Facade Pattern

Chapter Content

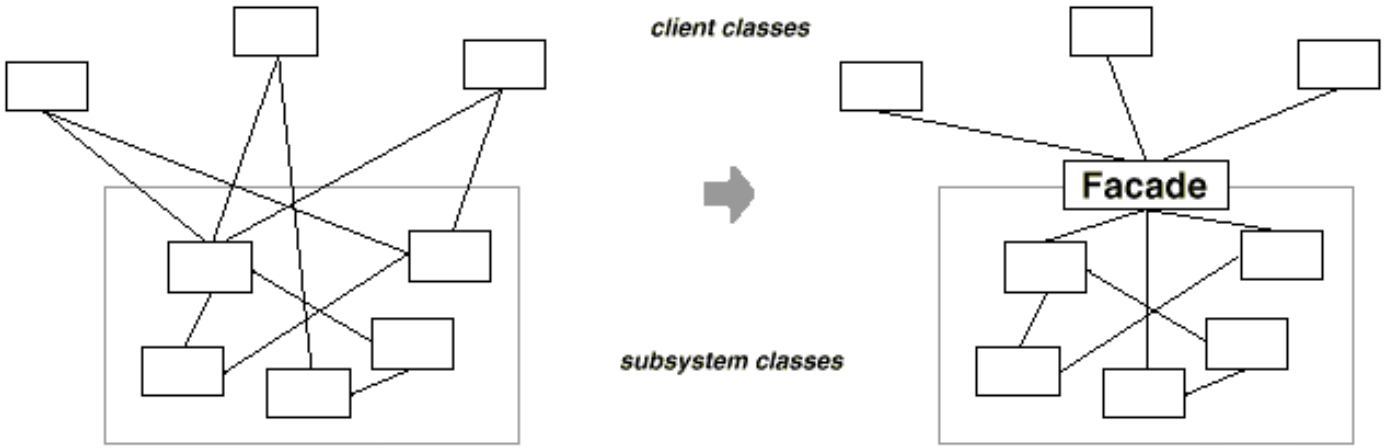
InterBit
Training & Consulting Ltd.

- Facade Pattern Overview
- Facade Pattern UML Diagram
- A Facade Example

Facade Pattern Overview

InterBit Training & Consulting Ltd.

- Provides a simplified interface to a complex system of classes (with complex inter-connections).
  - You may still allow users to use the complex, flexible interface when needed.



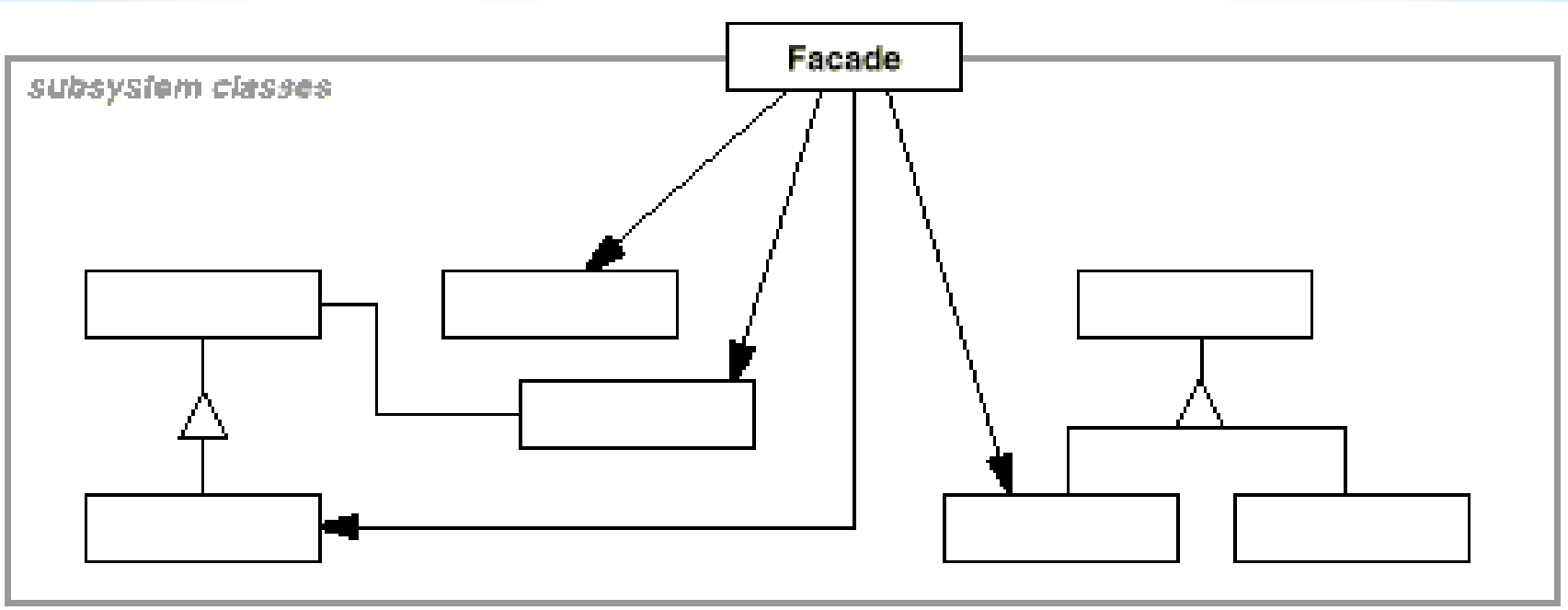
Facade Pattern Overview (cont.)

