JDBC DAGENE

- **1. Forbindelse til MySQL via JDBC**
  - Quick and dirty via maven projekt
  - DatabaseConnectionManager
  - Læser data fra ResultSet
- **2. Pattern implementation:**
  - Database Mapper (Larman)
  - Fra database til objekter
  - Strukturering af kode i datalag
- **3. Pattern implementation:**
  - Singleton design pattern
  - Repository pattern

1. FORBINDELSE TIL MYSQL VIA JDBC

- Layered architecture
- JDBC connection
  - Connect
  - Statement
  - ResultSet
  - Exception handling
    - Try/catch
- Pattern:
  - Connection Manager

LÆRINGSMÅL JDBC-1

- Forstå de overordnede abstrakte principper i layered architecture
- Forbind til en MYSQL server vha. JDBC
- Implementeret en connection manager
- C**R**UD Operationer fra Java
  - Oprettet objekter med data hentet fra DB
- Oprettet property fil
  - Hentet properties fra filen

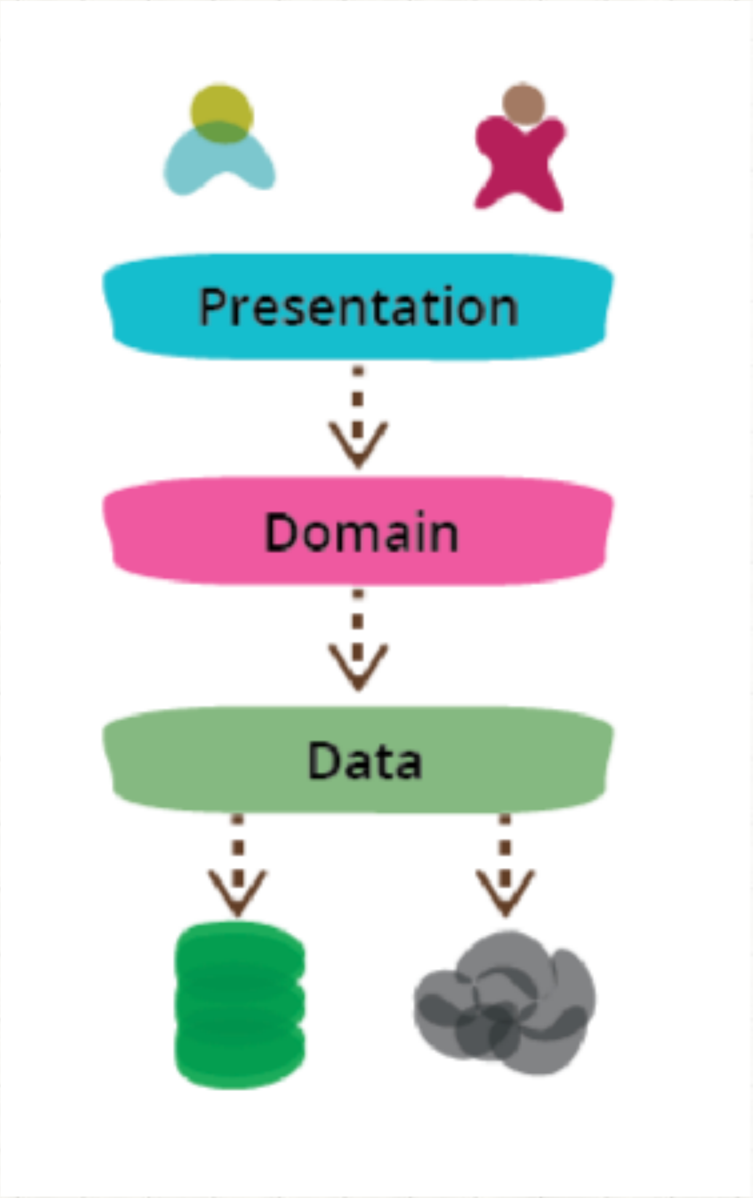
RESSOURCES

- https://www.linkedin.com/learning/learning-jdbc/
- https://docs.oracle.com/javase/tutorial/jdbc/basics/index.html
- https://www.tutorialspoint.com/jdbc/jdbc-introduction.htm
- https://docs.oracle.com/javase/tutorial/essential/environment/properties.html

- https://github.com/nicklasfrederiksenkea/VanillaJDBC/blob/master/src/JDBCExample.java
- Exercises: GitHub

