Peer Assisted Study Session

FIT2099 - Week 3

Monash University

Objectives

- Understand class diagrams and their various attributes
- Be able to draw a model from a given set of requirements

Estimated Time

Question 1 - Question 7 (20 Minutes)
Question 7 (15 Minutes)

Questions

1. What do we show on a class diagram at a minimum?

which classes are present,
what their names are, and
how these classes relate to one another.

They may also have:
attributes or *instance variables*,
class variables or *static variables*, and
the *methods* or *functions* available in each class.

Public = +
Private = -
Protected = #
Package = ~

2. What does an association between classes tell you?

Associations represent the *relationships* between classes.
They are represented as a line connecting the two related classes, and at their simplest, that's all they are.

3. What is association navigability?

If the association has a navigability (i.e. an arrow on one end) it indicates that one class knows about the other but not vice versa.

4. What is an aggregate association? What are the two types of aggregates and what are their differences?

Aggregate association is a whole formed by combining a several classes
UML 2.4.1 defines two kinds of aggregate association: *composite aggregation* and *shared aggregation*. Composite aggregation is shown as a filled (black) diamond and shared aggregation is shown as an unfilled (white) diamond.

Composite aggregation is the kind of aggregation that exists *between a whole and its parts.*
The three most important characteristics of composite aggregation as opposed to shared aggregation are:
the composite object does not exist without its components,
at any time, each component may be part of only one composite, and
component objects are *likely* to be of mixed types (although this isn't *always* the case.)

A shared aggregation is the kind of association that exists *between a group and its members.*
The three most important characteristics of shared aggregation are:
the aggregate object may potentially exist without its constituent objects (although not necessarily in a useful state),
at any time, each object may be a constituent of more than one aggregate (e.g. a person may belong to several clubs), and
constituent objects are typically of the same class (but, again, that's not always the case).

5. What is a generalisation in UML?

Generalisation in UML is used to represent the same kinds of relationship as inheritance in object-oriented programming languages.

6. What is a dependency? How does it differ from an association?

A *dependency* is a relationship which indicates that a change in specification of one thing may affect another thing that uses it, but not necessarily the reverse.

7. Why do we use stereotypes? Give three examples of stereotypes

A stereotype tells you something interesting about a class, for example that it's a Java interface.