InterBit Training & Consulting Ltd.

- **Advantages:**
  - Die-hard programmers can still delve into the complex API when flexibility is required.
  - However, you also provide a simple interface for simple cases.
  - For the sake of modularity and loose-coupling, it may *sometimes* be advisable to let systems interact with each other only through facades.

Facade Pattern UML Diagram

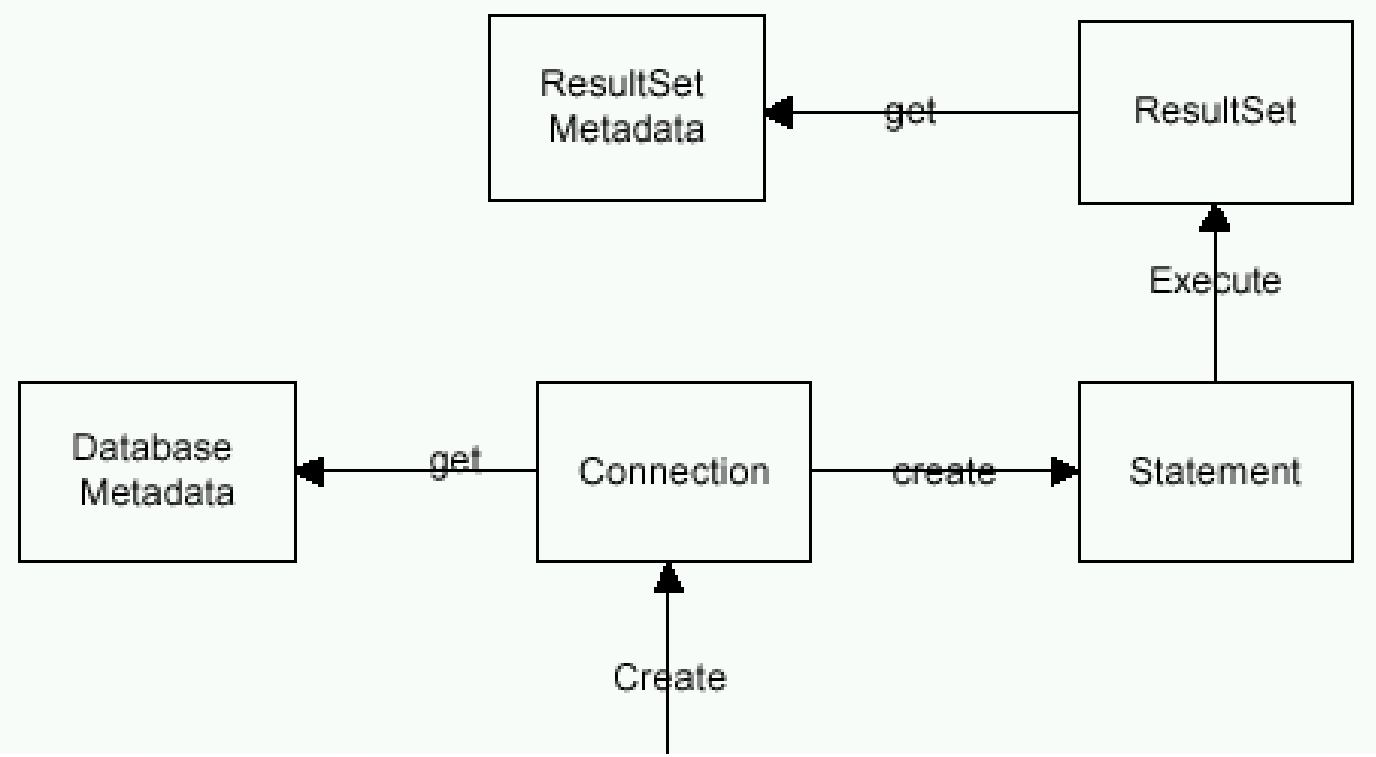
InterBit
Training & Consulting Ltd.



