

# ~HW2(延伸HW1)~

- 建立一個Car類別，類別的內容如下：

## 1. 成員變數

變數名稱	型別	說明
carId	String	車牌號碼
engineId	String	引擎號碼
color	String	車身顏色

1. 建構方法：
  - 定義一個對engineId初始化的建構方法
  - 定義一個對所有成員變數初始化的建構方法
2. 成員方法
  - 定義engineId只能讀取，不能修改的方法
  - 定義carId與color可以讀取的方法
  - 定義carId與color可以修改的方法
3. 成員變數的存取修飾子皆需設為private
4. 成員方法的存取修飾子皆需設為public

1. 建立一個CarFactory類別，類別的內容如下：

## 1. 成員變數

變數名稱	型別	說明
numOfCars	int	紀錄工廠生產了多少台車

1. 建構方法:無
2. 成員方法

方法名稱	回傳值型別	說明
getProducedCars	int	取得工廠生產了車量的總數
produceCar()	Car	<ol style="list-style-type: none"> <li>1. 隨機生成8位數字格式的engineId</li> <li>2. 隨機生成8個字元的carId，規則如下 <ul style="list-style-type: none"> <li>● C[二個英文字母]-[四個數字] 例如：CBF-5463</li> </ul> </li> <li>3. 隨機生成color，顏色有下列三種: White, Black, Blue</li> <li>4. 將上述生成的三個屬性值，建立Car物件，並回傳</li> </ol>
updateCarId(Car car, String carId)	boolean	<ol style="list-style-type: none"> <li>1. 更新輸入Car物件的carId值</li> <li>2. 在更新前需檢查輸入的carId值是否符合規則C[二個英文字母]-[四個數字]</li> <li>3. 更新成功回傳true，失敗回傳false</li> </ol>
updateCarColor(Car car, String color)	boolean	<ol style="list-style-type: none"> <li>1. 更新輸入Car物件的color值</li> <li>2. 在更新前需檢查輸入的color值是否符合規則</li> <li>3. 更新成功回傳true，失敗回傳false</li> </ol>

● **建立一個TaxiFactory類別，類別的內容如下：**

1. 成員變數

變數名稱	型別	說明
numOfCars	int	紀錄工廠生產了多少台車

1. 建構方法:無
2. 成員方法

方法名稱	回傳值型別	說明
getProducedCars	int	取得工廠生產了車輛的總數
produceCar()	Car	1. 隨機生成8位數字格式的engineId 2. 隨機生成8個字元的carId，規則如下 T[二個英文字母]-[三個數字] 例如：TEG-546 3. color顏色固定為Yellow 4. 將上述生成的三個屬性值，建立Car物件，並回傳
updateCarId(Car car, String carId)	boolean	1. 更新輸入Car物件的carId值 2. 在更新前需檢查輸入的carId值是否符合規則 3. 更新成功回傳true，失敗回傳false

● 建立一個ElectricCarFactory類別，類別的內容如下：

1. 成員變數

變數名稱	型別	說明
numOfCars	int	紀錄工廠生產了多少台車

1. 建構方法:無
2. 成員方法

方法名稱	回傳值型別	說明
getProducedCars	int	取得工廠生產了車量的總數
produceCar()	Car	<ol style="list-style-type: none"> <li>1. 隨機生成8位數字格式的engineId</li> <li>2. 隨機生成8個字元的carId，規則如下 E[二個英文字母]-[四個數字] 例如：EAT-5463</li> <li>3. 隨機生成color，顏色有下列三種:White, Black, Blue</li> <li>4. 將上述生成的三個屬性值，建立Car物件，並回傳</li> </ol>
updateCarId(Car car, String carId)	boolean	<ol style="list-style-type: none"> <li>1. 更新輸入Car物件的carId值</li> <li>2. 在更新前需檢查輸入的carId值是否符合規則E[二個英文字母]-[四個數字]</li> <li>3. 更新成功回傳true，失敗回傳false</li> </ol>
updateCarColor(Car car, String color)	boolean	<ol style="list-style-type: none"> <li>1. 更新輸入Car物件的color值</li> <li>2. 在更新前需檢查輸入的carId值是否符合規則</li> <li>3. 更新成功回傳true，失敗回傳false</li> </ol>

- 建立一個static類別，名為CarIdUtil，其內容如下

1. static方法

方法名稱	回傳值型別	說明
generateCarId()	String	隨機生成8個字元的carId，規則如下  C[二個英文字母]-[四個數字]  例如：CBF-5463
generateTaxiId()	String	隨機生成8個字元的carId，規則如下:  T[二個英文字母]-[三個數字]  例如：TEG-546
generateElectricCarId()	String	隨機生成8個字元的carId，規則如下:  E[二個英文字母]-[四個數字]  例如：EAT-5463
isValidCarId(String carId)	boolean	檢查輸入的carId值是否符合規則:  C[二個英文字母]-[四個數字]  符合回傳true，不符合回傳false
isValidTaxiId(String carId)	isValidCarId(String carId)	boolean
isValidElectricCarId(String carId)	boolean	檢查輸入的carId值是否符合規則:  E[二個英文字母]-[四個數字]  符合回傳true，不符合回傳false
generateLetter()	String	隨機生成一個A-Z中的一個字母
generateNumber()	String	隨機生成一個0-9中的一個數字

- 建立一個Organization類別，類別的內容如下：

1. 成員變數

變數名稱	型別	說明
numOfWhiteCars	int	紀錄買了多少台白色的車
numOfBlackCars	int	紀錄買了多少台黑色的車
numOfBlueCars	int	紀錄買了多少台藍色的車
numOfYellowCars	int	紀錄買了多少台黃色的車

1. 建構方法:無
2. 成員方法

方法名稱	回傳值型別	說明
buyCar(Car car)	void	加入購買的車輛，並計算購買車輛的車身顏色
getNumOfWhiteCars()	int	取得買入白色車量的總數
getNumOfBlackCars()	int	取得買入白色車量的總數
getNumOfWhiteCars()	int	取得買入白色車量的總數
getNumOfWhiteCars()	int	取得買入白色車量的總數

- 建立一個Main類別與main方法，其內容如下
  1. 產生一個Organization物件
  2. 該Organization購買了下列車輛
    - i. 3輛Car
    - ii. 2輛TaxiCar
    - iii. 5輛ElectricCar
  3. 印出該Organization購買各種車色的數量，例如：

**White cars: 4**

**Black cars: 1**

**Blue cars: 3**

**Yellow cars: 2**

