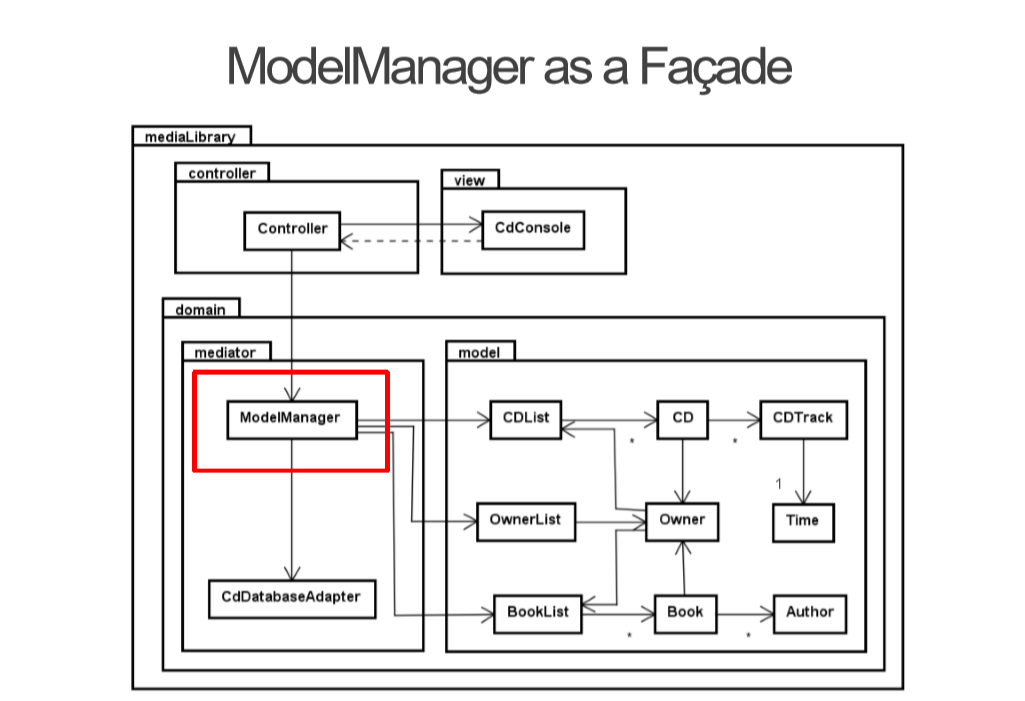
**Façade**

**A general UML class diagram of a Façade design pattern**

****

**The overall purpose for the Façade design pattern**

Facade pattern hides the complexities of the system and provides an interface to the client using which the client can access the system. This type of design pattern comes under structural pattern as this pattern adds an interface to existing system to hide its complexities.

**Purpose and relation of the general UML class diagram**

The ModelManager is the façade for model having model state (and additional actions).

The intent of Facade is to produce a simpler interface, which provides simplified methods required by client and delegates calls to methods of existing system classes.

Facade shows how to make a single object represent an entire subsystem.

**How to use a Façade design pattern**

When the client wants to access the data from his computer, The ModelManager interface allows the user to get their information in one call by creating an object of type ModelManager.

**Our System Description**

In our system, the data is also controlled through one class called DataHandler. The user is able to access all the data through this class. It doesn’t use an interface like it is shown on the original diagram. When the client wants to access the list of tours from his computer, The DataHandler class allows the user to get their list of information in one call by creating an object of type DataHandler.

What we need to remember is that by introducing the Facade into the code, is fine if the subsystem never changes, but if it does, the Facade could be broken.