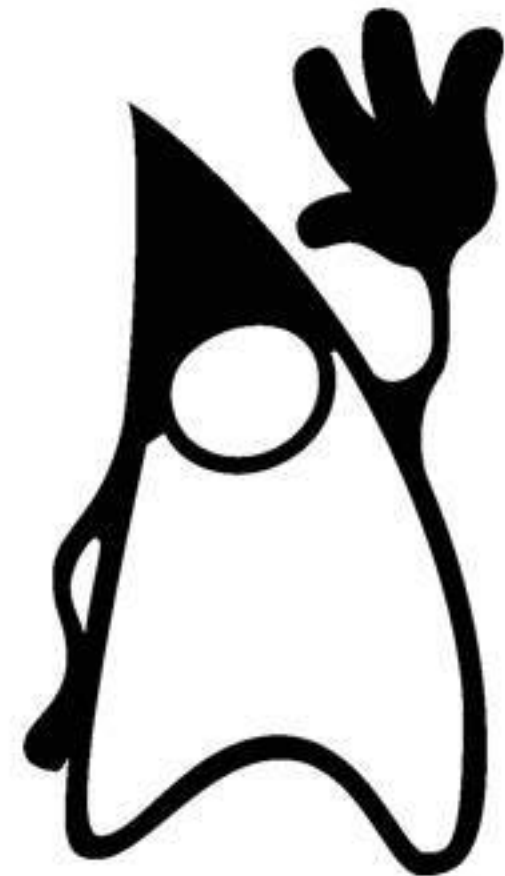




**END** ~~

Thank you!!



Edited by Ruei-Jiun Tzeng

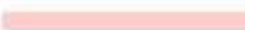




J01-03

# 物件導向的程式設計思維

曾瑞君 (Jim\_Tzeng)



# 學習目標

1. 以 Object-Oriented Analysis (OOA) 的思考方式分析程式需求
2. 找出關鍵物件 (objects)
3. 定義物件的屬性 (attributes) 和行為 (operations)
4. 塑模 (model classes) 與 UML
5. 使用工具繪製 UML





1/5

# 以 Object-Oriented Analysis (OOA) 的思考方式分析程式需求





# 工作來囉!!

---

建立一個線上購衣網站。該網站：

- 具備線上型錄(**Catalog**)，每3個月更新內容一次
- 型錄上有各式襯衫(**Shirt**)，並標明：**identifier (ID)**、**color**、**size**、**description**和**price**。
- 客戶(**Customer**)可以直接在網站上下訂單(**Order**)購買，或是打電話請客服代表(customer service representative, **CSR**)下單，付款(**Payment**)可使用信用卡。





2/5

## 找出關鍵物件 (objects)











# 甚麼是物件(object)?

---

- 物件(object)可以是實體或是抽象的，如「ATM」是實體的，「payment」是抽象的。
- 物件具備特性(attributes, characteristics)，通常是名詞，像是size，name，shape等。屬性的value，代表物件當時的state。
- 物件具備行為(operations, the things they can do)，通常是動詞，像是setting a value，displaying a price等。行為的發動，可能改變某些屬性的state。

# 購衣網站有哪“幾種”關鍵物件？

Order (form)	Shirt	Customer
		
Payment (form)	Catalog	CSR
		





- 
- 「種類」的英文有哪些？





3/5

## 定義物件的 屬性(attributes) & 行為(Behaviors)





# 分析物件的特性 & 行為

Class (分類)	Order	Shirt	Customer
Attributes (特性)	order ID date <b>*Shirt(s)</b> total price <b>*Form of payment</b> <b>*CSR</b> status	shirt ID price description size color code	customer ID name address phone number email address <b>*Order</b>
Behaviors (行為)	calculate the total price add shirt to order remove shirt from order submit the order	display information	assign a customer ID





# 分析物件的特性 & 行為

Class (分類)	Payment	Catalog	CSR
Attributes (特性)	Payment ID name address phone number email address <b>*Order</b>	<b>*Shirt(s)</b>	name extension
Behaviors (行為)	verify credit card number verify check payment	add a shirt remove a shirt	process order







4/5

## 塑模 (model classes)





# 設計 classes (類別)

---

- 每一個生活裡的物件(object)，基本上都是不同的。
- 雖然都是不同，但可以予以「分類」，所以有各種不同的類別(class)。
- 類別(class)像是「藍圖」，或是工廠裡的「模板」，是物件(object)的原型；製作出來的物件，就會具備不同的特色。如有些shirt的color是red，有些是blue；而每件衣服也會給不同的id，以做識別。
- 物件(object)裡的屬性(attribute)，可以有不同的value/state。


# 設計 classes (類別)

- A class is the way you define an object.
- Class 的同義詞有「category」、「template」、「blueprint」。
- 所以寫程式時我們寫類別(class)，程式執行可由一種類別產生多個物件(object)。物件是邏輯上的說法，實際上，每個物件在記憶體都會佔據一塊空間，具有獨立記憶體位址，我們稱為實體(instance)。
- 每個物件都是class的獨立實體。
- Objects are unique instances of classes。

Shirt
shirt ID
price
description
size
color code
display information







# Modeling(塑模) Classes

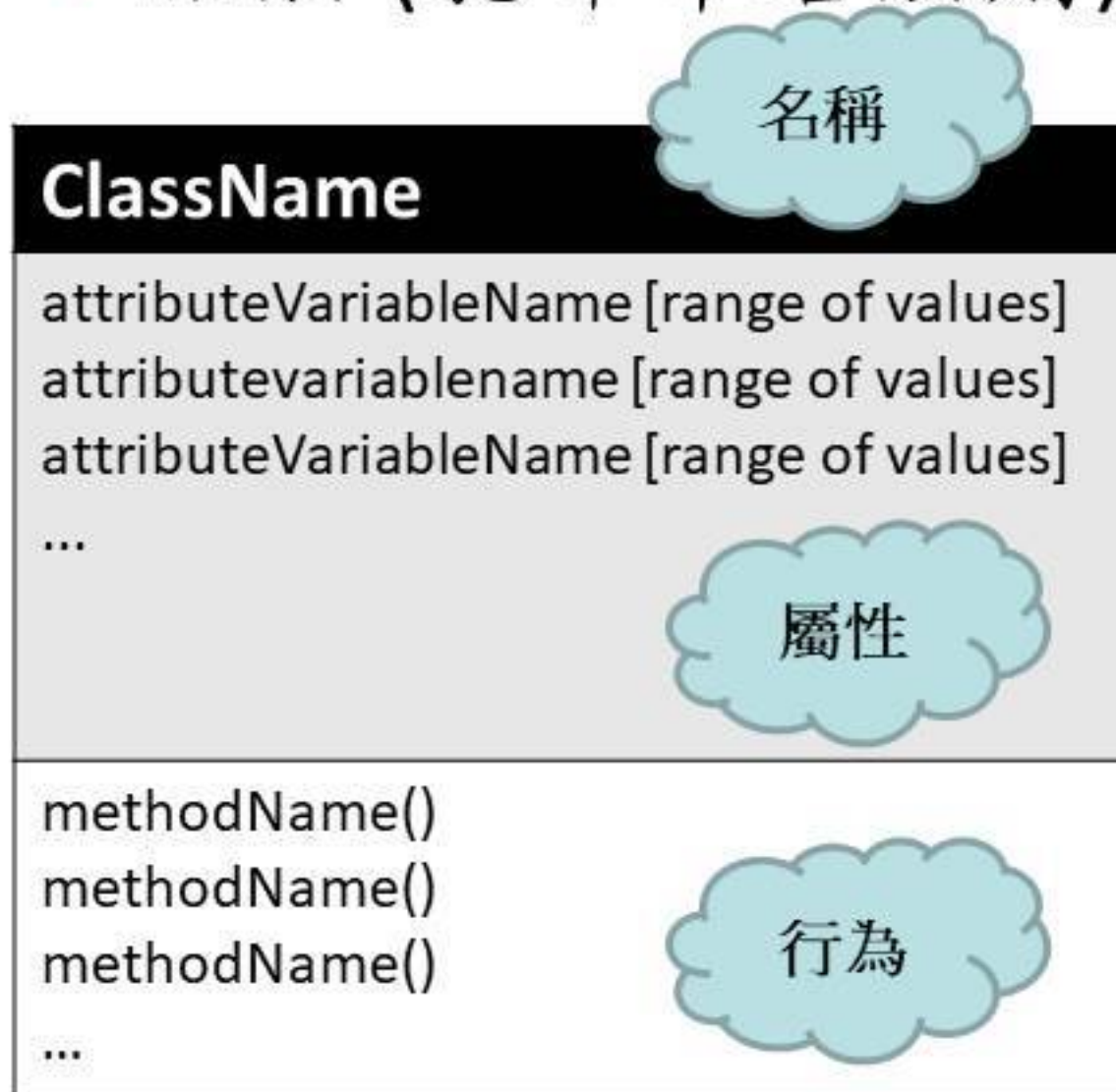
---

- 使用UML語言(Unified Modeling Language)
  - 建構物件導向(OO)程式設計裡的物件，物件屬性，物件行為，和物件關聯。
  - 描述物件間的互動關係，及物件狀態的改變。

