



แผนภาพคลาสและวัตฤ

ผู้ช่วยศาสตราจารย์สมเกียรติ ช่อเหมือน (tko@webmail.npru.ac.th)

สาขาวิชาวิศวกรรมซอฟต์แวร์ คณะวิทยาศาสตร์และเทคโนโลยี

NAKHON PATHOM
RAJABHAT UNIVERSITY
NAKHON
PATHOM
RAJABHAT
UNIVERSITY

เนื้อหาการเรียนรู้

- Class Diagram การจัดกลุ่มวัตถุด้วยการนิยามคลาส
- Analysis Class
- User Interfaces
- Modeling association
- Relationship
- Inheritance
- Aggregation and Composition

Class ?

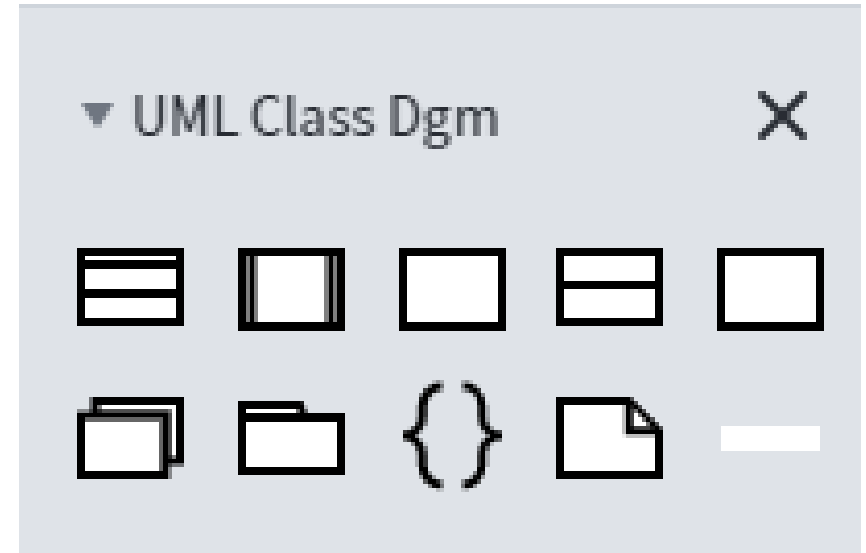
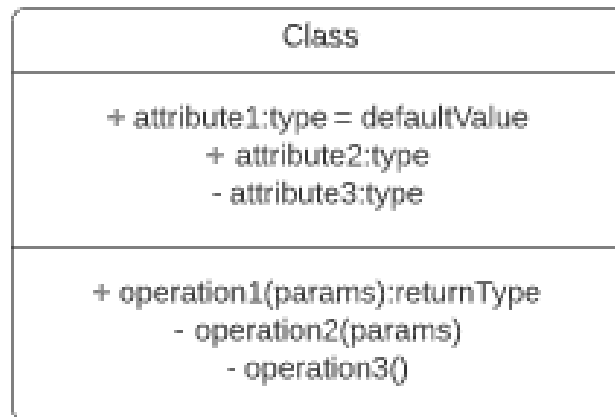


