

2021 OOP I

Personal Project - CheckPoint 7

作業繳交事前聲明：

- 1.請注意每次作業繳交日期！！！！
- 2.遲交之作業仍必須補繳交，若至期末尚有任何作業未繳交，學期成績將從59分向下計算！！！！

CheckPoint 7 繳交期限：

此次作業繳交期限為 **2021/6/27 23:59:00**，請同學於繳交期限內將作業(學號 `_checkpoint7.java`、學號 `_checkpoint7_baseState.java`、學號 `_checkpoint7_idleState.java`、學號 `_checkpoint7_lift.java`、學號 `_checkpoint7_loadingState.java`、學號 `_checkpoint7_movingState.java` 與程式文件說明檔 (請同學注意副檔名是否正確)，請務必注意英文字母大小寫 (Ex. A1083301_checkpoint7.java))上傳至 google classroom 內指定空間，逾時不候，遲交者將以每遲交一天扣作業成績一分。請同學繳交作業之前確認繳交的內容是否正確且檔案能正常編譯執行以及檔名是否正確無誤，若因檔案問題而無法正常運作，成績會以 0 分計算。同時，請同學在寄信時注意該有的信件禮儀。

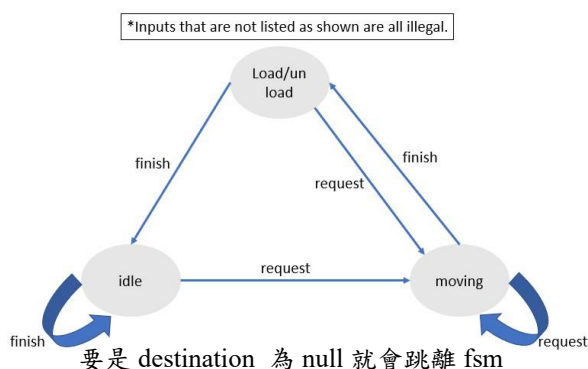
繳交的作業檔案除了上述的.java 外，仍需繳交程式文件說明檔，分別儲存成三種格式 (學號 `_checkpoint7.html`，學號 `_checkpoint7.pdf`，學號 `_checkpoint7.docx`)，與程式檔一起上傳至 google classroom 指定空間。

Checkpoint 7 功能需求:

此次作業要求模擬電梯上升下降狀態，來實作有限狀態機的概念，藉由使用者輸入的樓層，來顯示電梯整個移動過程，程式輸出結果如（圖一）所示，此次作業可參考有限狀態機圖表（圖二）及狀態轉移表（圖三）來了解狀態轉移模式。

```
a1073310@2021_OOPI_SERVER:ckp7$ java checkpoint7 1 5 3 6 8
Now it's on idle state.
Moving, lift is on floor==>1
Now it's on loading/unloading state.
Now it's on idle state.
Moving, lift is on floor==>1
Moving, lift is on floor==>2
Moving, lift is on floor==>3
Moving, lift is on floor==>4
Moving, lift is on floor==>5
Now it's on loading/unloading state.
Now it's on idle state.
Moving, lift is on floor==>5
Moving, lift is on floor==>4
Moving, lift is on floor==>3
Now it's on loading/unloading state.
Now it's on idle state.
Moving, lift is on floor==>3
Moving, lift is on floor==>4
Moving, lift is on floor==>5
Moving, lift is on floor==>6
Now it's on loading/unloading state.
Now it's on idle state.
Moving, lift is on floor==>6
Moving, lift is on floor==>7
Moving, lift is on floor==>8
Now it's on loading/unloading state.
Now it's on idle state.
```

（圖一）



（圖二）

Current State	Input		Next State
	s	s'	
idle	request	finish	moving
idle	0	1	idle
Load/unload	1	0	moving
Load/unload	0	1	idle
moving	1	0	moving
moving	0	1	Load/unload