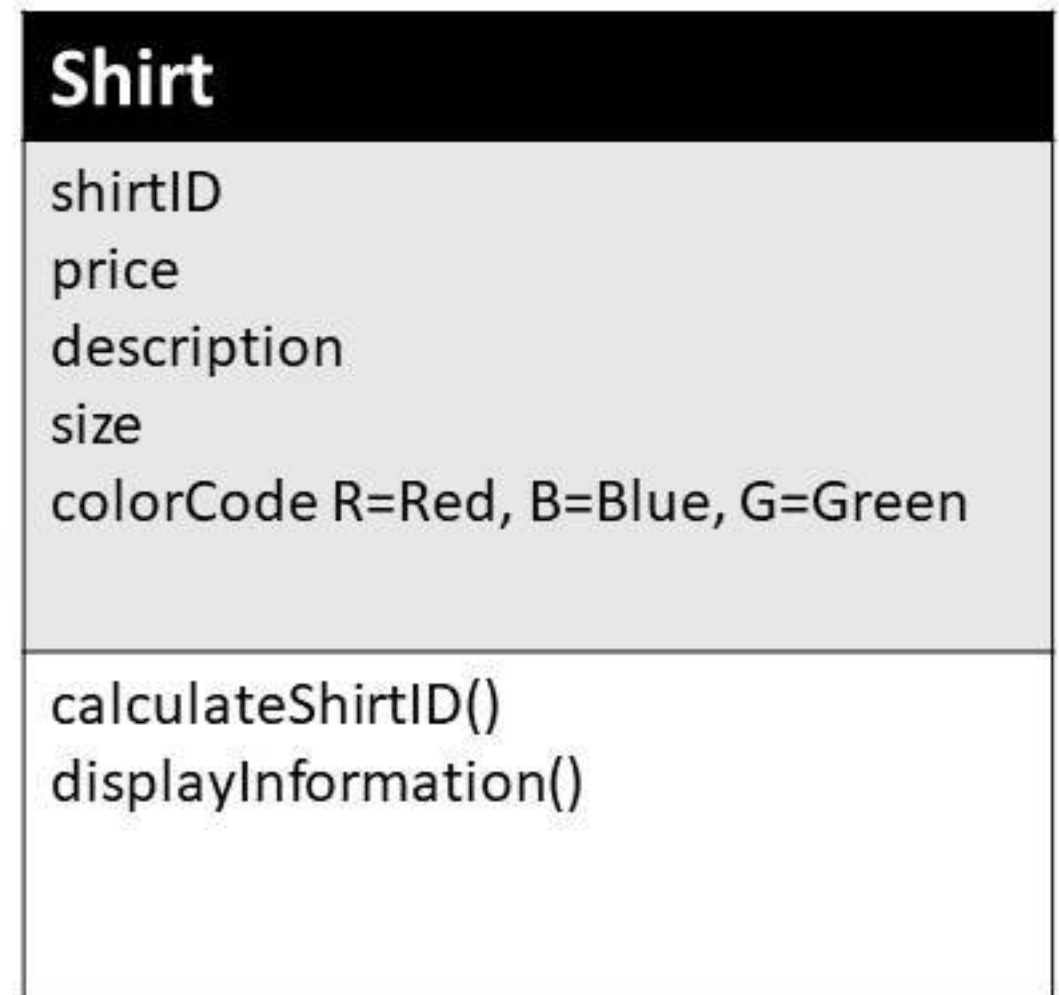


# 使用類別圖(Class Diagram)

- 語法 (駝峰命名法則)



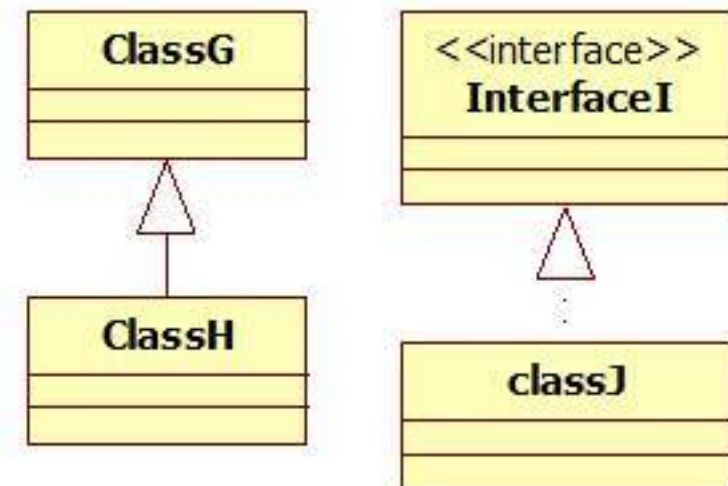
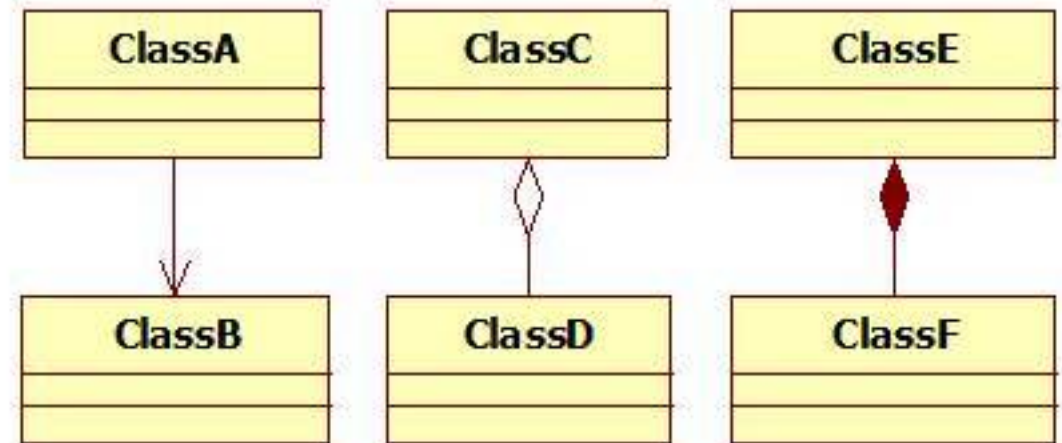
- 案例



# 類別間的關聯

## Has-A

- 關聯(Dependency)
  - ◆ ClassA 和 ClassB
- 聚合(Aggregation)
  - ◆ ClassC 和 ClassD
- 組合(Composition)
  - ◆ ClassE 和 ClassF