แผนภาพคลาส

- นิยามต้นแบบของวัตถุ
- อินเทอร์เฟซในการทำงานคลาส

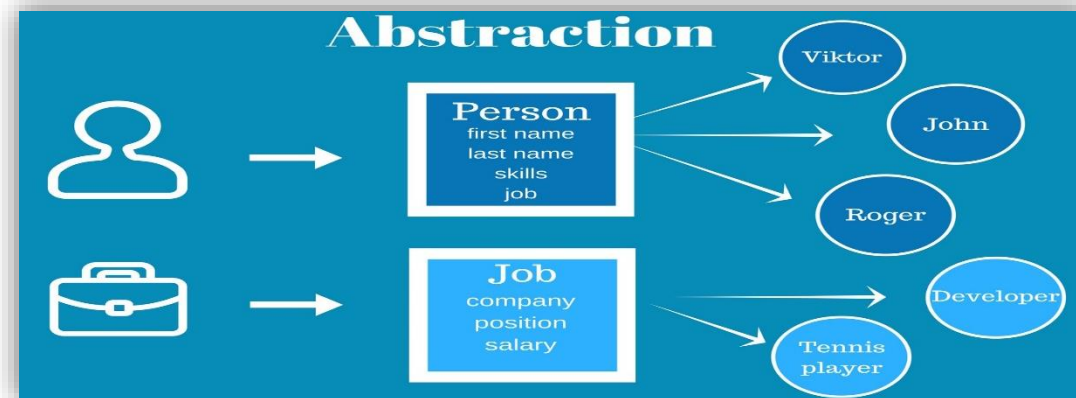
ClassDiagram1

สมมติ ๒๓๓ | ช่อ สมมติ ๐๒๒ | May 29, 2020



Abstraction

- กระบวนการนิยามคลาสตามคุณลักษณะของวัตถุ
 - **Classification Abstraction** การแยกวัตถุต่าง ๆ เป็นกลุ่ม
 - **Generalization Abstraction** การแยกคุณลักษณะพื้นฐาน
 - **Specialization Abstraction** การแยกคุณลักษณะเฉพาะ
 - **Aggregation Abstraction** การแยกคุณลักษณะใหม่ การรวมหรือประกอบกัน
 - **Association Abstraction** ความสัมพันธ์ระหว่างวัตถุ



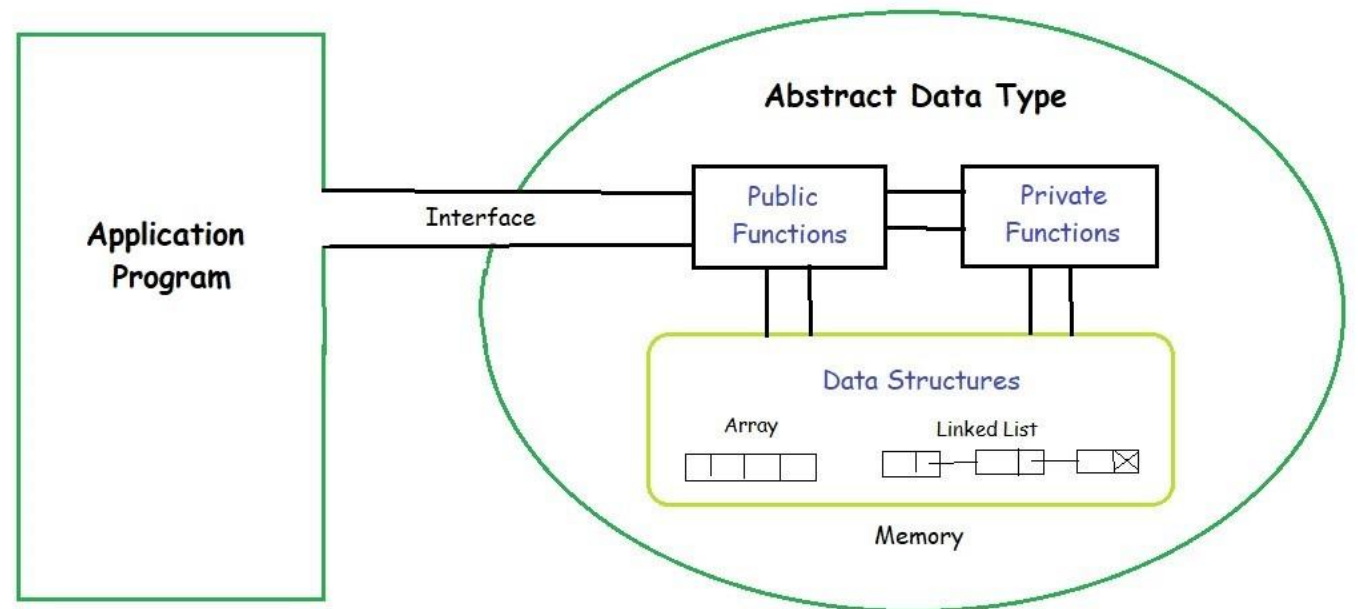
Abstract Data Type

- ประเภทของวัตถุที่นิยามเป็นต้นแบบ (class)
 - ชุดข้อมูล (data/field)
 - ชุดตัวดำเนินการ (method)
- ประเภทข้อมูลที่ใช้ นิยาม