（圖三）

CheckPoint 7 作業說明：

1. 作業提供檔案描述

此次作業所需要的檔案已上傳至 google classroom 中，請同學自行下載撰寫。作業中的檔案裡有一些已經設定好的變數名稱，且另有一些 "TODO" 的敘述與要求，請同學依據下方敘述，撰寫程式來完成所有作業要求。**檔案中架構及變數請勿更改，但可額外宣告函數及變數。**以下簡述檔案的架構與變數名稱：

以下兩點提醒請同學在完成這次作業注意：

1. 輸出並非我們要求的文字同學將會得到 0 分，下面提供會使用到文字
"Now it's on idle state."
"Moving, lift is on floor==>"
"Now it's on loading/unloading state."
2. 如果同學有在程式碼中更改檔名，需自行檢查是否有其他需要更改的部分，即使不在 TODO 裡，請同學注意!!

The following two points you need to notice:

1. Typos isn't to blame for the zero, here are the text you'll need.
"Now it's on idle state."
"Moving, lift is on floor==>"
"Now it's on loading/unloading state."
2. Noted that if you change the name of a file, you may also need to check if the code that isn't marked as TODO may need to be modified as well.

checkpoint7_lift.java 本次作業電梯上升狀態的程式檔，以下敘述各部分需完成的內容：

1. TODO 1

同學須完成建立建構子

- 提示一： 首先同學三個不同狀態的物件，並將其目前狀態設置為 idle 狀態
- 提示二： 設置 dList 基於傳入的 destination
- 提示三： 將目前的樓層設為 1，並將目標樓層使用 next() 方法來設定

TODO 1

Here you have to set up the constructor.

- Hint1: **First you need to create objects of three states**, and set the current state to idle state.
- Hint2: Set up the dList bases on destination be passed in.
- Hint3: Set the current floor to 1, and set the target floor by method next().

2. TODO 2

當這個方法被呼叫時，同學須從 destination list 裡挑選下一個目標樓層

- 提示一： 同學必須回傳 destination list 裡的第一個元素，並在回傳後將該元素刪除
- 提示二： 如果 destination list 裡沒有任何元素則回傳 null

TODO 2

Here you have to offer a new destination from the list once this method is called.

- Hint1: You have to always return the first element from the destination list, also remove the first element upon returning.
- Hint2: Return null if the dList has nothing left.

checkpoint7_idle.java 本次作業電梯空閒狀態的程式檔，以下敘述各部分需完成的內容：

3. TODO 3

同學需要以有限狀態機圖來定義電梯的空閒狀態

- 提示一：一旦觸發 doState 方法，同學需印出 "Now it's on idle state." (使用 println)
- 提示二：這個狀態不會做任何事但會轉換到 moving 狀態

TODO 3

Here you have to define the idle state based on the fsm graph.

- Hint1: Once trigger the doState method you have to print out "Now it's on idle state.". (Use println)
- Hint2: This state won't do anything but switch to moving state.

checkpoint7_loading.java 本次作業電梯開門狀態的程式檔，以下敘述各部分需完成的內容：

4. TODO 4

同學需要以有限狀態機圖表來定義電梯開門狀態

- 提示一：一旦觸發 doState 方法，同學需印出 "Now it's on loading/unloading state."（使用 println）
- 提示二：注意如果目前樓層與目標樓層一樣的話，表示會先拿到下一個目標樓層並將其狀態為 idle 狀態。反之，則轉換其狀態為 moving 狀態

TODO 4

Here you have to define the loading state based on the fsm graph.

- Hint1: Once trigger the doState method you have to print out "Now it's on loading/unloading state.". (Use println)
- Hint2: Noted that if the current floor equals to target floor that means it will first get a new destination floor then switch to idle state, otherwise it'll switch to moving state.

checkpoint7_movingState.java 本次作業電梯移動狀態的程式檔，以下敘述各部分需完成的內容：

5. TODO 5

同學需要以有限狀態機圖表來定義電梯移動狀態

- 提示一：一旦觸發 doState 方法，同學需印出 "Moving, lift is on floor==>currentFloor"（使用 println）
- 提示二：注意如果目前樓層與目標樓層一樣的話，須轉換其狀態為 loading 狀態。反之，如果目前樓層與目標樓層

不一樣的話，同學需要使其每次上/下一層樓並切換至 moving 狀態

TODO 5

Here you have to define the moving state based on the fsm graph.