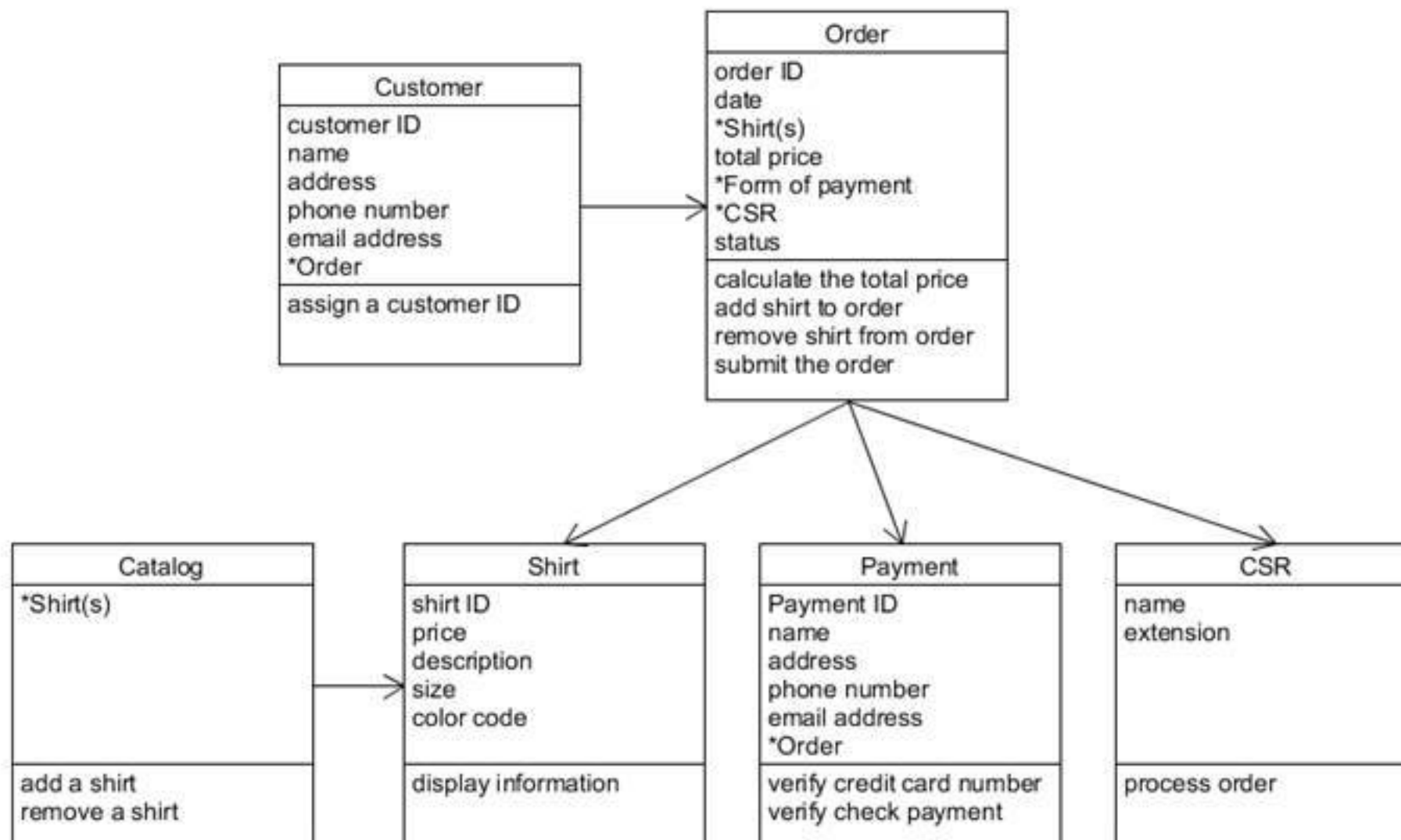
## Is-A

- (繼承)Inheritance
  - ◆ ClassG 和 ClassH



# 使用UML呈現類別間的互動關係

- 以 class diagram 搭配HAS-A的關聯性呈現購衣網站主要類別間的互動關係：





5/5

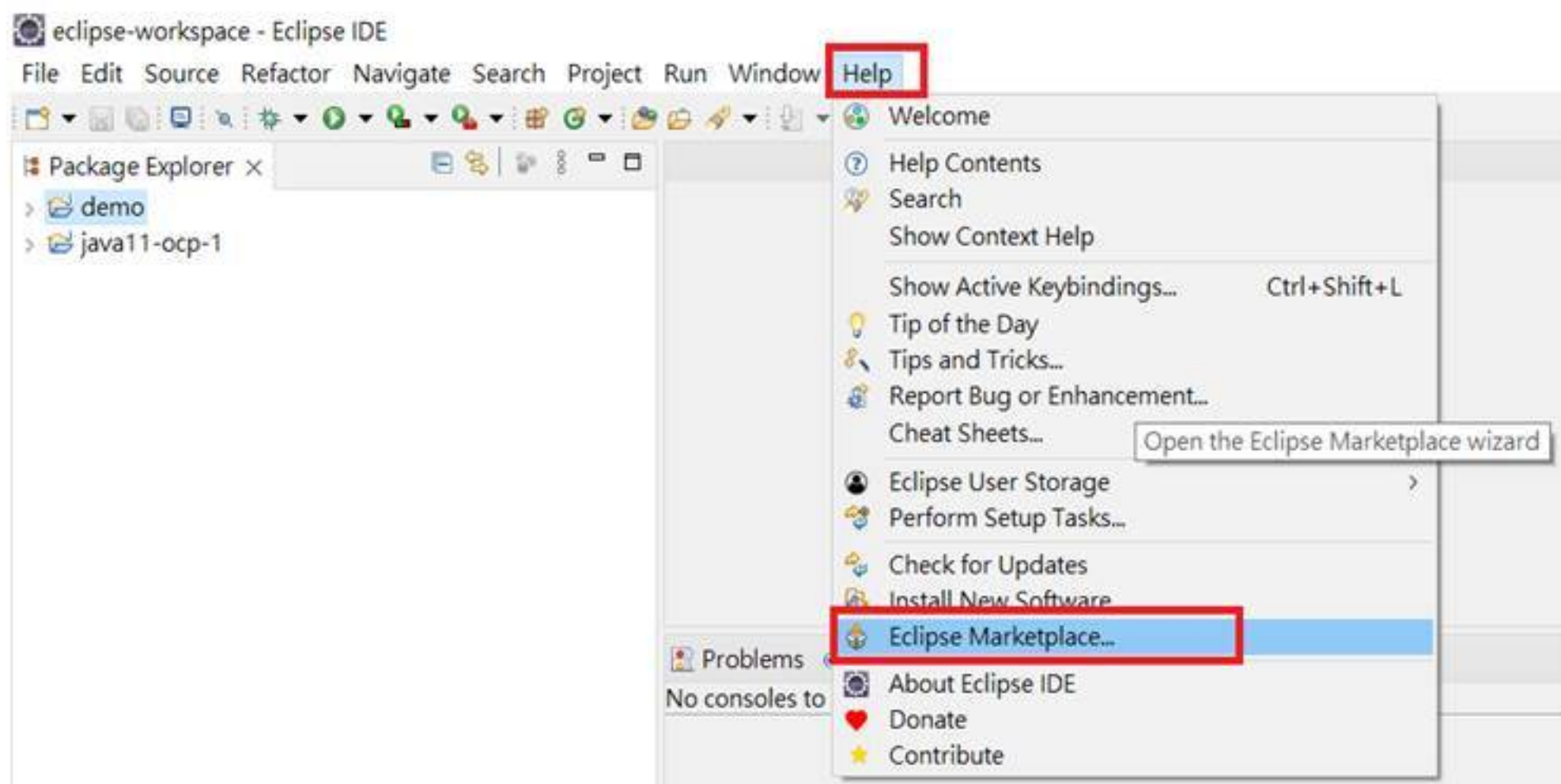
## 使用工具繪製UML





# 【STEP01】

- 繪製UML的工具很多，本書使用umlet，可以在Eclipse擴充後使用。
- 點擊Eclipse的Help頁籤，選擇Eclipse Marketplace：