implement

มุมมองการใช้งานที่เป็นอิสระ

Interface



Analysis Class

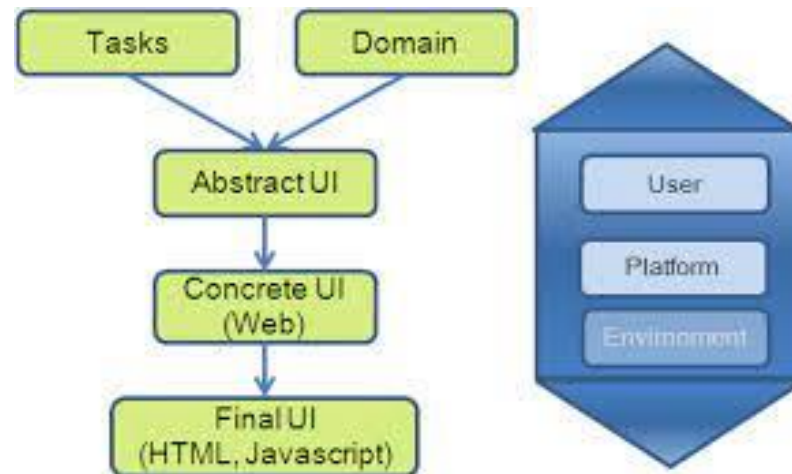
- ขอบเขตและหน้าที่ของคลาส (Boundary classes)
- พฤติกรรมที่รองรับกรณีการใช้งาน (Control classes)
- แบบจำลองข้อมูลและพฤติกรรม (Entity classes)