- Hint1: Once trigger the doState method you have to print out "Moving, lift is on floor==>currentFloor". (Use println)
- Hint2: Noted that if the current floor equals to target floor that means it will switch to loading state, otherwise it'll go up/down one floor each time and switch to moving state.

checkpoint7.java: 本次作業的主程式檔，以下敘述各部分需完成的內容：

6. TODO 6

同學需要讀入使用者在命令列輸入的樓層資料，並將其儲存至 arraylist 裡

- 提示一： 假設同學執行程式時輸入的指令為 "java checkpoint7 1 5 3 6 8"，其中 1 5 3 6 8 則是目標樓層
- 提示二： 電梯會一個一個到達存在 destination list 裡的目標樓層

TODO 6

Here you read in the data from command line storing them into a arraylist.

- Hint1: You'll execute this program by "java checkpoint7 1 5 3 6 8", that is to say the data will be 1,5,3,6, and 8.
- Hint2: The lift will goes to floors in the destination list one by one.

2. 繳交內容與評分方式

本次作業繳交時，應完成以下要求：

a、 繳交的作業檔名為**學號_checkpoint7.java**

學號_checkpoint7_baseState.java

學號_checkpoint7_idleState.java

學號_checkpoint7_lift.java

學號_checkpoint7_loadingState.java

學號_checkpoint7_movingState.java

b、 檔名需在 command line 加入

如：**java 學號_checkpoint7 樓層數字**

註:批改作業之測試資料不會出現 ≤ 0 的樓層，亦不會出現出重複相同的樓層（如 java 學號_checkpoint7 1 1），請同學注意

c、 作業批改環境為 linux，指令如下：

javac 學號_checkpoint7.java

java 學號_checkpoint7 1 5 3 6 8

若無法編譯，**以零分計算**。

d、 下指令後，使程式輸出與預期相符，輸出需與（圖一）相同。

作業的評分標準依以上要求的完成度為依據。同時請切記，**任何作業抄襲的行為都是不允許的，且作業要求輸出的檔案、數值，切勿使用不符合要求的方式產出**，如果違反以上規定將直接影響到你的作業成績。

若同學對於作業有疑問，歡迎詢問助教（或老師）

但請切記該有的禮節以及所有理應注意事項。

程式文件需求說明

封面 (Cover)：

需包含高大校徽與作業名稱。

需求描述 (Description of requirements)：

說明你的程式作業將會達到什麼需求。在作業中，這裡的描述通常會是助教的 Basic Requirement。同學們如果有額外希望達到的需求，也可以在這裡增加描述。(注意:這裡的描述必須跟作業實際能夠達到的需求相符合，否則應該增加描述不符合需求的原因)

OO 的考量 (The consideration of Object-Oriented)：

請同學說明在撰寫作業中，對於 OOA/OOD/OOP 三個階段的考量。

※建議同學可以在文件繪製 UML 與 Class Diagram 來解釋類別間的屬性與關係

功能/邏輯說明 (Description of function/logic)：

說明自己的程式裡，自己使用了或者設計了什麼「功能」或「邏輯」，並說明這個功能/邏輯的用途與執行方式。

使用說明 (Instructions for use)：

說明你的程式要如何執行，進行的過程中有什麼地方需要注意等等。這部分主要為了教別人如何用你的程式所寫的教學文件。

其他 (Other)：

任何有助於讓別人了解你的作業之說明。

同學請注意繳交**程式文件說明檔**時，需分別儲存成**三種**格式（**學號_checkpoint7.html**, **學號_checkpoint7.pdf**, **學號_checkpoint7.docx**）並上傳到 google classroom 內指定空間