## 【STEP02】

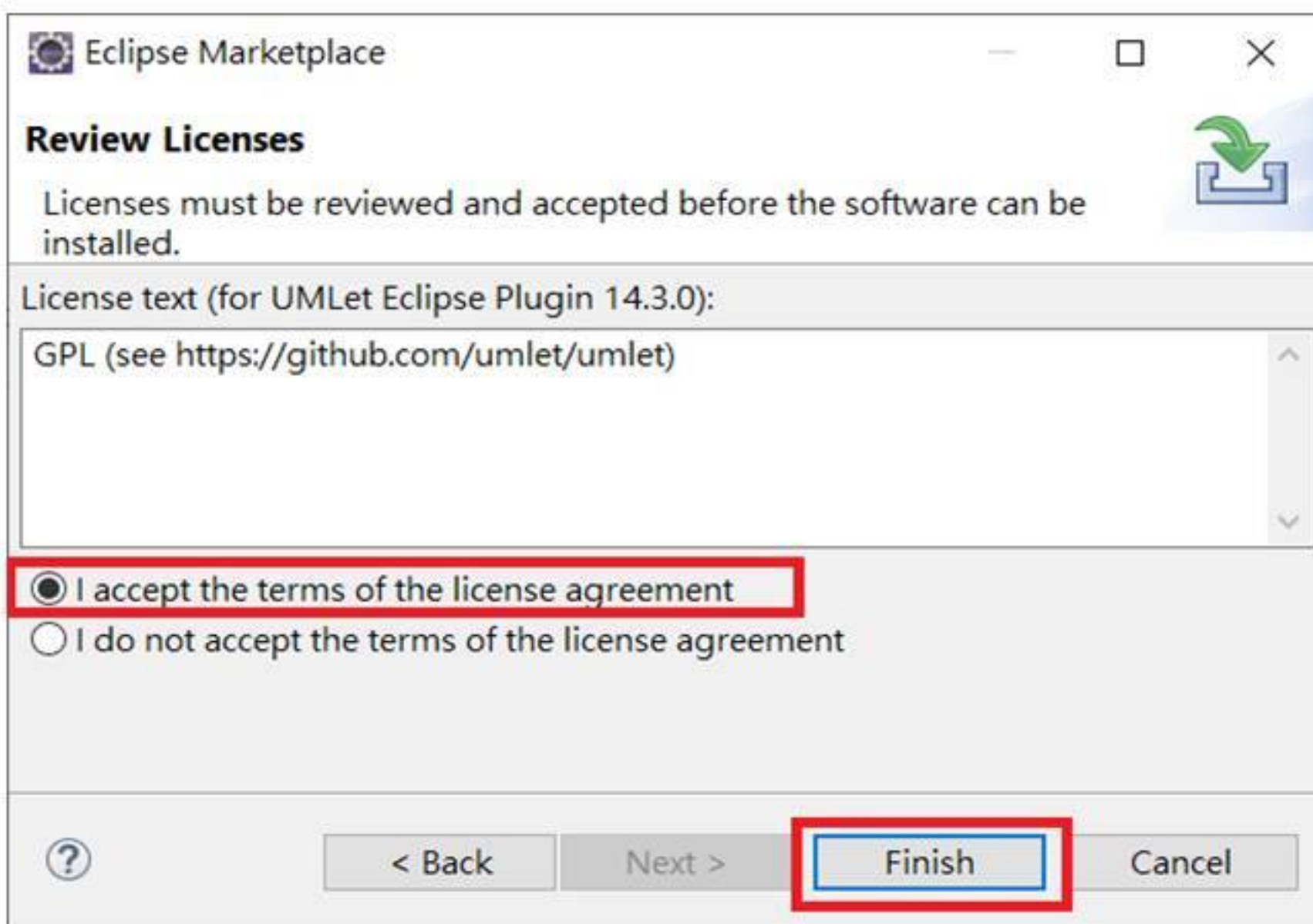
- 鍵入「umlet」，  
點擊Go按鍵。找出解決方案後，  
點擊Install按鍵：