interface

GUI

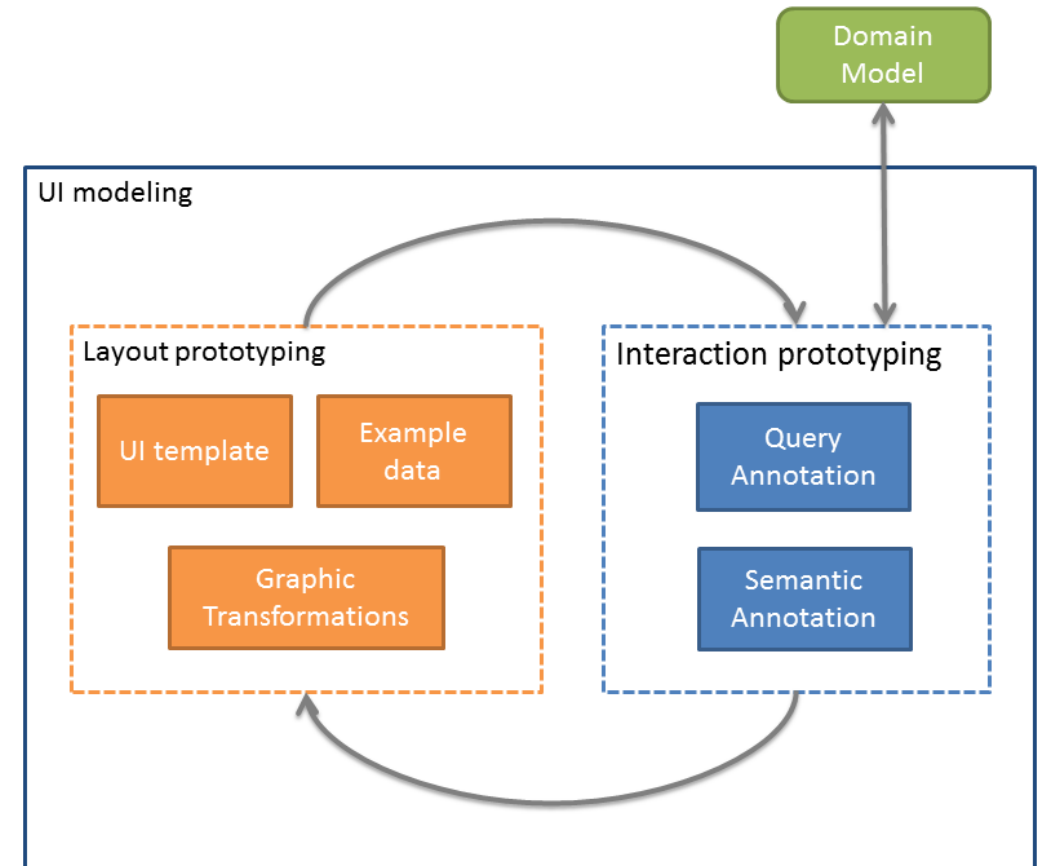
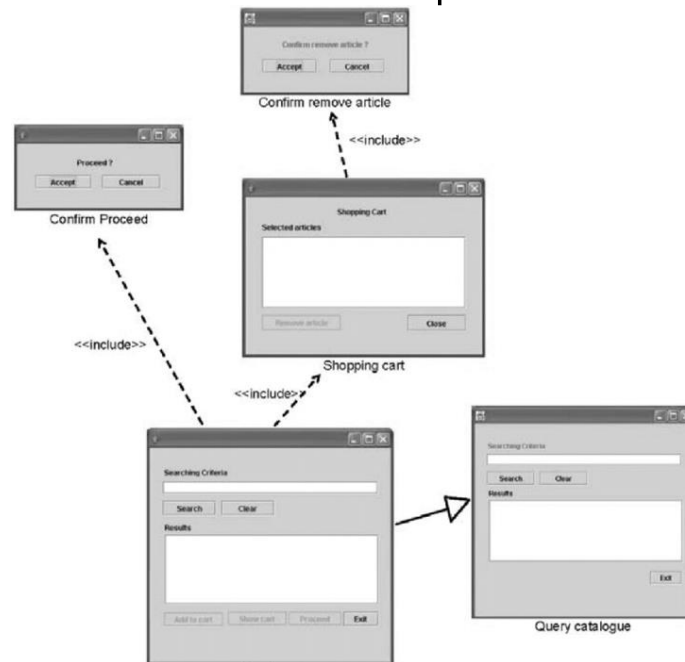
API

Instance Object



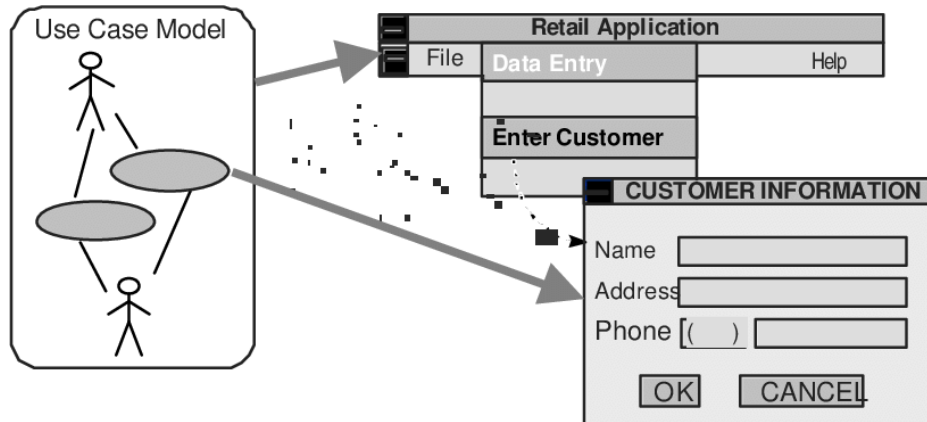
User Interfaces (UI)

