|  |
| --- |
| 教師：曾智義 課程名稱：基礎程式設計 學期：112-1 |
| 一、第07週課堂練習 |
| 綜合運用至目前所學(基本運算、變數、字串、List、Dict、Tuple、Set、If、For、While、Def） |
| 二、個人/成員 |
| 林彧頎\_資工系\_A1105505(由於詢問過老師可以個人交，因此我們組是交個人作業) |
| 三、議題規劃 |
| 市面上有許多種的訂餐系統和額外的訂位系統，但卻沒有一個整合的訂餐與訂位系統，為此我想設計一款訂餐與訂位系統可以透過訂餐人員輸入相關資訊，並再由服務人員依照訂單需求進行座位劃位，讓使用者可以使用統合的方法於一個軟體中執行出完善的作業模式!  而本系統也針對使用者有一項額外的規劃，為了使用者的健康，使用者可以透過輸入自身的身高、體重，提供飲食建議，並推薦更符合他的健康飲食，不僅僅達到商業廣告功能，更可以使消費者的飲食健康提升。  針對員工端，本系統也新增員工資料管理，並建立簽到系統草圖，可對於未來規劃資料匯出的整合~ |
| 四、問題定義  (自行設計問題，問題可列舉或敘述方式，自由模擬，以下為例子) |
| 1. 需要透過資料輸入訂餐與訂位系統中所需的相關資料，包含:訂餐者、人員數、菜品、菜品數量、用餐時間。 2. 將上述的內容計算出總金額，與結果顯示給使用者，並模擬使用者付款。 3. 於服務人員端，可以透過上述的資訊，自動化的進行桌分配(人數過多時，需要幾個桌位) 4. 訂餐系統針對list、dictionary增加套餐功能(增加餐廳行銷與使用者選擇) 5. 使用者輸入自身身體數值(此針對List、Dict、Tuple、Set)，提供餐廳飲食建議 6. 針對服務人員端新增員工資料管理與簽到系統(未來可以輸出檔案) 7. 過程中需使用(基本運算、變數、字串、List、Dict、Tuple、Set、If、For、While、Def)進行邏輯判斷等等… |
| 五、程式碼設計  (將執行的程式碼貼入並註解，綜合運用所學，以下為例子) |
| 1. import math  2. import datetime  3. print('以下是一個訂餐訂位系統請依照指示執行!\n')  4. print('------------------------------------\n')  5. #使用者輸入相關資訊  6. print('welcome 日日香~\n')  7. name=input('請問您的大名:\n')  8. print('歡迎您~'+name+'\n')  9. people\_num=int(input('請問'+name+'會有幾位用餐呢?\n'))  10. test=True  11. #新增使用者輸入自身身體數值，提供餐廳飲食建議  12. print('-----------------------------------\n')  13. print('本系統可以提供您:輸入自身身體數值，提供餐廳飲食建議\n')  14. diet=int(input('請問需要測試嗎(要(1)/不要(0))?\n'))  15. user={}  16. #使用def定義飲食建議功能  17. def diet\_recommand():  18. p\_num=1  19. for i in range(people\_num):  20. #建tuple關於個人資料建立(name,birthday,date)  21. print('請輸入第'+str(p\_num)+'位的姓名、生日')  22. n=input('姓名')  23. b=input('生日:yyyy/mm/dd(請依照此格式)')  24. current\_time = datetime.datetime.now()  25. time\_now = current\_time.strftime('%Y/%m/%d')  26. key\_t=(n,b,time\_now)  27. print('請輸入第'+str(p\_num)+'位的體重(kg)與身高(m)\n')  28. p\_num=p\_num+1  29. weight=float(input('體重(kg)\n'))  30. height=float(input('身高(m)\n'))  31. bmi=float(weight/(height)\*\*2)  32. #將數據儲存於list  33. value\_l=['體重(kg)',weight,'身高(m)',height,'BMI',bmi]  34. #儲存到dict(key: tuple / value:list)  35. user[key\_t]=value\_l  36. #進行if else判斷  37. if(bmi<=18.5):  38. print('您的bmi為: '+str(bmi)+' 這表示您的體重過輕\n')  39. print('您可以任意點選我們餐廳的食物\n推薦您我們的招牌鍋貼喔~\n')  40. elif(bmi>=18.5 and bmi<24.0):  41. print('您的bmi為: '+str(bmi)+' 這表示您的體重正常\n')  42. print('您平時的飲食維持得不錯\n我推薦您可以搭配水餃與菜頭湯\n飲養又均衡~\n')  43. elif(bmi>=24.0):  44. print('您的bmi為: '+str(bmi)+' 這表示您的體重過重\n')  45. print('我不建議您喝紅茶，甜度太高，可以嘗試我們的無糖豆漿喔~\n')  46. #當問題得到1的回答，則執行飲食建議功能  47. if(diet==1):  48. diet\_recommand()  49. print('------------------------------------------------------------------------')  50. #新增套餐(list+dict應用)  51. m\_set1=[70,'鍋貼','紅茶']  52. m\_set2=[80,'水餃','紅茶']  53. menu={'鍋貼':50,'水餃':60,'菜頭湯':35,'紅茶':25,'豆漿':25,'套餐1':m\_set1,'套餐2':m\_set2}  54. order={'鍋貼':0,'水餃':0,'菜頭湯':0,'紅茶':0,'豆漿':0,'套餐1':0,'套餐2':0}  55. #使用者訂餐(while迴圈測試)  56. while(test):  57. print('好的~以下是我們的菜單\n')  58.  