Edited by Ruei-Jiun Tzeng

## 【STEP03】

- 同意授權合約(license agreement)後開始安裝：

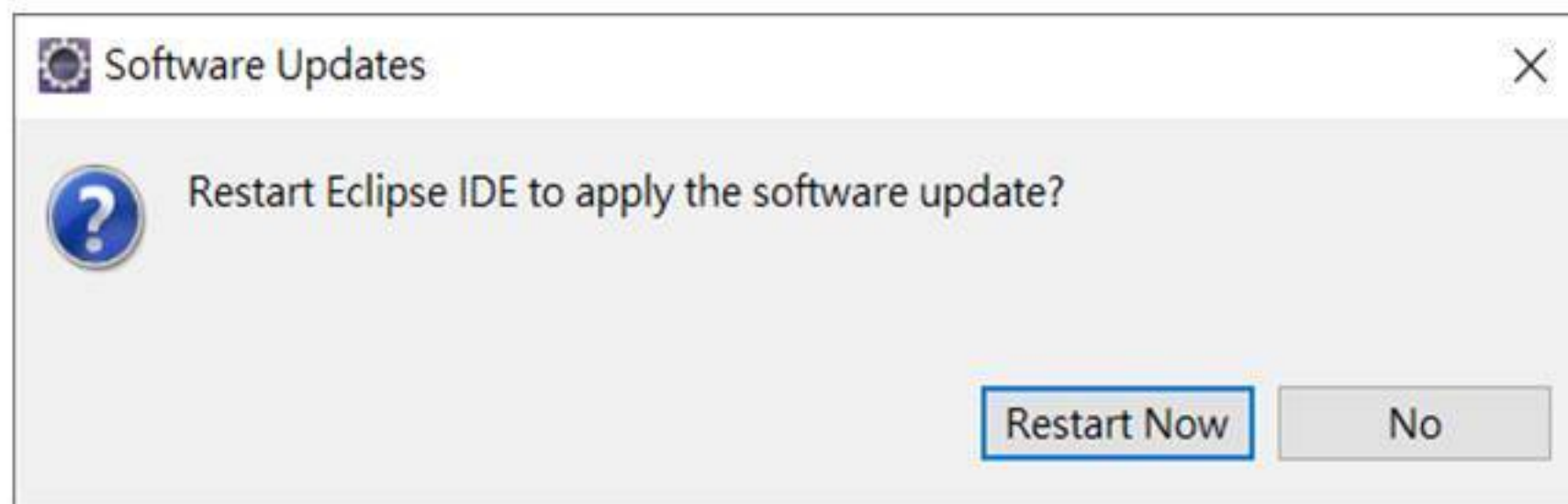


## 【STEP04】 【STEP05】

- 在Eclipse右下角可以看到安裝進度：



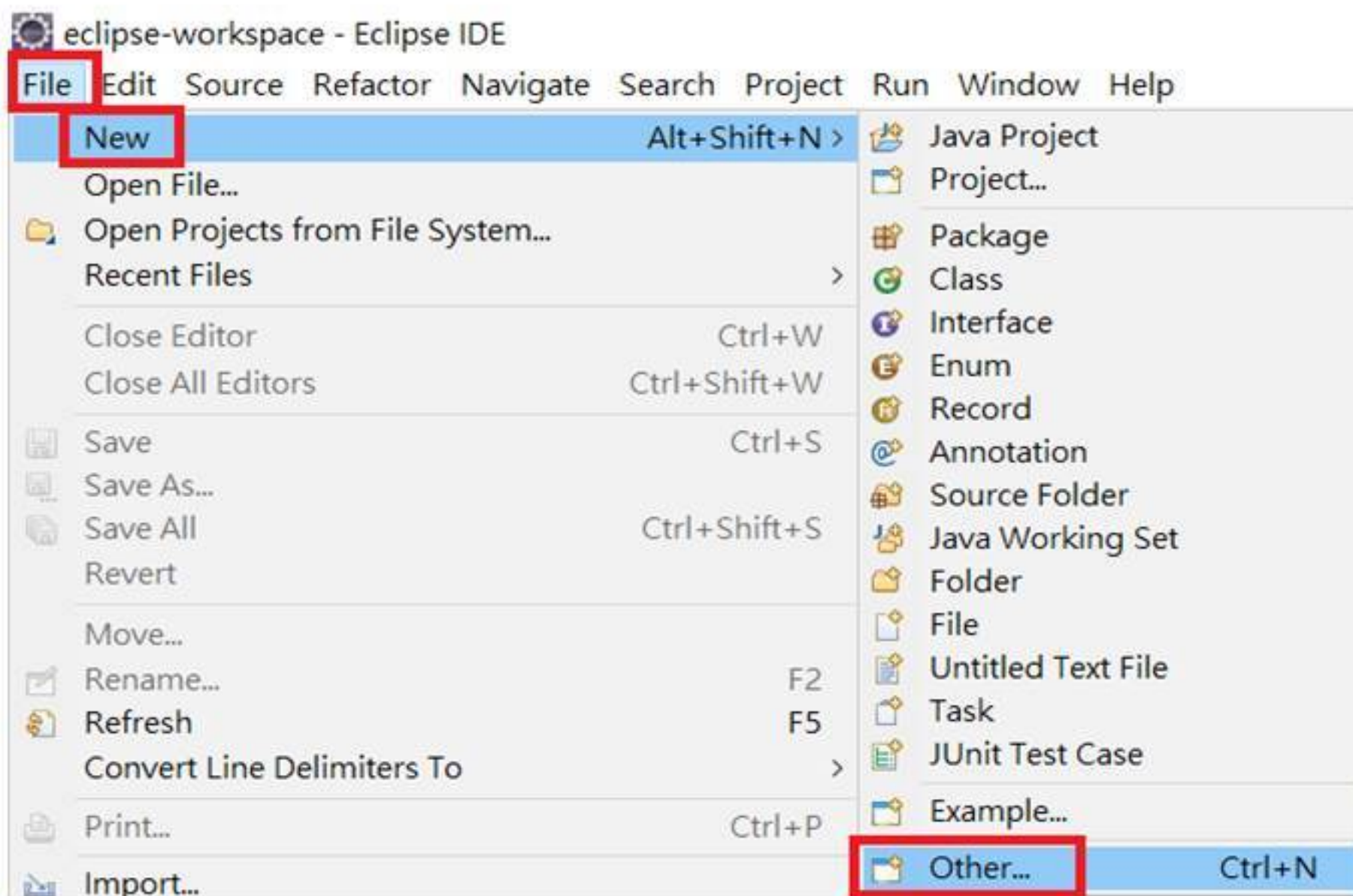
- 重新啟動Eclipse後完成Umlet的安裝：





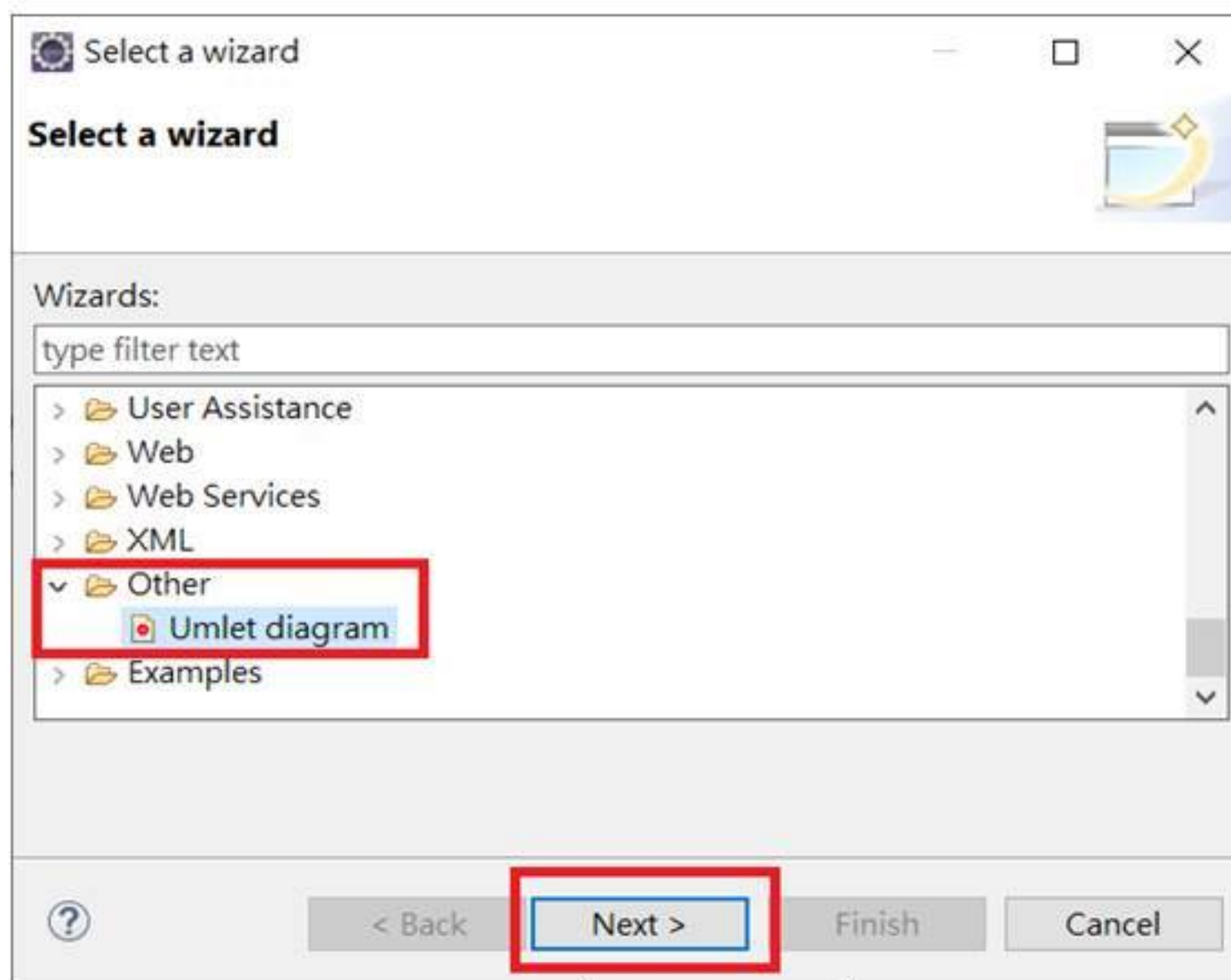
# 【STEP06】

- 點擊Eclipse的File頁籤，選擇New，再選擇Other：