- การปฏิสัมพันธ์มากกว่า 50% เป็นส่วนต่อประสานผู้ใช้งาน
- การออกแบบซอฟต์แวร์ตามสถานการณ์การใช้งาน
- แบบจำลองของโครงสร้างภายในและวัตถุ
- UI modeling



Class diagram

- แบบจำลองวัตถุ



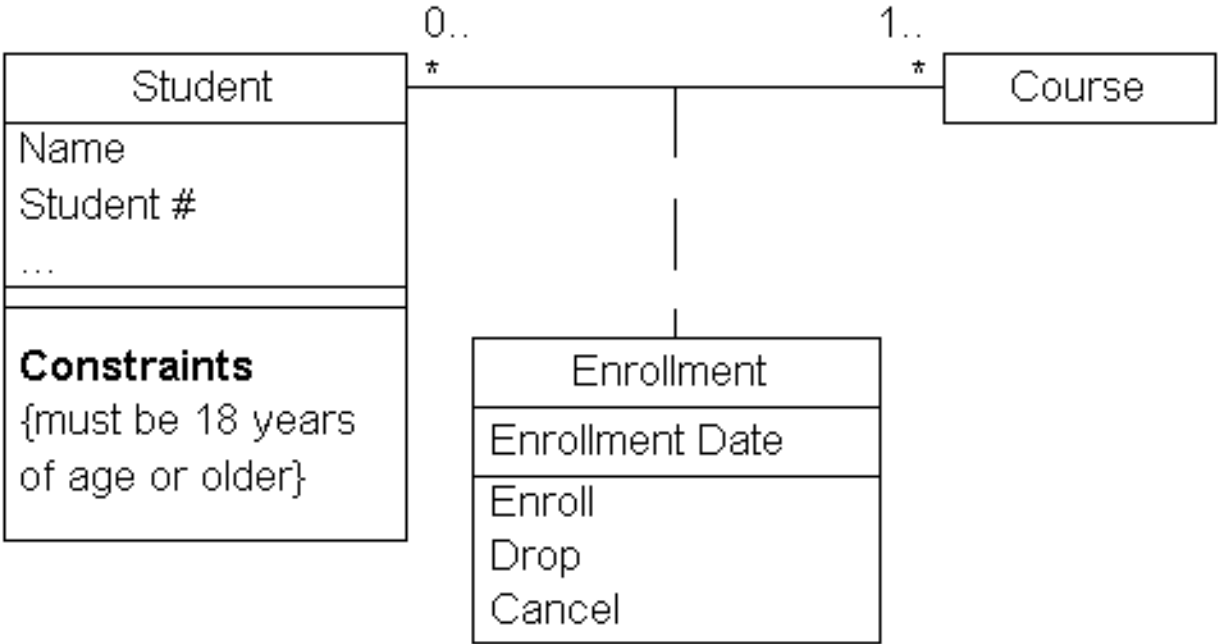
Analysis

Order
Placement Date Delivery Date Order Number
Calculate Total Calculate Taxes

Design

Order
- deliveryDate: Date - orderNumber: int - placementDate: Date - taxes: Currency - total: Currency
calculateTaxes(Country, State): Currency # calculateTotal(): Currency getTaxEngine() {visibility=implementation}

