**需求描述:**

1. 使用者在使用此程式時可能會有何需求？(1和2的問題是對應的)

使用管理模式:

a. 安排顧客座位

b. 查看店內狀況

c. 查看廚師狀況

d. 模擬經過時間

e. 結束系統

使用點餐模式:

a. 點餐

b. 結束點餐

1. 設計程式時如何考慮這些需求？

使用管理模式:

1. 利用scanner和substring讀取並判斷該座位是否有人，沒有人的話

就把該座位給新來的人，並把flag設為1;有人的話，則印出”該座位有人” ，再把mode設為1，跳至點餐模式

1. 檢查4個座位的flag，如果flag為1代表有人，則利用for迴圈印出

他們未上桌和已上桌的餐點

1. 利用for迴圈印出兩個廚師分別正在準備的餐點和剩餘要準備的餐點
2. 利用scanner和substring讀取經過時間，然後分別對廚師餐點的準備和顧客餐點的食用進行模擬

e. 直接break跳出while迴圈(點餐系統是利用while迴圈不斷在裡面跑)

使用點餐模式:

a. 利用scanner和substring讀取餐點和指派的廚師，還有點餐的桌號

b. 結束點餐，將mode設為0，跳回管理模式

1. 程式中有哪些地方特別吸引使用者？

a. 能夠輸入簡單指令為顧客點餐以及安排廚師們的工作分配

b. 隨時可以了解餐點準備狀況和店內狀況

c. 可以模擬廚師餐點的準備和顧客餐點的食用

**程式流程:**

next

add

status

chef

status

table

new

end

exit

mode=1

mode=0

點餐模式

start

點餐系統

管理模式

安排顧客位子

mode=1

查看店內狀況

查看廚師狀況

模擬經過時間

結束系統

end

點餐

結束點餐

mode=0

**功能/邏輯說明:**

1. 點餐系統是利用while迴圈，讓程式不斷的跑，直到break
2. 不管在管理模式或點餐模式，只要做完一件事，就會跑回while迴圈的一開始，然後利用mode這個flag來判斷要進入管理模式還是點餐模式
3. 因為有2個廚師和4個位子，我創建了廚師的class和位子的class，然後創建object

廚師的class: 有meal、seat、time、oktime的arraylist，分別儲存餐點名稱、點餐的桌號、需要製作的時間、完成餐點的時間，printmeal function是用來印出餐點名稱，addtime function是用來把需要製作的時間加入time的arraylist，work function是在模擬廚師製作餐點

位子的class: 有meal、done、eat、time、oktime的arraylist，分別儲存餐點名稱、是否已上桌、是否吃完、需要食用的時間、完成餐點的時間，printmeal function是用來印出餐點名稱，addtime function是用來把需要食用的時間加入time的arraylist，eat function是在模擬顧客食用餐點

1. 模擬廚師製作餐點的work function是先確認廚師有餐點未完成且還有時間，判斷正在製作的餐點是否可以同時準備多份，再根據剩餘的時間與餐點製作所需時間比較，判斷是否完成該餐點，直到剩餘時間為0或無餐點可以製作
2. 模擬顧客食用餐點的eat function是先確認顧客餐點已上桌且未食用完，判斷最早完成的餐點(oktime arraylist)當作要食用的餐點，再根據現在時間和完成此餐點的時間的關係以及此餐點需要食用的時間和剩餘時間的關係，判斷是否食用完該餐點，直到剩餘時間為0或無餐點可以食用

**使用說明:**

1.javac hw3.java (編譯)

2.java hw3 (執行)

程式剛開始執行是管理模式

管理模式:

1.輸入new [桌位編號] (ex:new 1)，可以安排顧客座位，並跳至點餐模式

2.輸入status table，可以查看店內狀況，呈現哪些桌位有人、個別桌位上已上桌和剩餘未上桌的餐點有哪些

3.輸入status chef，可以查看廚師狀況，呈現個別廚師正在準備的餐點和剩餘要準備的餐點有哪些

4.輸入next [時間（分）] (ex:next 5)，可以模擬經過時間

5.輸入exit，可以結束系統

點餐模式:

1.輸入add [餐點編號] [廚師編號] (ex:add M2 S1)，可以新增餐點

2.輸入end，可以結束點餐模式，回到管理模式