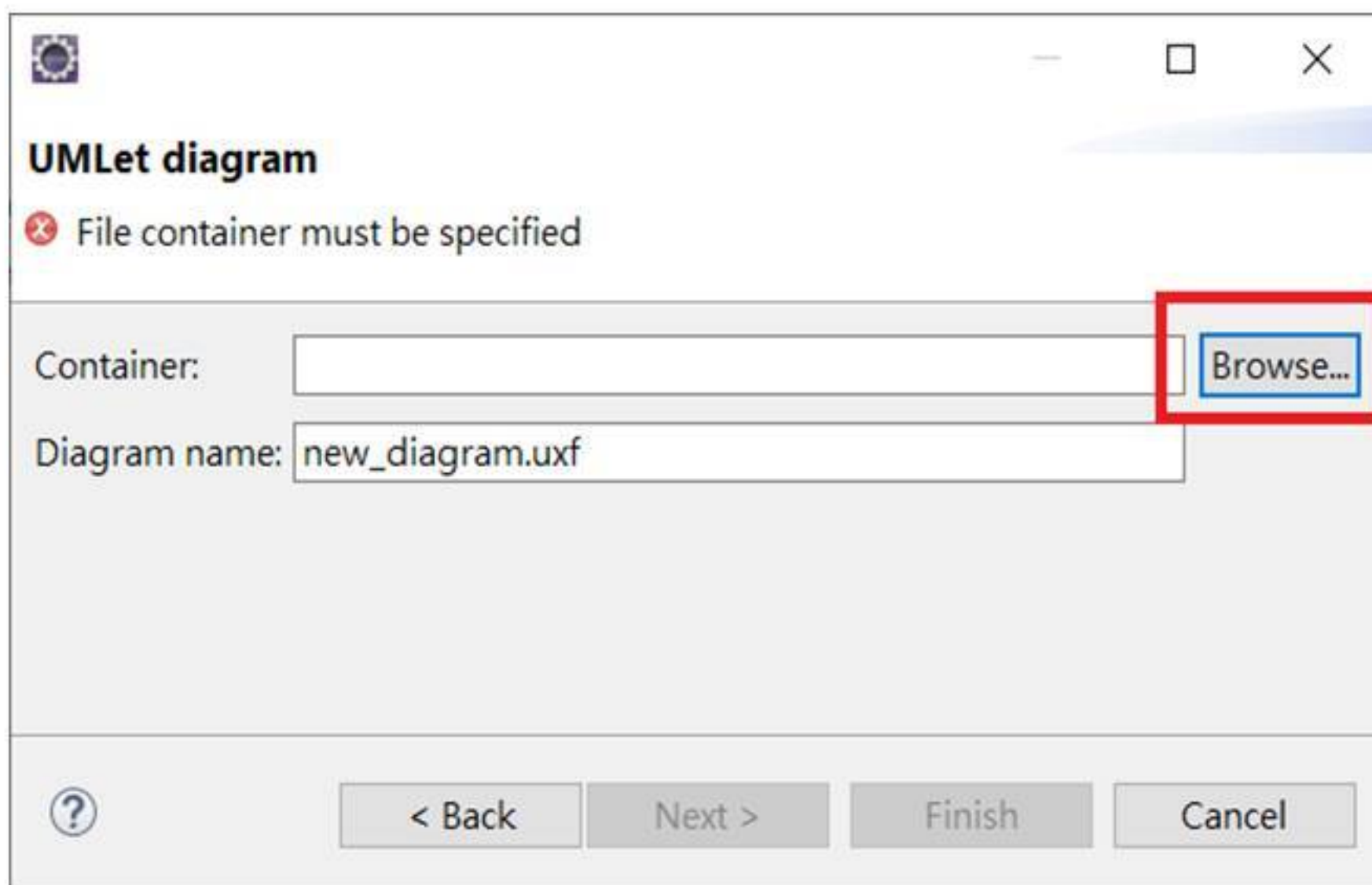
# 【STEP07】

- 點擊Other節點，點擊Umllet diagram：



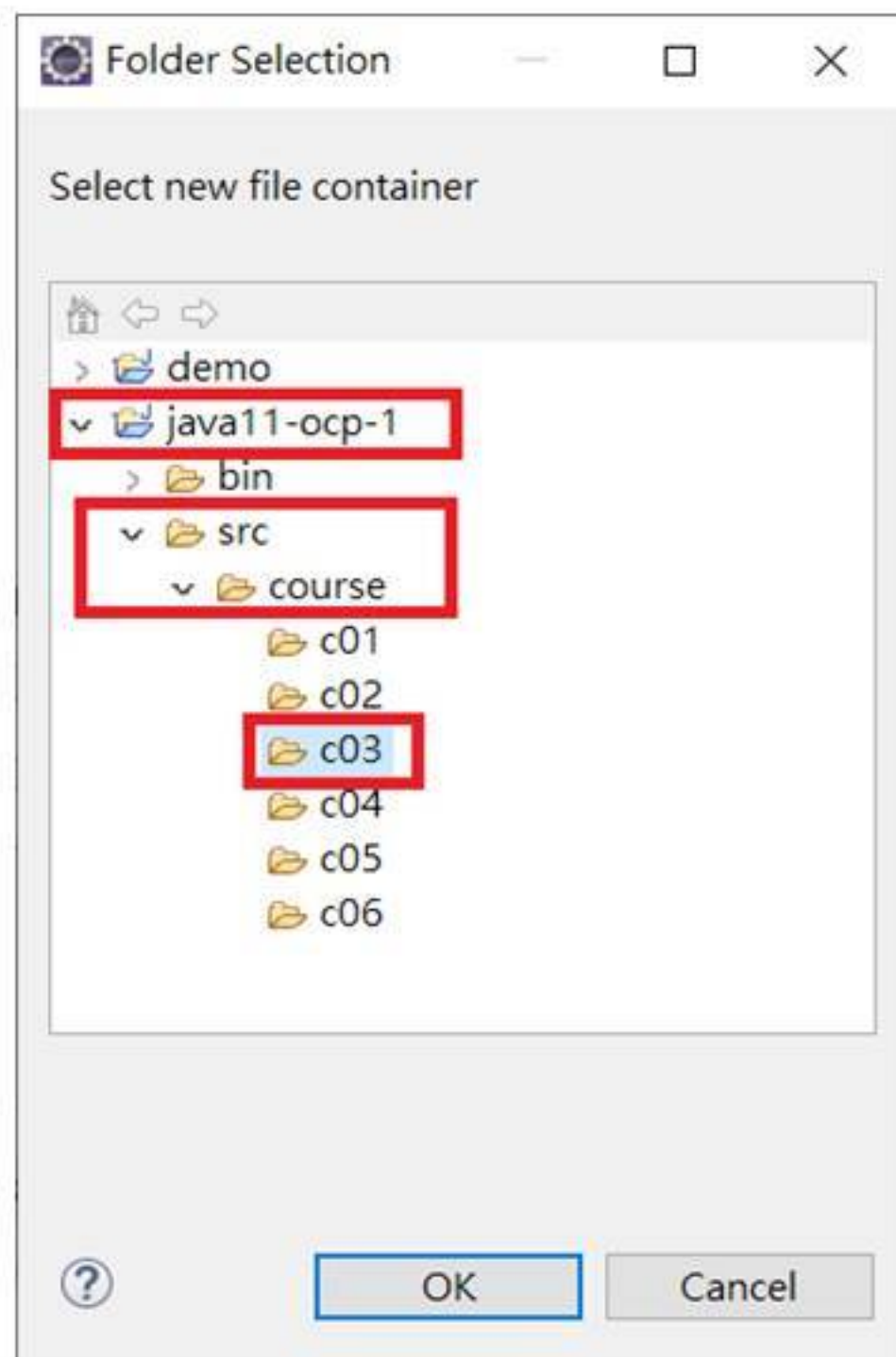
## 【STEP08】

- 點擊Browse按鍵，提供Umlet diagram的預期儲存位置：



## 【STEP09】

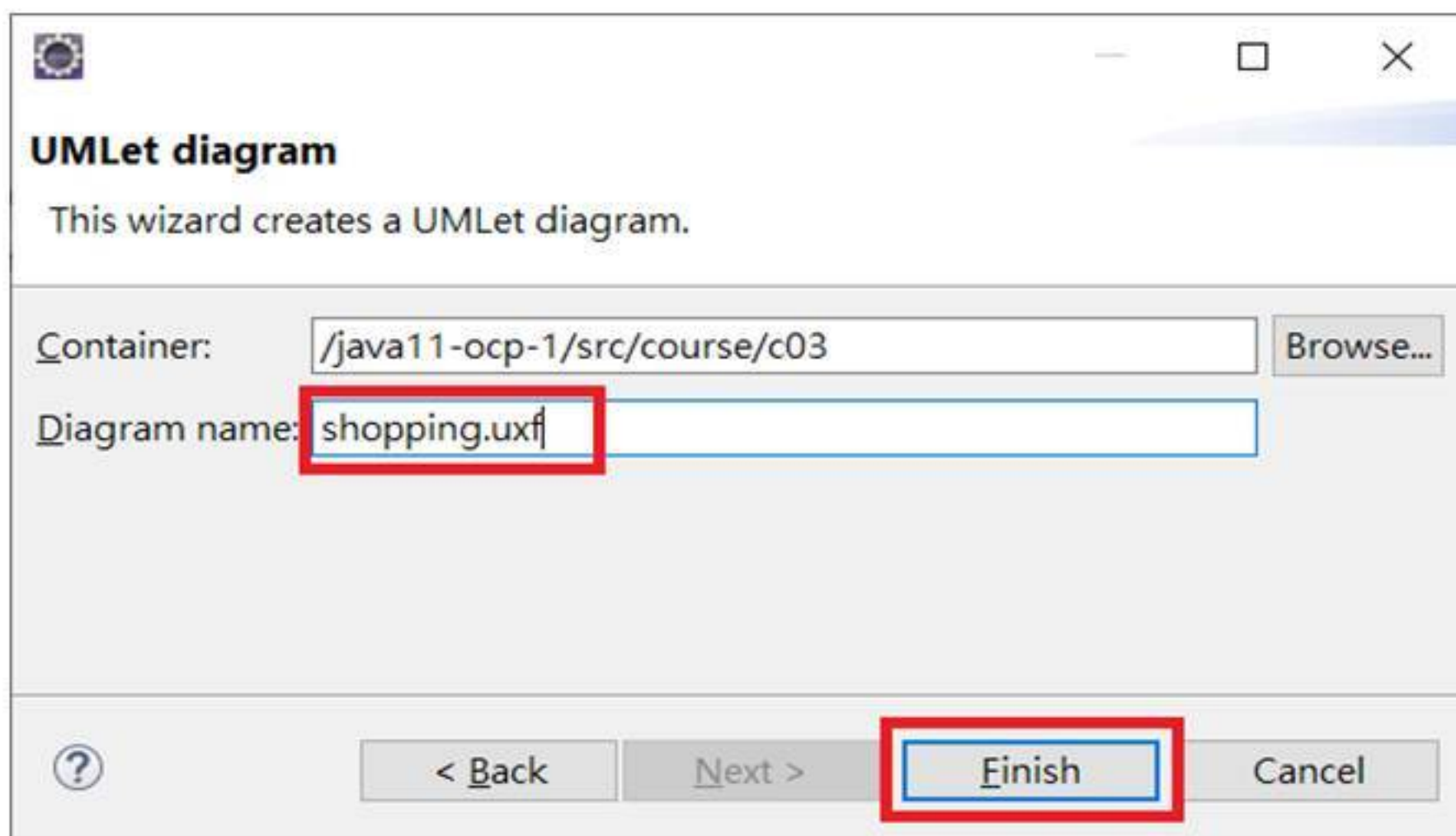
- 點擊要儲存的專案，本例為「java11-ocp-1」；再點選要儲存Umllet diagram檔案的資料夾目錄：