Modeling association



Class Style

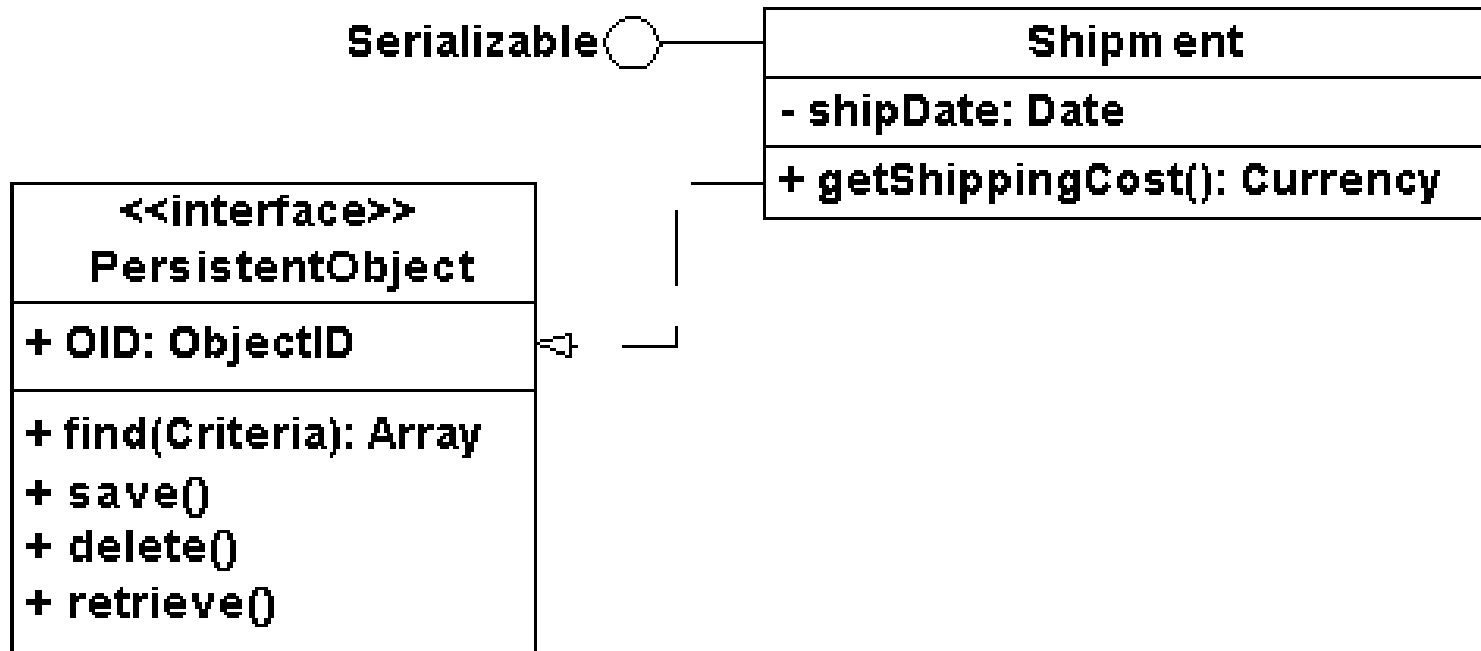
With Scaffolding

OrderItem
numberOrdered: int - item: Item - order: Order
<<constructor>> + <u>OrderItem(Order)</u> : OrderItem + <u>findAllInstances()</u> : Vector + <u>findForItem(Item)</u> : Vector + <u>findForOrder(Order)</u> : Vector + <u>getNumberOrdered()</u> : int + <u>getTotal()</u> : Currency + <u>setNumberOrdered(amount: int)</u> # <u>calculateTaxes(Country, State)</u> : Currency # <u>calculateTotal()</u> : Currency # <u>getItem()</u> : Item # <u>getOrder()</u> : Order - <u>getTaxEngine()</u> - <u>setItem(Item)</u> - <u>setOrder(Order)</u>

