

Applying the Information Hiding Principle in the Design of an Online Shopping System

EECS3311 Software Design
Jackie Wang

Summer 2015
Lassonde School of Engineering, York University

Abstract

This document supplements a tutorial video that demonstrates the design of an online shopping system, implemented via both a bad design (with the representation of cart orders exposed to the client) and a good design (with the representation of cart orders hidden from the client). You are expected to follow this tutorial to reproduce the project and illustrations, to complete the contracts and implementations of both designs, and to write more test cases to be confident that your software is correct.

Contents

1	Exposing Details of Cart Orders to Clients	1
2	Hiding Details of Cart Orders from Clients	2

1 Exposing Details of Cart Orders to Clients

From the supplier's side (i.e., *BAD_CART*):

- The attribute *orders* may be accessed by any other class:

feature orders: ARRAY[ORDER]

From the client's side (e.g., *BAD_SHOP*):

- All commands and queries can access the *orders* field of the *cart* attribute.
- That is, the implementation body of each command or query, if using *orders*, will be defined in such a way that is specific to the current representation of *orders* (e.g., *ARRAY*).

Question: What if the supplier changes the representation of *orders* (e.g., from *ARRAY* to *LINKED_LIST*)? What sort of compile time errors might occur?

2 Hiding Details of Cart Orders from Clients

From the supplier's side (i.e., *GOOD_CART*):

- The attribute *orders* may not be accessed by any other class:

```
feature {NONE}
orders: ARRAY[ORDER]
```

From the client's side (e.g., *GOOD_SHOP*):

- All commands and queries cannot access the *orders* field of the *cart* attribute.
- That is, the implementation body of each command or query will be defined in such a way that is independent of the current representation of *orders* (e.g., *ARRAY*).
- All the commands of the addition and removal of orders, as well as queries of the total price of cart orders, are moved to the supplier's side (i.e., *GOOD_CART*). This makes the supplier's class the only place that will be affected by any future changes on the representation of *orders*.

Question: What if the supplier changes the representation of *orders* (e.g., from *ARRAY* to *LINKED_LIST*)? What sort of compile time errors might occur?