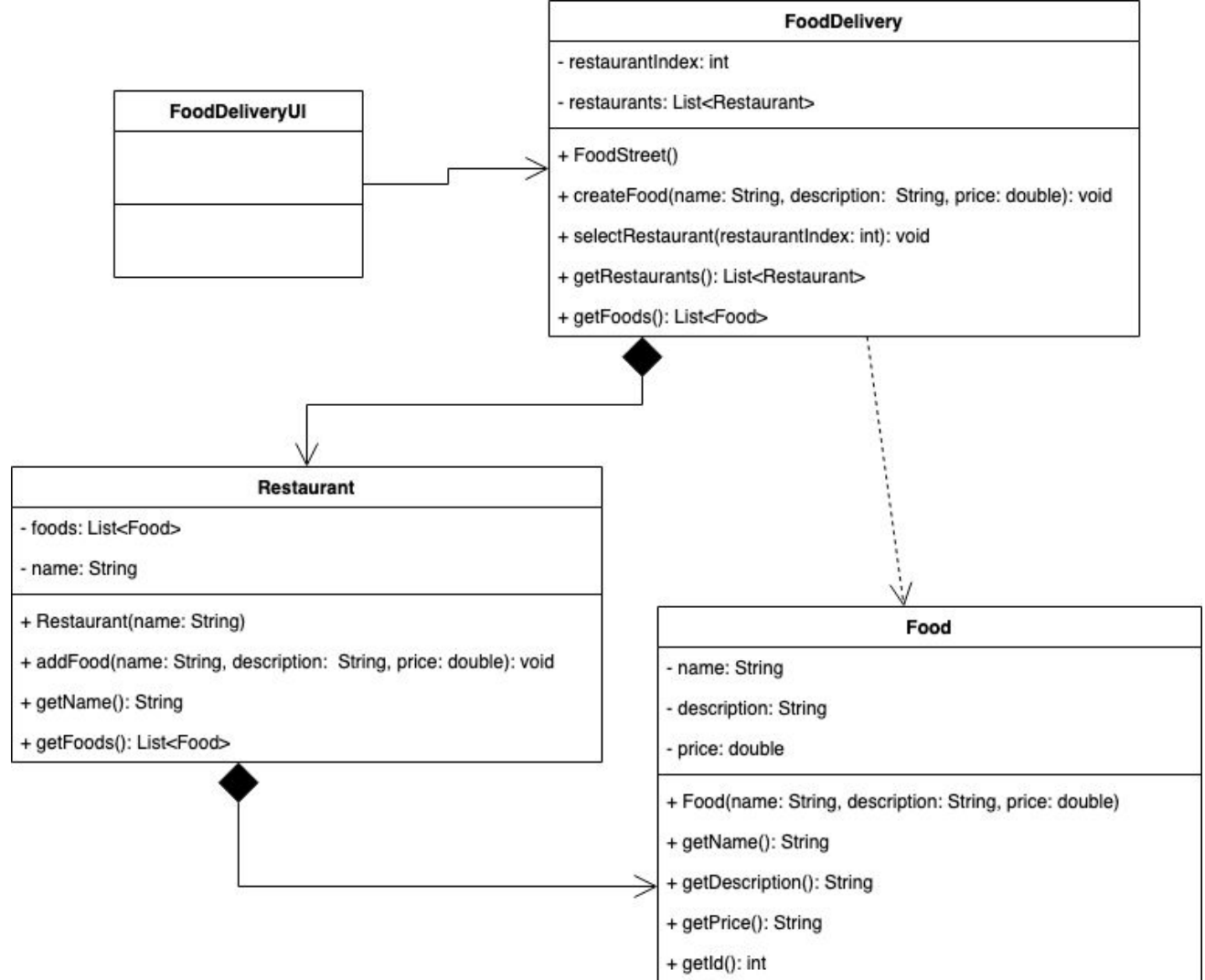


# Java Development Homework 5

Due before May 10 23:59

# Class Diagram



# 作業說明

1. 未來作業會持續擴充, **請確保程式的維護性**。
2. 這次作業要設計一個簡易的外送平台系統, 需要有幫餐廳新增食物、查看所有食物等的功能。系統有四個 Class, 分別為FoodDeliveryUI、FoodDelivery、Restaurant、Food。

FoodDeliveryUI: 因為使用者介面直接寫在 main 並不是很好的設計, 所以最好是將所有的 UI 功能封裝到一個 Class 上面, 稱為 FoodDeliveryUI, 較為整潔。

FoodDelivery: 用來管理多個 Restaurant, 封裝管理 Restaurant 的邏輯。後續作業 FoodDelivery 會進一步增加管理多個 Customer (目前還沒) 跟多個 Restaurant, 所以請確實使用 FoodDelivery 來封裝管理的邏輯, 讓之後方便擴充。

createFood(): 使用 Restaurant 來呼叫 addFood()

selectRestaurantIndex(): 儲存目前在哪個餐廳

Restaurant: 有三間, 分別為"Starbucks"、"Mcdonald's"、"KFC"。一個 Restaurant 有多個 Food。

addFood(): 新增 Food object 在 List

Food: 可以由使用者介面在 Restaurant 新增 Food, 屬性分別有id、名稱、描述、價格。

3. 請依照上面的 Class Diagram 進行撰寫, 上述的 public operation 必須要相同, private operation 可以依照自己的設計定義。