Façade provides a simple interface to a complex system, consisting of various classes.

A Facade Example

- JDBC is quite complex:



Standard (complex) JDBC:

InterBit Training & Consulting Ltd.

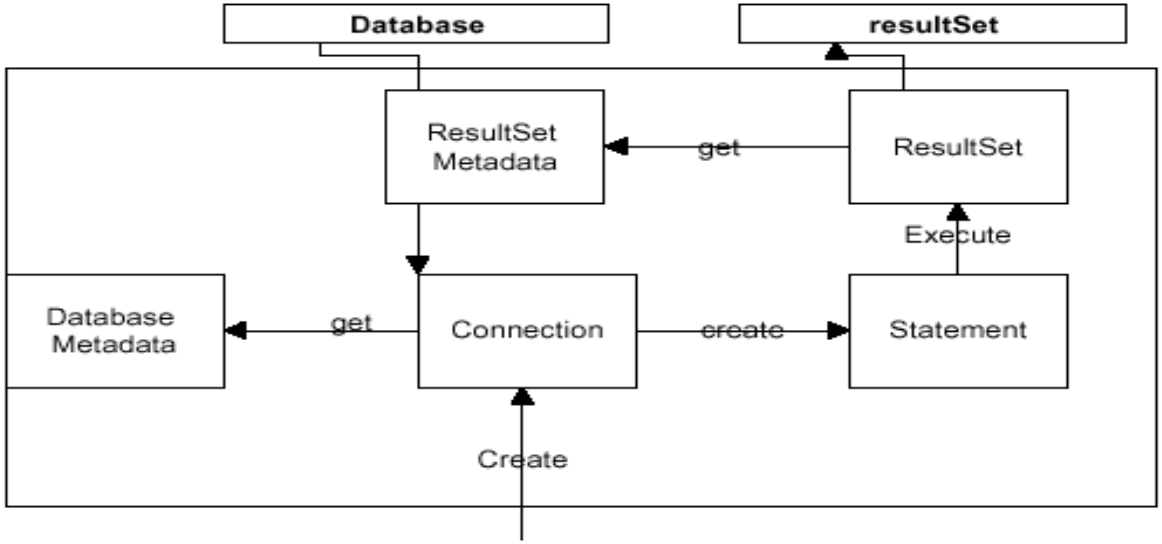
```
String url = "jdbc:odbc:company";
String driver = "sun.jdbc.odbc.JdbcOdbcDriver";
String user = "admin";
String pswd = "admin";

Class.forName(driver);
Connection con = DriverManager.getConnection(url, user, pswd);
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("select * from Employee");
...
ArrayList tables = new ArrayList();
DatabaseMetaData dbmd =  con.getMetaData();
ResultSet trs = dbmd.getTables(null,null,null, new String[] {"TABLE"});
while(rs.next()) {
   tables.add( trs.getString("TABLE_NAME")) ;
}
```

Running a SELECT

Obtaining Table Names

A Facade Example

- Our facade will be less flexible, but easier to use: users only worry about 2 classes: Database, ResultSet



Our Facade

InterBit Training & Consulting Ltd.

```
// A single-connection, simplified Database façade:

class Database {

   private Connection con;

   public Database(String driver, String url, String user, String pswd) {...}
   public void connect() throws Exception {…}
   public void disconnect() throws Exception {…}
   public ResultSet query(String sql) throws Exception {…}
   public int  execute(String sql) throws Exception {...}
   public String[] getTableNames() throws Exception {...}
}
```

Facades make life easier

InterBit
Training & Consulting Ltd.

```
// Usage:
Database db = new Database(...);
db.connect();
String[] tableNames = db.getTableNames();
...
ResultSet rs = db.query("select * from employee");
...
db.disconnect();
```

Obtaining Table Names

Running a SELECT

Our facade would help simple stand-alone single-threaded applications, but it doesn't support transactions & Prepared Statements.