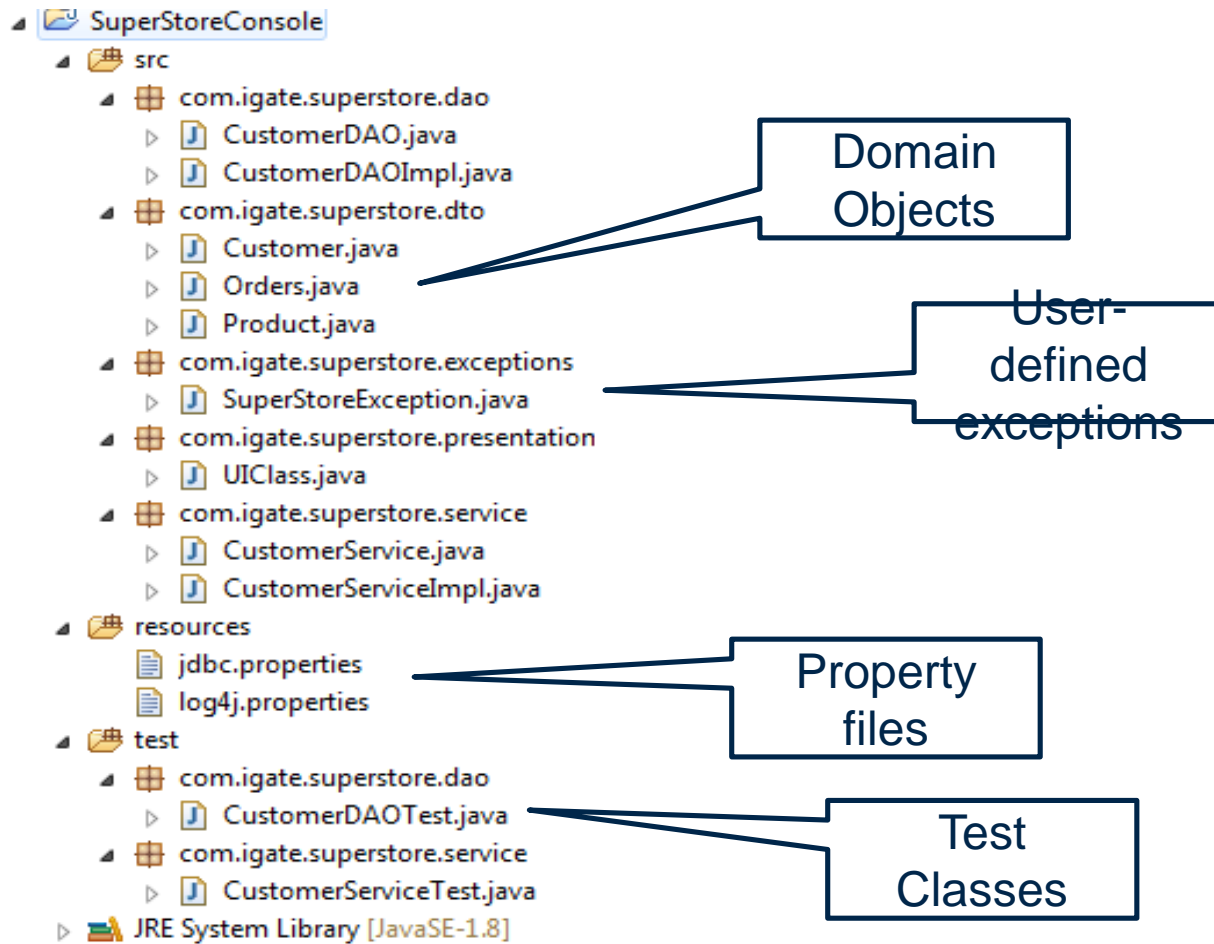
Data Transfer Objects

- Data transfer objects (DTO) or Value Objects (VO) encapsulates business data necessary to represent real world elements, such as Customers or Orders
- These object are POJO's to store data values and expose them through properties
- They contain and manage business data used by the entire application



Data Transfer Objects

Sample Layered Application Structure





Thank You!