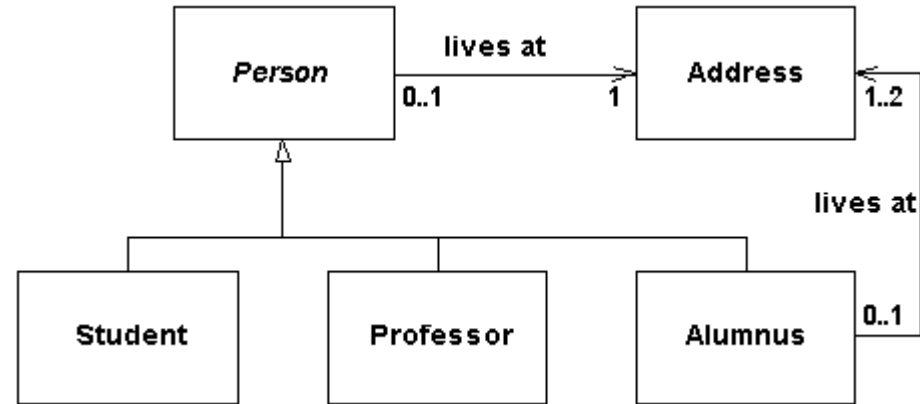
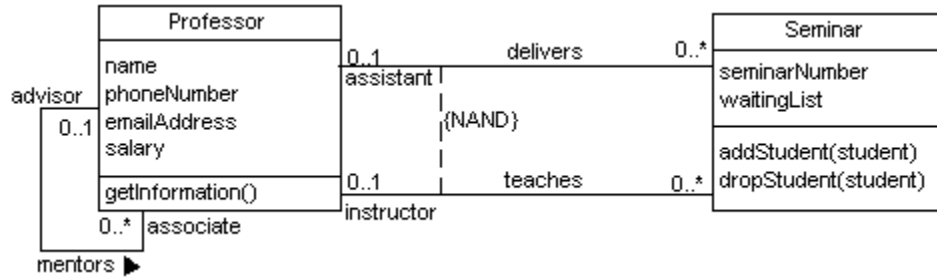
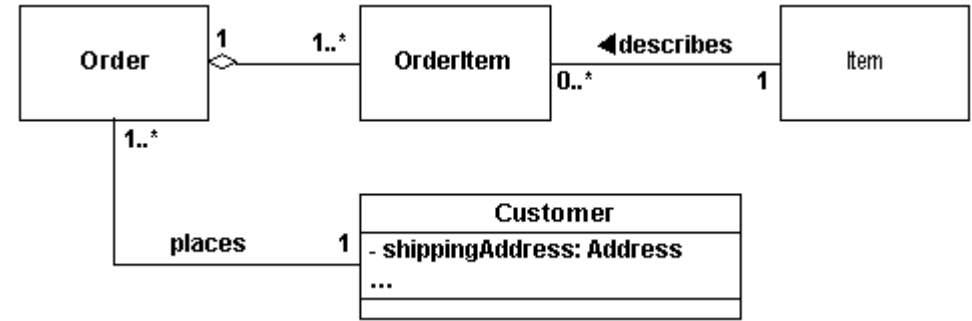
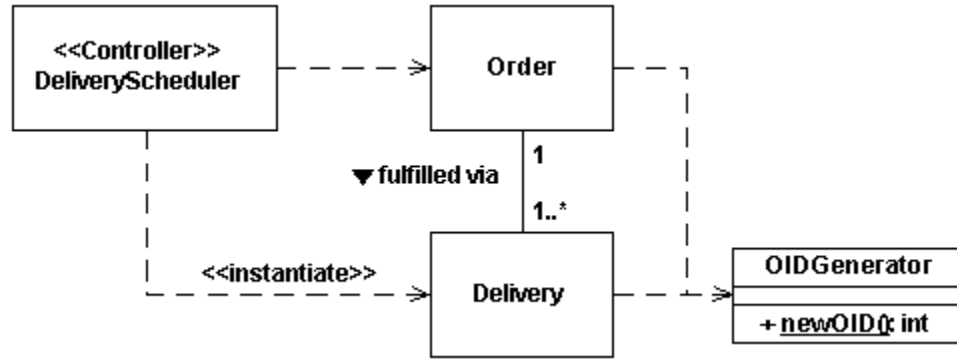
Without Scaffolding

OrderItem
numberOrdered: int
+ <u>findForItem(Item)</u> : Vector + <u>findForOrder(Order)</u> : Vector # <u>calculateTaxes()</u> : Currency # <u>calculateTotal()</u> : Currency - <u>getTaxEngine()</u>