59. #dict的for 迴圈  60. for m in menu.items():  61. print(m)  62. choose=input('請問您想選擇?(請輸入菜名(或套餐名))')  63. print(choose)  64. t=True  65. #while進行偵錯是否輸入錯誤  66. while(not(choose =='鍋貼' or choose =='水餃' or choose =='菜頭湯' or choose =='紅茶' or choose =='豆漿'or choose =='套餐1'or choose =='套餐2')):  67. print('輸入錯誤請重新輸入\n') #使用者輸入錯誤例外處理  68. choose=input('請問您想選擇?(請輸入菜名(或套餐名))\n')  69. q=int(input('請問要幾份?'))  70. order[choose]=int(order[choose])+q  71. print('請問您還要繼續點嗎?\n')  72. keep=int(input('要(1)/不要(2)'))  73. #if判斷點餐是否繼續  74. if(keep==2):  75. test=False  76. #使用者訂餐結果與金額  77. print('以下是您的訂餐內容\n')  78. #dict的for 迴圈  79. for o in order.keys():  80. #if判斷  81. if(order[o]>0):  82. print(o+":"+str(order[o])+"份\n")  83. print('\n以下是您的訂餐總金額\n')  84. total=0  85. sum={'鍋貼':0,'水餃':0,'菜頭湯':0,'紅茶':0,'豆漿':0,'套餐1':0,'套餐2':0}  86. #dict的for 迴圈  87. for o in order.keys():  88. #if判斷  89. if(o!='套餐1' and o!='套餐2' and int(order[o])>0):  90. sum[o]=int(order[o])\*int(menu[o])  91. #新增套餐(list+dict應用) & elif判斷  92. elif((o=='套餐1' or o=='套餐2') and int(order[o])>0):  93. sum[o]=int(order[o])\*int(menu[o][0])  94. for s in sum.values():  95. total=int(total)+int(s)  96. print(str(total)+'元')  97. #使用者支付頁面與判斷(有錯誤例外處理)  98. pay=int(input('請付款，請輸入付款金額:\n'))  99. while(pay<total):  100. print('付款不足!\n')  101. pay=int(input('請付款，請重新輸入付款金額:\n'))  102. if(pay>total):  103. print('找您'+pay-total+'元，謝謝您!\n')  104. if(pay==total):  105. print('整付，謝謝您~\n')  106.  107. now = datetime.datetime.now()  108. print('最後確認您的用餐時間為(現在時間):'+now.strftime("%Y-%m-%d %H:%M:%S")+'後的兩小時喔~\n')  109.  110. print('-------------------------------------------------------------------------------------\n')  111. #模擬服務人員(劃位分析  112. print('模擬服務人員\n')  113. employee1=['店長','09123456789','女','到']  114. employee2=['店員','09987654321','男','未到']  115. employee={'1':employee1,'2':employee2}  116. employee\_num=2  117. store\_test=int(input('請輸入您要執行的操作:桌位管理(1)/員工管理(2)/員工簽到(3)'))  118. #桌位管理  119. if(store\_test==1):  120. print('訂位人數有:'+str(people\_num)+'\n')  121. print('如果想要都是兩人桌的話會需要\n'+str(people\_num//2)+'但可能有'+str(people\_num%2)+'人沒位子~\n\n')  122. print('如果想要都是三人桌的話會需要\n'+str(people\_num//3)+'但可能有'+str(people\_num%3)+'人沒位子~\n\n')  123. table={'兩人桌':2,'三人桌':3}  124. print('您可以參考目前剩餘桌位排桌位~\n')  125. for t in table.keys():  126. print('目前有:'+t+'剩餘:'+str(table[t])+'\n')  127. print('謝謝您使用本系統~\n')  128. #員工管理  