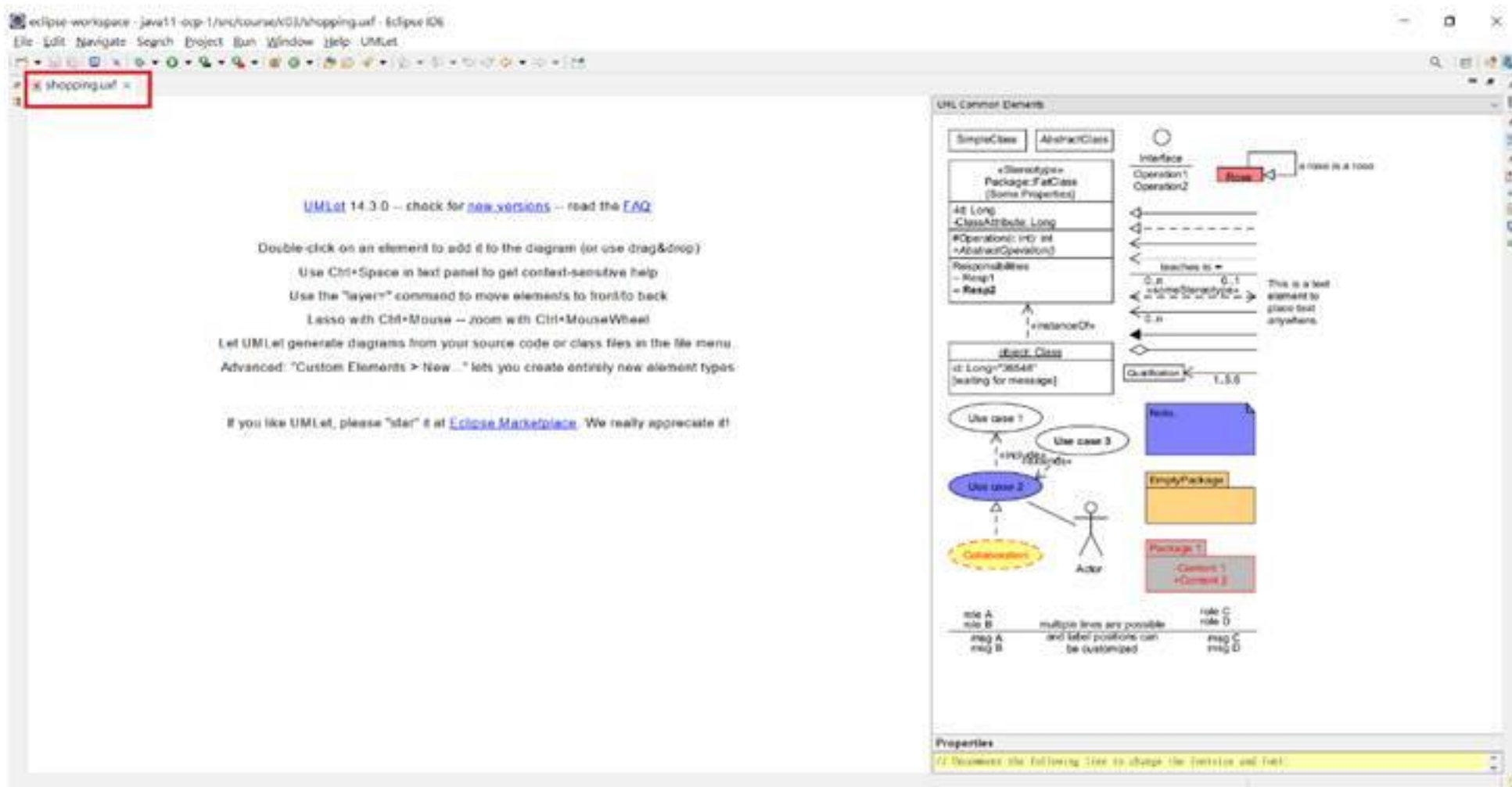
## 【STEP10】

- 輸入Umllet diagram檔案名稱，本例為「shopping.uxf」，再點擊Finish按鍵。完成後可以開啟Umllet diagram檔案並開始編輯：



# 【STEP11】

- 開始編輯Umlet diagram。可以雙擊左上角的「shopping.uxf」頁籤，將編輯區域擴張到最大：

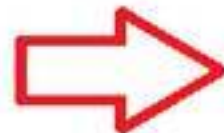
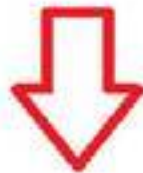


## 【STEP12】

- 開始編輯Umllet。編輯方式是：
  - 在右上方的UML Common Elements窗格裡雙點擊要建立UML元件。
  - 在左側的編輯窗格發現先前在右上方點擊的元件已經複製一份過來。
  - 利用右下方的Properties窗格即可編輯類別圖內容。
  - 如下圖：



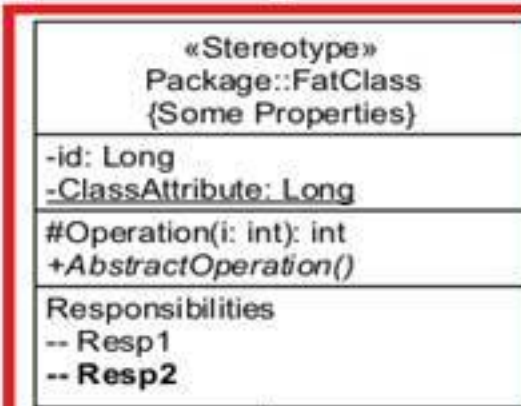
\*shopping.uxf x



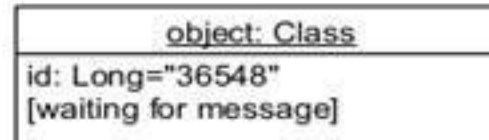
### UML Common Elements

SimpleClass

AbstractClass

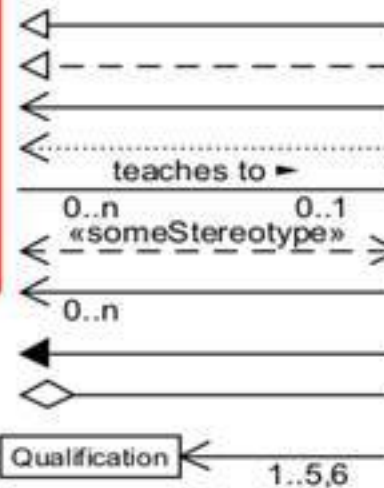
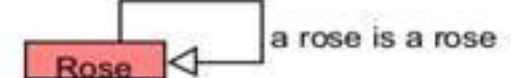


«instanceOf»



Interface

Operation1  
Operation2



This is a text  
element to  
place text  
anywhere.