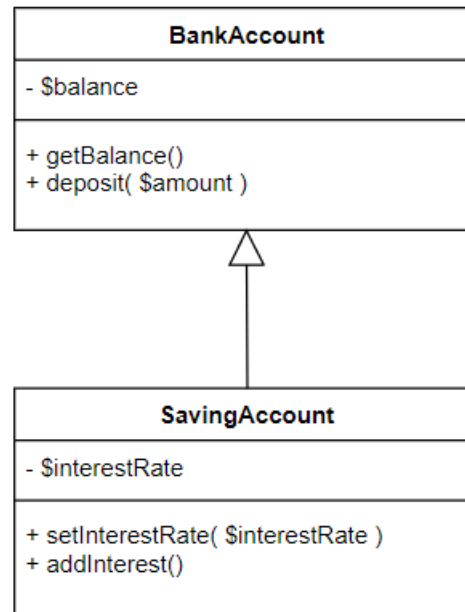
Interfaces Class



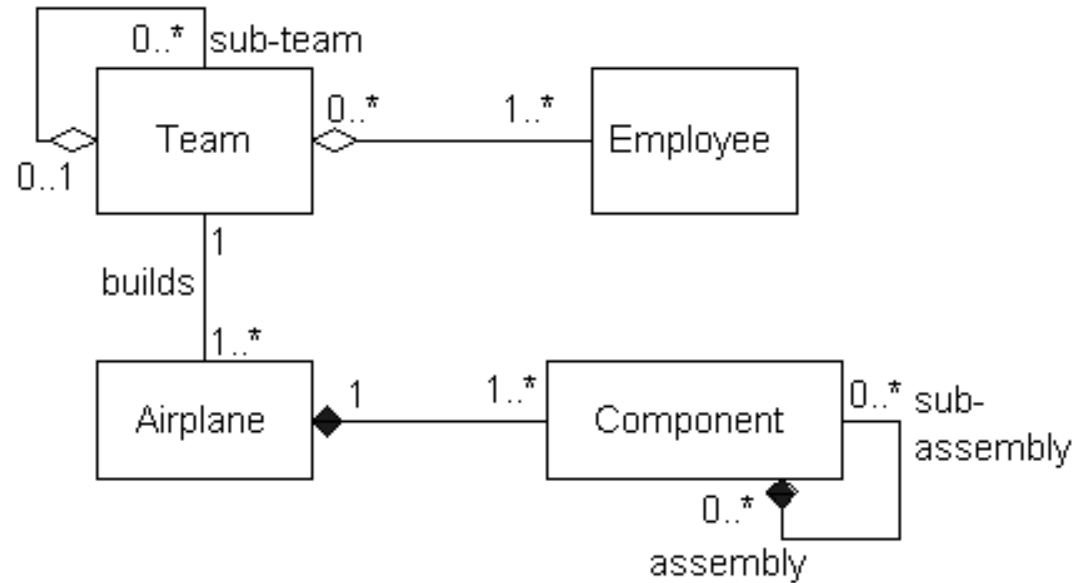
Relationship



Inheritance



Aggregation and Composition



สรุปท้ายบท

- การจัดกลุ่มวัตถุด้วยการนิยามคลาส ผ่านการวาดแผนภาพคลาส
- การวิเคราะห์และออกแบบกลุ่มของออบเจกต์ที่ใช้เป็นส่วนประกอบ
- อาจเริ่มจาก **User Interfaces** ซึ่งเห็นได้ชัดเจนที่สุด
- ในแบบจำลองการสร้างความสัมพันธ์ที่เชื่อมโยงระหว่างคลาส
 - **Relationship**
 - **Inheritance**
 - **Aggregation and Composition**

Account
□ accountName : String
□ accountNumber : String
□ balance : double
● deposit ()
● withdraw ()
● getAccountName ()
● setAccountName ()
● getAccountNumber ()
● setAccountNumber ()
● getBalance ()
● setBalance ()