

# 物件導向程式設計期末專題

## 線上訂餐系統

第18組

智網二甲 A113223002 賴昱婷

智網二甲 A113223006 劉育宗

首先是

系統規格書！



# 線上訂餐系統 (Online Food Ordering System) - 功能規格書

版本：1.0

## 【系統概述】

一個模擬 Uber Eats 或 foodpanda 核心流程的線上訂餐系統，可管理餐廳菜單、餐點客製化、訂單處理、折扣與外送費計算。系統設計重點在於不同餐點類型的計價方式、套餐遞迴計算，以及折扣與準備時間的整合。

## 【類別規格】

### 抽象類別：MenuItem ( 菜單項目 )

#### 屬性：

- name ( 字串 ) : 餐點名稱，不可為空或 null
- basePrice ( double ) : 餐點基礎價格，不可為負數，必須大於 0

#### 行為：

- 建構子 MenuItem(String name, double basePrice)
- 若 name 為 null 或空字串，拋出 IllegalArgumentException
- 訊息："name cannot be null or empty"
- 若 basePrice  $\leq 0$ ，拋出 IllegalArgumentException
- 訊息："basePrice must be greater than 0"
- calculatePrice(): double ( 抽象方法 )
  - 由子類別實作
  - 回傳該餐點計算後的最終價格
- toString()
  - 回傳格式："name - price"

### 類別：MainDish ( 主餐 )

#### 繼承：

- MenuItem ( 菜單項目 )

#### 屬性：

- size ( 字串 ) : 份量選項 ( Small / Medium / Large )

#### 行為：

- 建構子 MainDish(String name, double basePrice, String size)
- 驗證規則同 MenuItem ( 菜單項目 )
- 若 size 非合法選項，拋出 IllegalArgumentException
- 訊息："Invalid size option"
- 不同 size 會影響最終價格

- calculatePrice(): double
  - 根據 size 調整價格後回傳結果
  - Small : 原價
  - Medium : +20
  - Large : +40

### 類別：SideDish ( 附餐 )

#### 繼承：

- MenuItem ( 菜單項目 )

#### 屬性：

- upgraded ( 布林 ) : 是否升級附餐，預設為 false

#### 行為：

- 建構子 SideDish(String name, double basePrice)
- 驗證規則同 MenuItem
- upgrade()
  - 將 upgraded 設為 true
- calculatePrice(): double
  - 若 upgraded 為 false，回傳 basePrice
  - 若 upgraded 為 true，回傳 basePrice 加升級費用 ( +10 )

### 類別：Beverage ( 飲料 )

#### 繼承：

- MenuItem ( 菜單項目 )
- 實作 Customizable 介面

#### 屬性：

- customizations ( ArrayList<String> ) : 冰塊選項 ( 去冰 / 少冰 / 正常冰 )

#### 行為：

- 建構子 Beverage(String name, double basePrice)
- 驗證規則同 MenuItem
- 初始化 customizations 為空清單

- applyCustomization(String option)
  - 將客製化選項加入清單
  - 若 option 為 null 或空字串，拋出 IllegalArgumentException
  - 將 option 加入 customizations

- getCustomizations()
  - 回傳所有客製化選項

- calculatePrice(): double
  - 回傳 basePrice

### 類別：ComboMeal ( 套餐 )

#### 繼承：

- MenuItem ( 菜單項目 )

#### 屬性：

- items ( ArrayList<MenuItem> ) : 套餐內含餐點
- discountRate ( double ) : 套餐折扣率，固定為 0.8

行為：

- 建構子 ComboMeal(String name)
- 初始化 items 為空清單
- 設定 discountRate 為 0.85
  
- addItem(MenuItem item)
  - 若 item 為 null，拋出 IllegalArgumentException
  - 訊息："MenuItem cannot be null"
  - 將 item (餐點) 加入套餐
  
- calculatePrice(): double
  - 加總所有內含 MenuItem 價格 (遞迴計算)
  - 套用套餐折扣 (85 折)
  - 套餐內之餐點不再套用個別折扣
  - 回傳四捨五入後的整數價格

介面：Discountable (可折扣項目)

行為：

- calculateDiscount(DiscountType type): double
  - 回傳折扣金額
  
- isDiscountApplicable(DiscountType type): boolean
  - 判斷是否可套用指定折扣

介面：Customizable (可客製化項目)

行為：

- applyCustomization(String option): void
  
- getCustomizations(): ArrayList<String>
  
- getCustomizationFee(): double

類別：Restaurant (餐廳)

屬性：

- name (字串) : 餐廳名稱
- menu (ArrayList<MenuItem>) : 餐廳菜單項目

行為：

- 建構子 Restaurant(String name)
  - 若 name 為 null 或空字串，拋出 IllegalArgumentException
  - 訊息："Menu item already exists"
  
- addMenuItem(MenuItem item)
  - 若菜單中已有相同名稱餐點，拋出 IllegalArgumentException
  - 訊息："Menu item already exists"
  
- getMenu()
  - 回傳所有菜單項目

類別：Order (訂單)

