

DESIGN REVIEW COUPLING AND COHESION OF AIMS PROJECT

Couplings

1. Content coupling

No.	Description	Related modules	Proposed solutions
1.	For the public getter getDeliveryInfo(), in the future, there might be a module which manages to get delivery info by calling the method and change the data by calling put() function of HashMap. Therefore, the value of delivery info will be changed while its object Order knows nothing.	Class Order	We can use method setAccess(false), so that we can only change the delivery info through Order class.

2. Common coupling

No.	Description	Related modules	Proposed solutions
1.	There are many modules that need to access or modify the data of the cart, and we would tend to use a public static attribute of class Cart to handle the needs of communication among classes. However, the public static is not bound to objects, just like imperative programming paradigm.	Class Cart	Using static block initialization Or lazy initialization

3. Control coupling

No.	Description	Related modules	Proposed solutions
1.	Book, CD, DVD has same type of media with same method	Class Media, Book, CD, DVD	Make them extend the general class Media
2.	There is new type of Order: Rush Order	Class Order	Rush Order extends Class Order

4. Stamp coupling

No.	Description	Related modules	Proposed solutions
1.	One class sends a collection or object as parameter but only a few are used.		Just pass only necessary parameters

5. Data coupling

The project has data coupled modules, so its design is good.

6. Uncoupled
None

Cohesion

1. Coincidental

No.	Description	Related modules	Proposed solutions
1.	Coincidental cohesion	Class Configs, Utils	Define public static final parameter for other classes to pass and use.

2. Logical

No.	Description	Related modules	Proposed solutions
1.	Separate codes in 2 methods in the same class	Media, Book, CD, DVD	Put those methods in different classes.

3. Temporal

No.	Description	Related modules	Proposed solutions
1.	Parts of modulus are grouped by when they are processed	Example: Class PlaceOrderController	This cohesion is needed

4. Procedural

No.	Description	Related modules	Proposed solutions
1.	Loops, multiple conditions and steps in the codes.	Example: Class PlaceOrderController	This cohesion is needed

5. Communicational

No.	Description	Related modules	Proposed solutions
1.	Work on the same input or return the same type output data	Class InterbankSubsystemController, PaymentController	Good

6. Sequential

No.	Description	Related modules	Proposed solutions
1.	Create classes can do more than one functionality or quite the opposite	CartScreenHandler, MediaHandler	Good

7. Informational

No.	Description	Related modules	Proposed solutions
1.	Duplicate the previous information	Media, Order	Extends and create after all element has been defined

8. Functional

No.	Description	Related modules	Proposed solutions
1.	The output of one method is the input of another, reuse the related functions.	API	Good