### Properties

```

<<Stereotype>>
Package::FatClass
{Some Properties}
    
```

```

- id: Long
- ClassAttribute: Long
    
```

```

#Operation(i: int): int
/+AbstractOperation()/
    
```

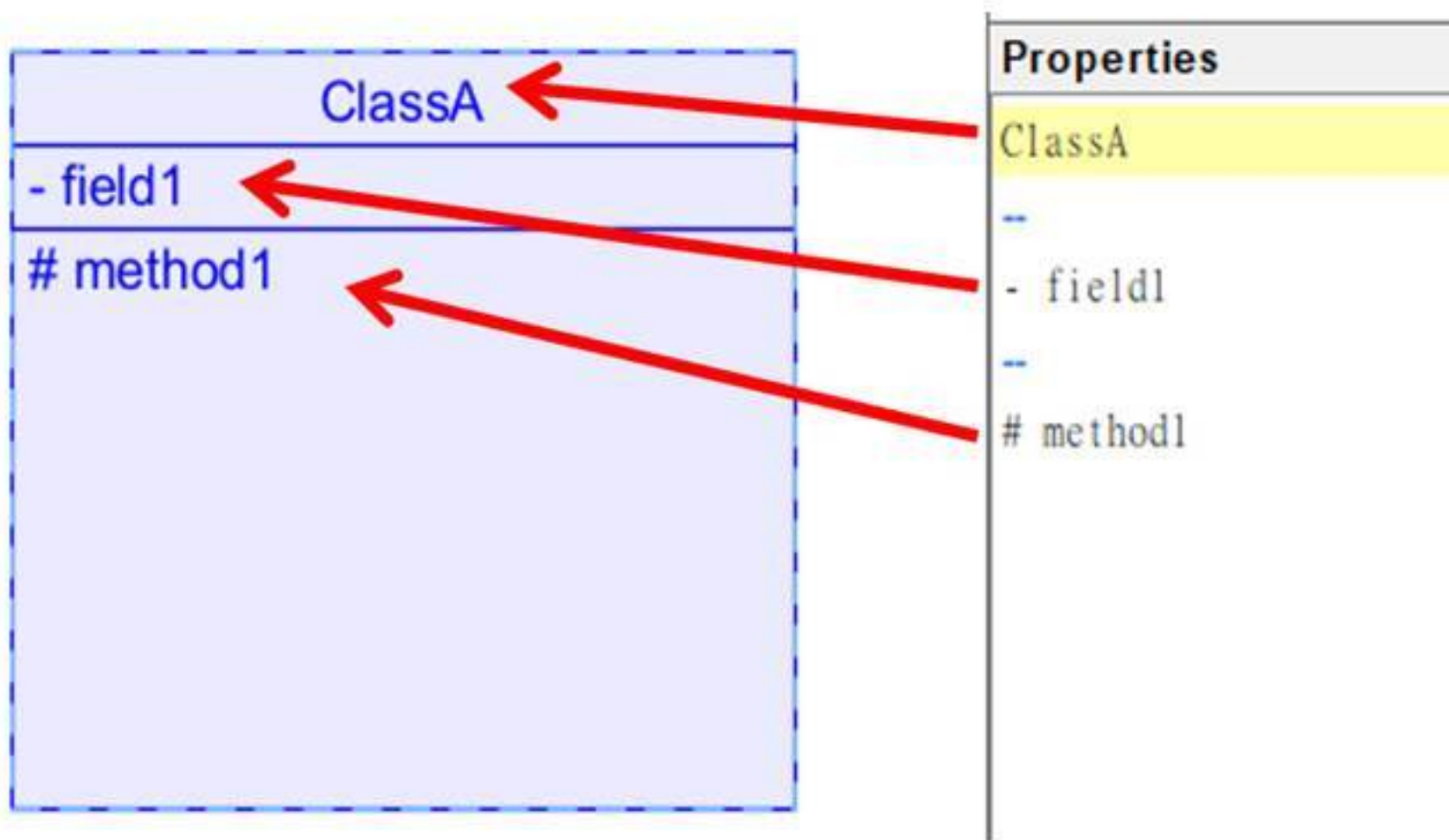
```

Responsibilities
-- Resp1
*-- Resp2*
    
```



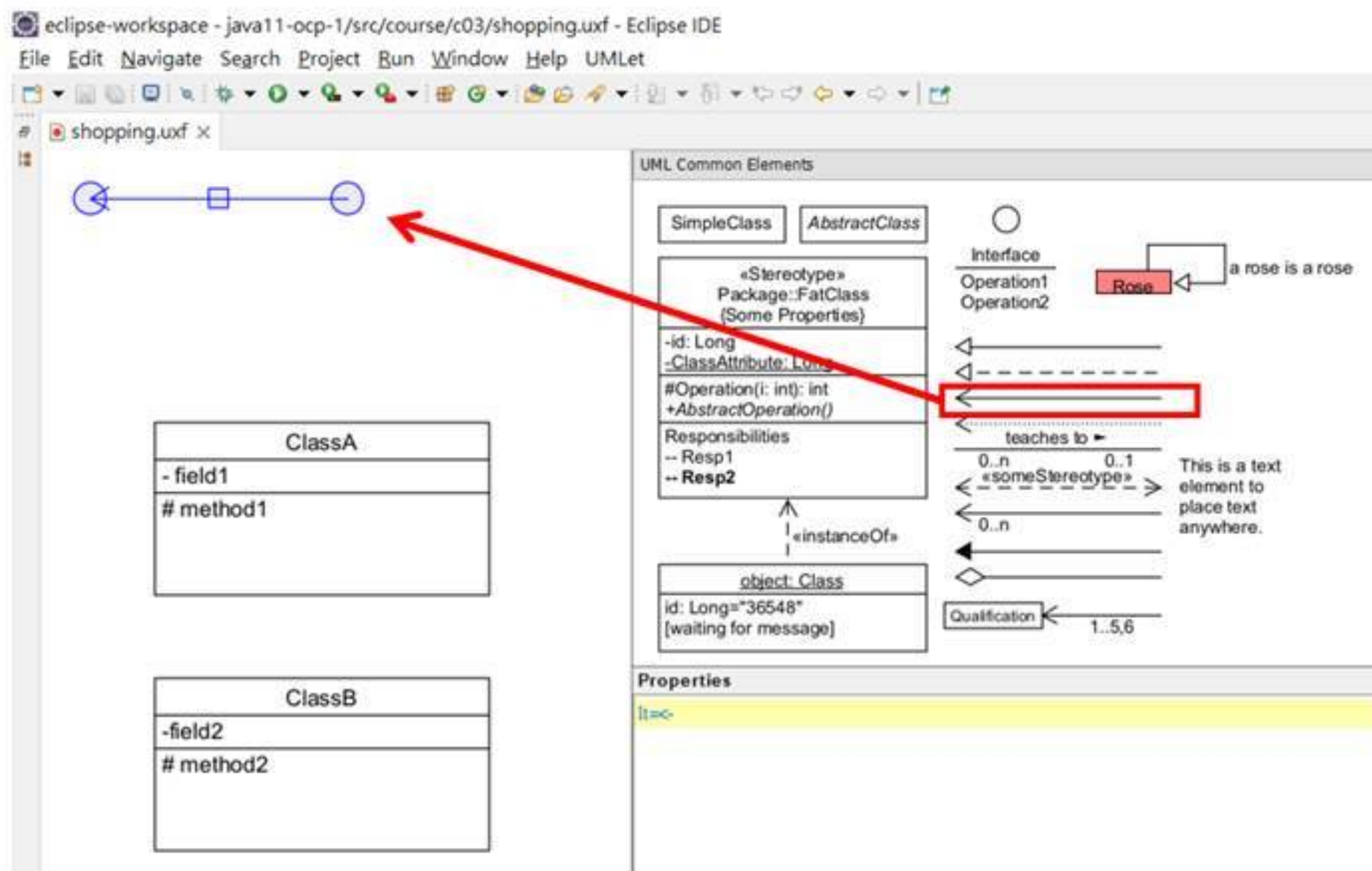
# 【STEP13】

- 建立ClassA的類別圖，依次再建立ClassB：



# 【STEP14】

- 在右上方的UML Common Elements窗格裡雙點擊要建立的線條。



# 【STEP15】

- 將線條調整至定位，即可完成：

eclipse-workspace - java11-ocp-1/src/course/c03/shopping.uxf - Eclipse IDE  
File Edit Navigate Search Project Run Window Help UMLet

\*shopping.uxf x

UML Common Elements

SimpleClass AbstractClass

«Stereotype»  
Package::FatClass  
(Some Properties)

-id: Long  
-ClassAttribute: Long  
#Operation(i: int): int  
+AbstractOperation()

Responsibilities  
- Resp1  
- Resp2

Interface  
Operation1  
Operation2

Rose

a rose is a rose

teaches to

0..n 0..1  
«someStereotype»

This is a text  
element to  
place text  
anywhere.

0..n

Qualification 1..5,6

Properties

```
// Uncomment the following line to change the fontsize and font:  
// Fontsize=14  
// fontfamily=SansSerif //possible: SansSerif,Serif,Monospaced
```



## 【STEP16】

---

- 以鍵盤快速鍵「Ctrl + S」儲存檔案，或點擊Eclipse的File頁籤，再點擊Save按鍵，可以將檔案儲存在硬碟空間。