Report

1. Coupling

1.1. Content Coupling

|  |  |  |
| --- | --- | --- |
| Related modules | Description | Improvement |
| The class Order in the package entity.order, attribute delivery of HashMap type and a public getter getDeliveryInfo() for attribute. | Might have module manages to get deliveryInfo object by call getDeliveryInfo() of Order class then changes the delivery object by calling put() of HashMap class | In that new module, the method put() of HashMap can not be call since this attribute are set to setAccess(False), if want to change the delivery object, then we have to define Order in that module to modifies. |

1.2. Common Coupling

|  |  |  |
| --- | --- | --- |
| Related modules | Description | Improvement |
| Cart in entiry.cart | Plenty of modules that need to communicate with the class  Cart like the handler of Home screen when adding items to the cart, the handler of  Cart screen when showing current items in the cart, and the handler of Invoice  screen when showing the items to pay | Apply singleton pattern: 1. Static block initialization → check exception before generates instance; 2. Lazy initialization → single thread; 3. Thread safe singleton → private static violate var; 4. Double check locking Singleton → double check the instance before generating. |

1.3. Control Coupling

|  |  |  |
| --- | --- | --- |
| Related modules | Description | Improvement |
| Media, Book, CD, DVD in entity.media | They has same type of media with same method. | Make all Book, CD, and DVD extends the general class Media then they can use and changes their own methods. |

1.4. Stamp Coupling

None → Avoid passing the instantiated object(s) which encapsulate the shared data to all

objects/classes which need the shared data, i.e., dependency injection in Common Coupling step.

1.5. Data Coupling

Satisfy

1.6. Uncoupling

None

2. Cohesion

2.1. Coincidental cohesion

|  |  |  |
| --- | --- | --- |
| Related modules | Description | Improvement |
| Configs, Utils | Coincidental cohesion | Define same necessary parameters and links in same class related. |

2.2. Logical cohesion

|  |  |  |
| --- | --- | --- |
| Related modules | Description | Improvement |
| Media, Book, CD, DVD in entity.media | Separate code in 2 methods and put in same class | Since we using extends, then we put them in different class but still can use the general methods of media class. |

2.3. Temporal cohesion

None

2.4. Procedural cohesion

None

2.5. Communicational cohesion

|  |  |  |
| --- | --- | --- |
| Related modules | Description | Improvement |
| InterbankSubsystemController, PaymentTransaction | work on the same input or return the same type  of output data | Concat these 2 ouput values or return output value of Interbank for Payment to finally deals with it. |

2.6. Sequential cohesion

None

2.7. Informational cohesion

|  |  |  |
| --- | --- | --- |
| Related modules | Description | Improvement |
| Media, Order | Duplicate the previous information | Extends and create after all element of media, order has been defined. |

2.8. Functional cohesion

|  |  |  |
| --- | --- | --- |
| Related modules | Description | Improvement |
| API | The output of one method is the input to another, reuse the related functions. | Good |