屬性：

- items (ArrayList<MenuItem>) : 訂單餐點
- customer (Customer) : 下單顧客

行為：

- addItem(MenuItem item)
  - 若 item 為 null，拋出 IllegalArgumentException
  - 將餐點加入訂單
  
- calculateTotalPrice(): double
  - 加總所有餐點價格
  - 訊套用折扣邏輯
  - 回傳最終金額

類別：Customer (顧客)

屬性：

- customerId (字串) : 顧客編號，不可為空或 null
- name (字串) : 顧客姓名，不可為空或 null

行為：

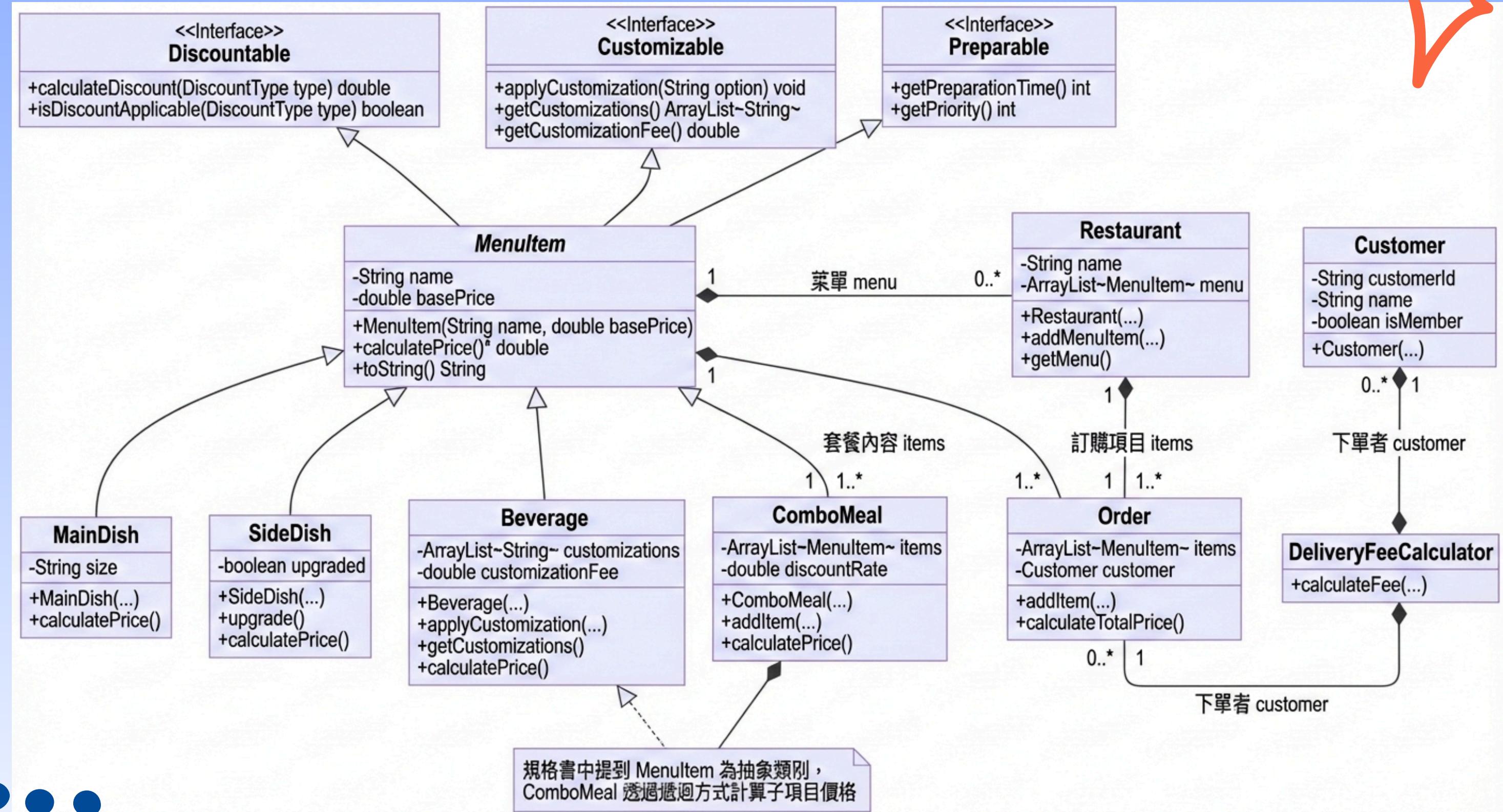
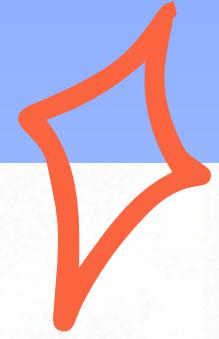
- 建構子 Customer(String customerId, String name)
- 判斷顧客是否符合折扣條件

類別：DeliveryFeeCalculator (外送費計算器)

行為：

- calculateFee(double distance, boolean isPeakTime): double
  - 依距離計算基本外送費
  - 尖峰時段加收額外費用
  - 回傳最終外送費金額