主程式如下：

```
public class Main {  
    public static void main(String[] args) {  
        FoodDelivery foodDelivery = new FoodDelivery();  
        FoodDeliveryUI ui = new FoodDeliveryUI(foodDelivery);  
        ui.start();  
    }  
}
```

請實作以下功能：

1. 當執行程式，會先顯示多個 Restaurant 的名稱(商店名稱請設定為跟以下圖片一樣)。

```
1. Starbucks  
2. Mcdonald's  
3. KFC  
Select Restaurant:
```

2. 使用者輸入想要進入的餐廳 Index, 系統顯示餐廳介面。例如以下兩張圖。

1. Starbucks

2. McDonald's

3. KFC

Select Restaurant: 1

(Starbucks)1. 新增食物

(Starbucks)2. 查看所有食物

(Starbucks)3. 返回餐廳選擇

請輸入選擇: |

1. Starbucks

2. McDonald's

3. KFC

Select Restaurant: 2

(McDonald's)1. 新增食物

(McDonald's)2. 查看所有食物

(McDonald's)3. 返回餐廳選擇

請輸入選擇:

3. 使用者進入餐廳後，輸入返回指令 (3)，系統會返回並顯示餐廳選擇介面。

```
1. Starbucks
2. McDonald's
3. KFC
Select Restaurant: 1
(Starbucks)1. 新增食物
(Starbucks)2. 查看所有食物
(Starbucks)3. 返回餐廳選擇
請輸入選擇: 3
1. Starbucks
2. McDonald's
3. KFC
Select Restaurant: |
```

4. 使用者可以新增食物到目前餐廳裡。當 查看所有食物時，會顯示新增的食物。目前食物的名稱重複沒有關係。

- id為所有食物新增的順序，不分店家
- 當沒有食物，請直接印 "No. Name Price Description"
- print format 請使用 "%-8d %-16s %-8.2f %-8s\n"

請輸入選擇： 1

請輸入食物名稱： Fish burger

請輸入食物描述： a burger with fish

請輸入食物價格： 90

請輸入選擇： 1

請輸入食物名稱： Chicken burger

請輸入食物描述： a burger with chicken

請輸入食物價格： 100

請輸入選擇： 2

No.	Name	Price	Description
1	Fish burger	90.00	a burger with fish
2	Chicken burger	100.00	a burger with chicken

請輸入選擇： |

5. 使用者輸入範圍外的餐廳時，需要要求使用者重新輸入直到正確的範圍為止

1. Starbucks

2. McDonald's

3. KFC

Select Restaurant: 4

請選擇畫面上的餐廳

Select Restaurant: 5

請選擇畫面上的餐廳

Select Restaurant: 6

請選擇畫面上的餐廳

Select Restaurant: 1

(Starbucks)1. 新增食物

(Starbucks)2. 查看所有食物

(Starbucks)3. 返回餐廳選擇

請輸入選擇：



6. 使用者輸入的價格小於 0 或等於 0 時, 需要要求使用者重新輸入直到正確的範圍為止

(McDonald's)1. 新增食物

(McDonald's)2. 查看所有食物

(McDonald's)3. 返回餐廳選擇

請輸入選擇: 1

請輸入食物名稱: Fish burger

請輸入食物描述: a burger with fish

請輸入食物價格: -5

價錢必須大於零, 請重新輸入。

請輸入食物價格: 0

價錢必須大於零, 請重新輸入。

請輸入食物價格: 15

請輸入選擇:

7. 當使用者在目前商店輸入非正常的選擇，系統會要求使用者重新輸入，直到正確的選項範圍為止。

1. Starbucks

2. McDonald's

3. KFC

Select Restaurant: 3

(KFC)1. 新增食物

(KFC)2. 查看所有食物

(KFC)3. 返回餐廳選擇

請輸入選擇: 5

請輸入正確選項: 4

請輸入正確選項: 100

請輸入正確選項: 2

No.	Name	Price	Description
-----	------	-------	-------------

請輸入選擇:

8. 在選擇餐廳的介面輸入 -1 結束程式

```
1. Starbucks
```

```
2. Mcdonald's
```

```
3. KFC
```

```
Select Restaurant: -1
```

```
Process finished with exit code 0
```

### 注意事項：

1. 一定要按照 class diagram 撰寫程式, 否則會視情況扣分
2. 為了強調 UI 與 Model 的分離, 請不要在 UI 以外的 class (ex. FoodDelivery, Restaurant, Food) 使用 print

# Submission

Please archive your source code to STUDENT\_ID.zip (download the example zip file from Moodle) and **upload to Moodle** before deadline.

Your zip file should follow the following format.

STUDENT\_ID.zip

|- src

|- METS-INF

| |- MSNIFEST.MF

All the source files (\*.java) are put in the src directory.

The entry point (i.e. main class) of the program is specified in the MSNIFEST.MF file.

No late submission is accepted.