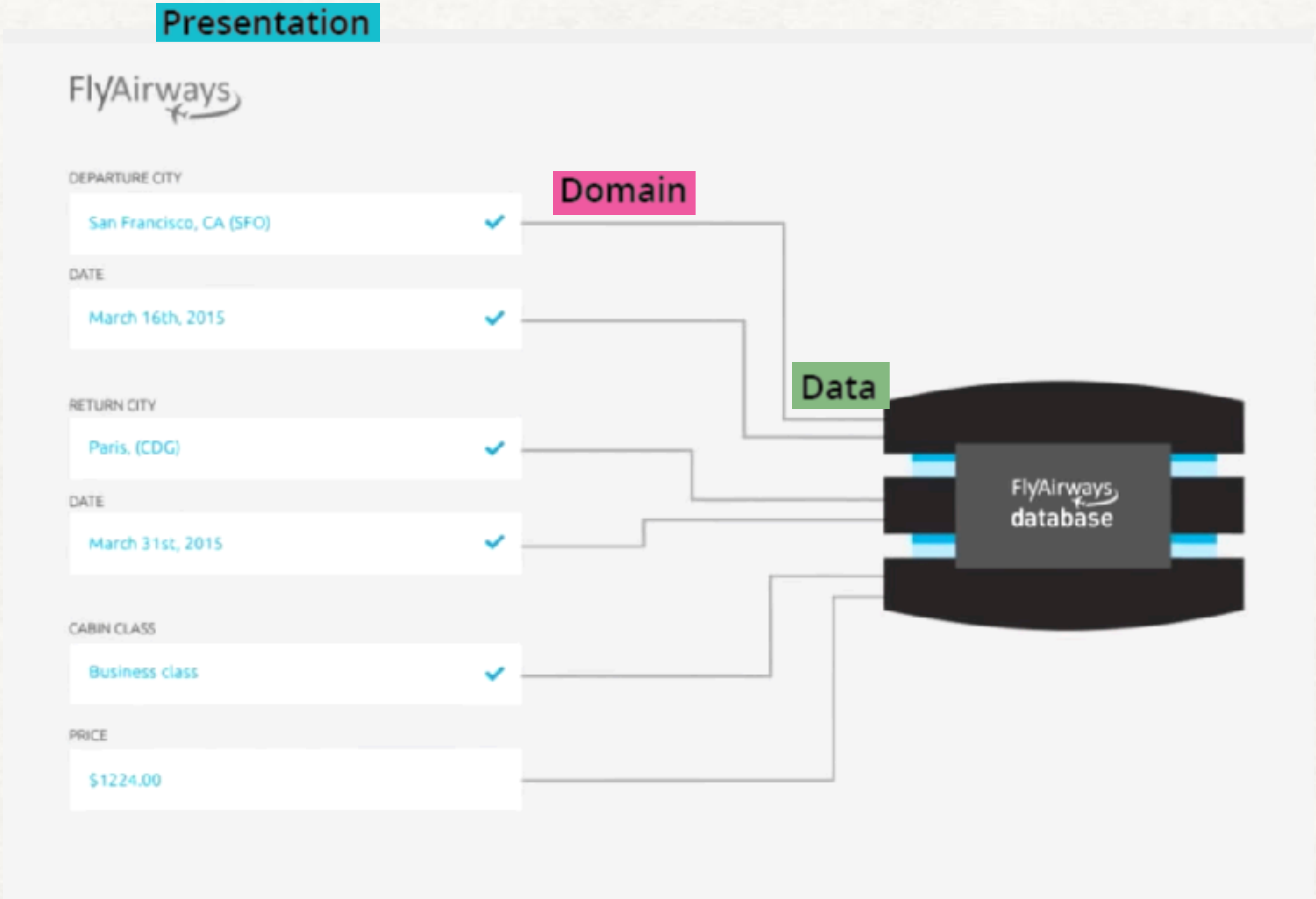
LAYERED ARCHITECTURE

- **Presentation:**
  - Brugerinteraktion
- **Domain:**
  - Domænelogik
- **Data:**
  - Datalag