# URL





再來是

測試案例文件！



## 測試案例清單

### 【MenuItem 抽象類別測試】

TC-MI001: 名稱為 null 時拋出例外

輸入 : name=null

預期 : 拋出 IllegalArgumentException , 訊息為 "name cannot be null or empty"

TC-MI002: 名稱為空字串時拋出例外

輸入 : name=""

預期 : 拋出 IllegalArgumentException , 訊息為 "name cannot be null or empty"

TC-MI003: 價格小於 0 時拋出例外

輸入 : basePrice=-1

預期 : 拋出 IllegalArgumentException , 訊息為 "basePrice cannot be negative"

TC-MI004: 正常建立菜單項目

輸入 : name="chickenRice", basePrice=120

預期 : 物件建立成功 , getName() 與 getBasePrice() 回傳正確數值

### 【MainDish 類別測試】

TC-MD001: 指定無效份量選項拋出例外

輸入 : size="XL"

預期 : 拋出 IllegalArgumentException , 訊息為 "Invalid size option"

TC-MD002: Small 份量價格

輸入 : basePrice=100, size="Small"

預期 : calculatePrice() 回傳 100.0

TC-MD003: Medium 份量加價 20

輸入 : basePrice=100, size="Medium"

預期 : calculatePrice() 回傳 120.0

TC-MD004: Large 份量加價 40

輸入 : basePrice=100, size="Large"

預期 : calculatePrice() 回傳 140.0

### 【SideDish 類別測試】

TC-SD001: 未升級附餐價格為基礎價格

輸入 : basePrice=40

預期 : calculatePrice() 回傳 40

TC-SD002: 升級附餐後加價 10

前置 : 呼叫 upgrade()

預期 : calculatePrice() 回傳 50

### 【Beverage 類別測試】

TC-BV001: 客製化增加成功並加入清單

動作 : applyCustomization("去冰")

預期 : customizations.size() 增加 1

TC-BV002: 客製化不影響價格

前置 : 加入多個客製化選項

預期 : getCustomizationFee() 回傳 0.0 , calculatePrice() 等於 basePrice

### 【ComboMeal 類別測試】

TC-CM001: 套餐價格套用 85 折

前置 : 放入一個 100 元與一個 50 元餐點 ( 總計 150 )

預期 : calculatePrice() 四捨五入後回傳 128.0 ( 計算 :  $150 * 0.85 = 127.5$  , Math.round 後為 128)

TC-CM002: 套餐名稱初始化

前置 : name="超值全餐"

預期 : getBasePrice() 為 0.0 ( 依據建構子 super(name, 0) )

### 【Restaurant 類別測試】

TC-RS001: 餐廳名稱為空字串拋出例外

輸入 : name=""

預期 : 拋出 IllegalArgumentException , 訊息為 "Restaurant name cannot be empty"

TC-RS002: 新增重複名稱餐點拋出例外

前置 : 菜單已有 name="cola" 的餐點

動作 : addMenuItem(new Beverage("cola", 30))

預期 : 拋出 IllegalArgumentException , 訊息為 "Menu item already exists"

### 【Order 類別測試】

TC-OR001: 計算訂單總價 ( 多型測試 )

前置 : 加入 MainDish(Large) 140 元 + SideDish(Upgraded) 50 元

預期 : calculateTotalPrice() 回傳 190.0

### 【Customer 顧客類別測試】

TC-CS001: 顧客編號為空字串拋出例外

輸入 : customerId=""

預期 : 拋出 IllegalArgumentException

TC-CS002: 顧客姓名為空字串拋出例外

輸入 : name=""

預期 : 拋出 IllegalArgumentException

TC-CS003: 正常建立顧客物件

輸入 : customerId="C001", name="Mary"

預期 : 物件建立成功

### 【DeliveryFeeCalculator 外送費測試】

TC-DF001: 非尖峰時段外送費計算

輸入 : distance=5, isPeakTime=false

預期 : 回傳 50.0 ( 距離 \* 10 )

TC-DF002: 尖峰時段外送費加成

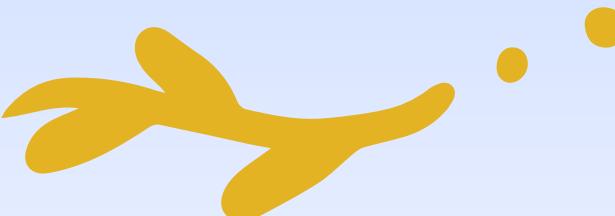
輸入 : distance=5, isPeakTime=true

預期 : 回傳 70.0 ( 50 + 20 )



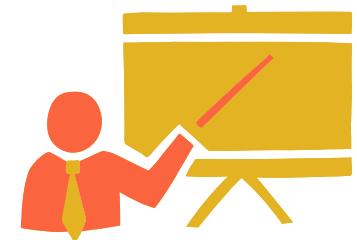
接著是

與AI溝通！



# 如何下

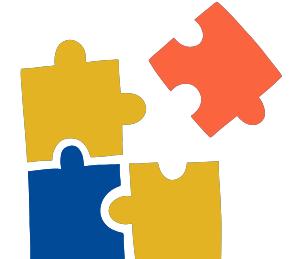
## 提示詞



### 明確指令



### 提供細節



### 逐步指示