129. elif(store\_test==2):  130. print('員工資料:\n')  131. for e in employee.keys():  132. print(e+'號')  133. print(employee[e])  134. e\_test=int(input('請輸入:無操作(0)/新增員工(1)/刪除員工(2)/修改員工資料(3)'))  135. #員工新增  136. if(e\_test==1):  137. name=input('姓名:')  138. tel=input('電話')  139. gender=input('性別(男|女)')  140. list\_add=[name,tel,gender,'未到']  141. employee[str(employee\_num+1)]=list\_add  142. print('已新增')  143. print('員工資料:\n')  144. for e in employee.keys():  145. print(e+'號')  146. print(employee[e])  147. #員工刪除  148. elif(e\_test==2):  149. print('請問要刪除幾號的資料?\n')  150. e\_num=int(input())  151. del employee[str(e\_num)]  152. print('已刪除')  153. print('員工資料:\n')  154. for e in employee.keys():  155. print(e+'號')  156. print(employee[e])  157. #員工修改資料  158. elif(e\_test==3):  159. print('請問要修改幾號的資料?\n')  160. e\_num=int(input())  161. name=input('姓名:')  162. tel=input('電話')  163. gender=input('性別(男|女)')  164. list\_change=[name,tel,gender,'未到']  165. employee[str(e\_num)]=list\_change  166. print('已修改')  167. print('員工資料:\n')  168. for e in employee.keys():  169. print(e+'號')  170. print(employee[e])  171. #員工簽到  172. elif(store\_test==3):  173. print('員工資料:\n')  174. for e in employee.keys():  175. print(e+'號')  176. print(employee[e])  177. print('請問要簽到幾號的資料?\n')  178. e\_num=int(input())  179. employee[str(e\_num)][3]='到'  180. print('已簽到')  181. |
| 六、執行結果  (將正確執行的結果貼入，文字為主截圖為輔，以下為例子) |
| |  |  |  | | --- | --- | --- | | 結果的流程為:  1.需要透過資料輸入訂餐與訂位系統中所需的相關資料，包含:訂餐者、人員數、菜品、菜品數量、用餐時間。  2.將上述的內容計算出總金額，與結果顯示給使用者，並模擬使用者付款。  3.於服務人員端，可以透過上述的資訊，自動化的進行桌分配(人數過多時，需要幾個桌位)  (我有在程式中，做例外處理)  4.訂餐系統針對list、dictionary增加套餐功能(增加餐廳行銷與使用者選擇)  5.使用者輸入自身身體數值，提供餐廳飲食建議  6.針對服務人員端新增員工資料管理與簽到系統(未來可以輸出檔案)  程式設計思維，需考慮人(user)與AI夥伴(App)的人際互動對話  (以下針對這次的主要新增功能做展示，)尼它細節操作，請參考上周 |  |  | | (下一頁繼續) |  |  | |
| 七、學習意涵詮釋  (可分為兩個部分：一、學到什麼程式語法 二、學到解決什麼問題，以下為例子) |
| 1.在條件分之中有: if()、elif()、else，須注意記得內縮，內縮是python文法的一部份。  2.if成立則執行下面內縮之敘述，elif():上述不成立下一條件判斷，else:如果if或elif的條件已成功就不會執行else，但以上條件皆不成立，則會執行else，同時須記得不行將else放在elif或if之前。  3.需要學習程式防錯的必要。  4. 可以在判斷中使用and or使程式更簡潔。  5. 迴圈(重複執行相同的動作): for、while。  6.for: 執行固定次數的迴圈，在for中除了可以要求次數，也可以對串列型別的各項資料、重覆相同的處理動作。  7.while: 有可能不小心設成無窮迴圈，可以使用^C跳出迴圈。  8. break:跳出迴圈。  9. continue :繼續重複迴圈，continue後的動作不執行。  10.break與continue可以利用判斷是進行跳出迴圈或繼續執行。  11. 函式:將多項處理動作彙集在一起, 供後續呼叫使用，也是一種重複執行的概念。  12.函式除了自訂義，也有python內建的，ex: len()、max()、min()…。  13. 例外處裡的方法:使用try 與except。  14.我在過程中學會使用更好的方法做資料處理，也在過程中學習到飲食結合商業的方法。  15.我透過此次的作業，我對於程式開發與結合議題有更好的認識，我本來只是想開發一個簡單的系統，但隨著上周的作業，我在這周想到或許可以結合人的健康需求搭配商業行銷，使得使用者與店家之間可以有更好的連結，也讓我認識友善店家的概念! |
| 八、參考說明  (列出所參考資料並說明如何應用或修改其內容，修改ChatGPT亦可，以下舉例) |
| 無參考資料 |