https://martinfowler.com/bliki/PresentationDomainDataLayering.html

LAYERED ARCHITECTURE



Presentation

Domain

Data

FlyAirways

DEPARTURE CITY

San Francisco, CA (SFO) ✓

DATE

March 16th, 2015 ✓

RETURN CITY

Paris, (CDG) ✓

DATE

March 31st, 2015 ✓

CABIN CLASS

Business class ✓

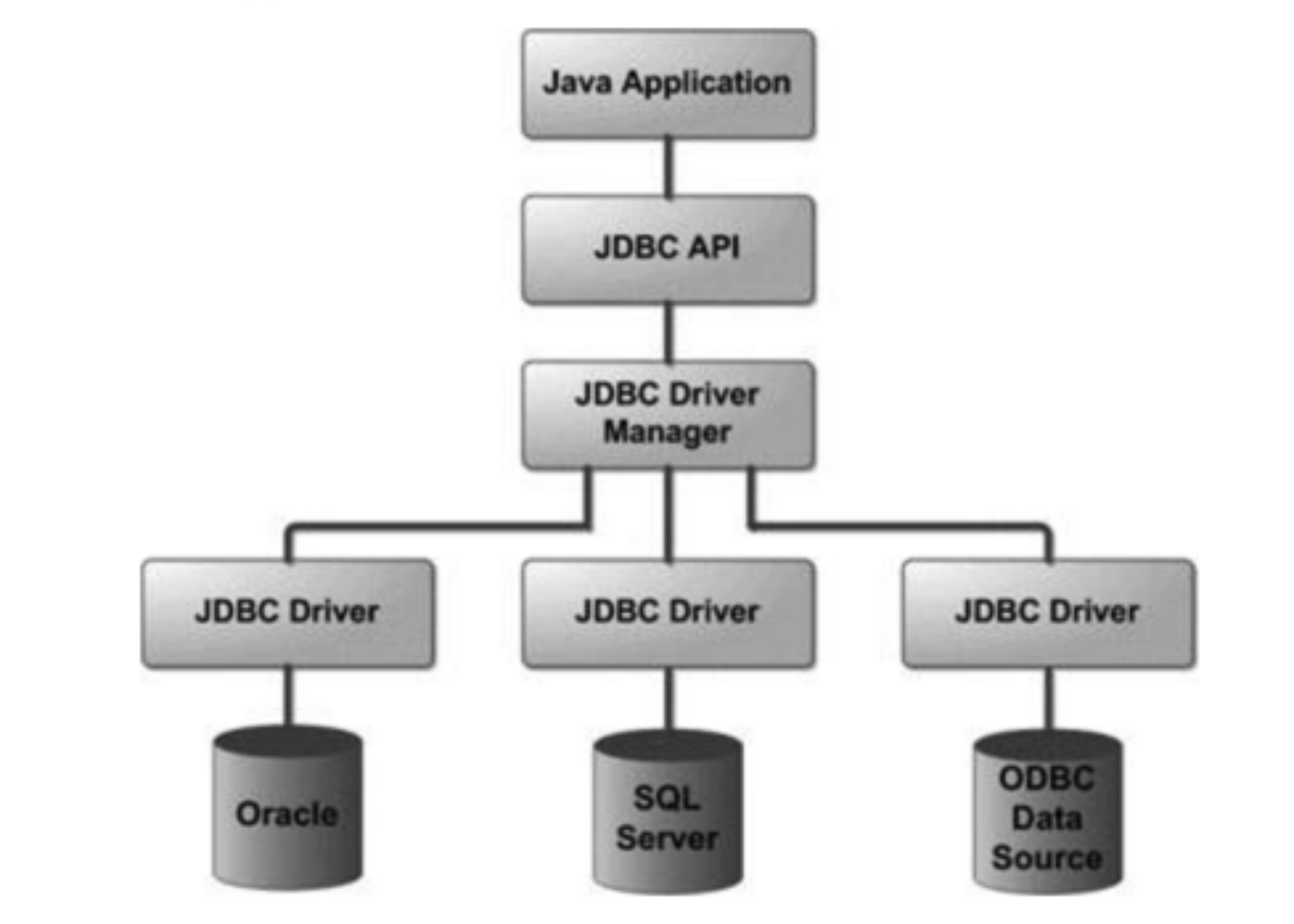
PRICE

$1224.00

https://www.youtube.com/watch?v=s7wmiS2mSXY

JDBC API

- Java DataBase Connectivity
- Database-independent
- Forbindelse til forskellige databaser kræver interaktion med det samme interface
- Package: java.sql