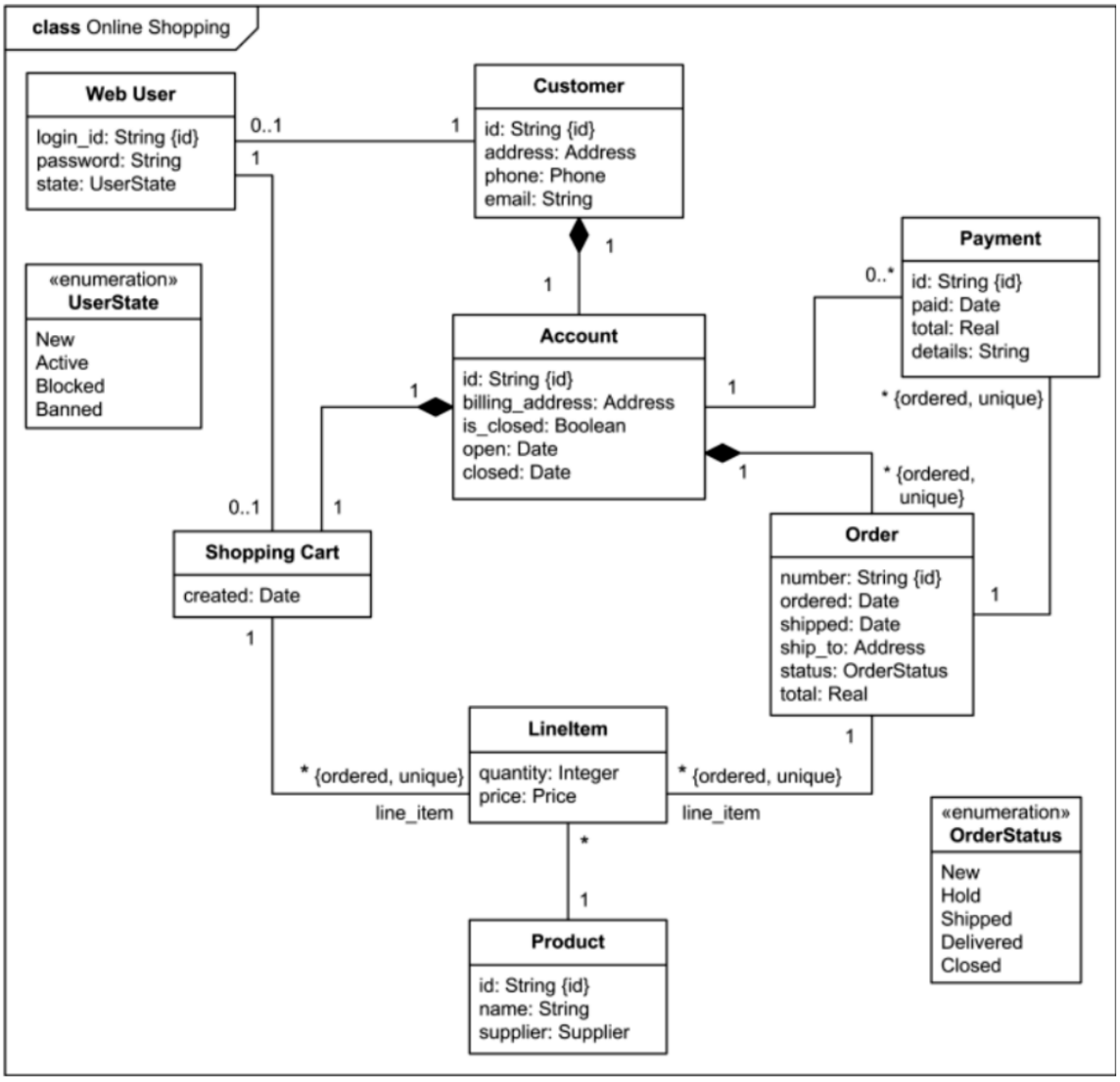
<<abstract>>
<<interface>>
<<enumeration>>

8. Case study (Online Shopping):
   Draw the UML diagram on paper and insert it at the bottom of the page

   Each customer has a unique id and is linked to exactly one account. Every account owns a shopping cart and orders. Customers can register as a web user to be able to buy items online. Customers are not required to be a web user because purchases could also be made by phone or by ordering from catalogues. Web user has a login name which also serves as unique id. A web user could be in several states - new, active, temporary blocked, or banned, and be linked to a shopping cart. A shopping cart belongs to an account.

   Every account owns the corresponding customer's orders. A customer may have no orders. Customer orders are sorted and unique. Each order could refer to several payments, possibly none. Every payment has a unique id and is related to exactly one account.

   Each order has a current order status. Both order and shopping cart have line items linked to a specific product. Each line item is related to exactly one product. A product could be associated to many line items or no item at all.


class Online Shopping
Web User
login_id: String {id}
password: String
state: UserState
0..1
1
Customer
id: String {id}
address: Address
phone: Phone
email: String
1
1
«enumeration»
UserState
New
Active
Blocked
Banned
Payment
id: String {id}
paid: Date
total: Real
details: String
0..*
Account
id: String {id}
billing_address: Address
is_closed: Boolean
open: Date
closed: Date
1
1
* {ordered, unique}
1
* {ordered, unique}
0..1
1
Shopping Cart
created: Date
Order
number: String {id}
ordered: Date
shipped: Date
ship_to: Address
status: OrderStatus
total: Real
1
1
1
LineItem
quantity: Integer
price: Price
* {ordered, unique}
line_item
* {ordered, unique}
line_item
«enumeration»
OrderStatus
New
Hold
Shipped
Delivered
Closed
*
1
Product
id: String {id}
name: String
supplier: Supplier