EXERCISE 1

- Group up (2-3)
  - Opret en Teams chat, lav exercises sammen
- Create a new maven project
- Use the dependency management tool to import JDBC mysql driver

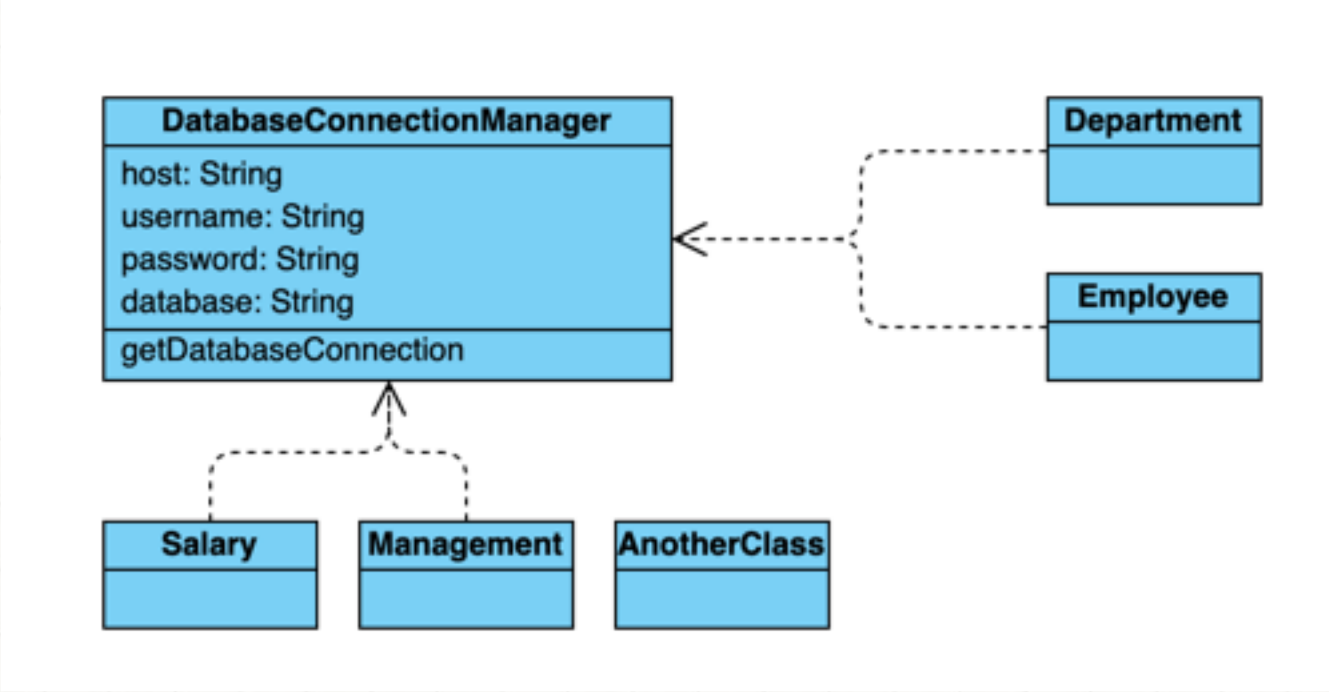
EXERCISE 2

- Create a connection to your local MySQL instance
- Create database with emp_dept.sql
- Connect, create statement, execute & print all values from 1 column.
- Challenge: Print all rows & columns use resultset.getMetaData()

OOP: DB CONNECTION MANAGER

- Definerer et objekt med det overordnede ansvar for at oprette forbindelse til databasen

- Returnere en forbindelse til databasen, der kan genbruges af andre klasser



EXERCISE 3

- Refactor your code and add a: DatabaseConnectionManager class
- Move all connection related attributes to that class
- Enable the class to:
- Return a working connection object through a method
- (such that other classes can easily connect to your database using this returned object).

EXERCISE 4

- Create 2 classes: Employee & Department
- Employee has all the attributes of the columns in emp
- Department has all the attributes of the columns in dept

- Employee has a method: getAllEmployees
- The method returns a map with empno as key & the related Person object as value
  - **Challenge** Print the map sorted by hiredate

- Department has a method: getSetOfDepartments
- The method returns a set containing all department objects

EXERCISE 5

- Create an application.property file
- Configure your DatabaseConnectionManager such that the database connectivity attributes are gathered from your property file
  - (instead of hard-coded in your DatabaseConnection attributes)