Redesign Report

R05922130 王瀚磊

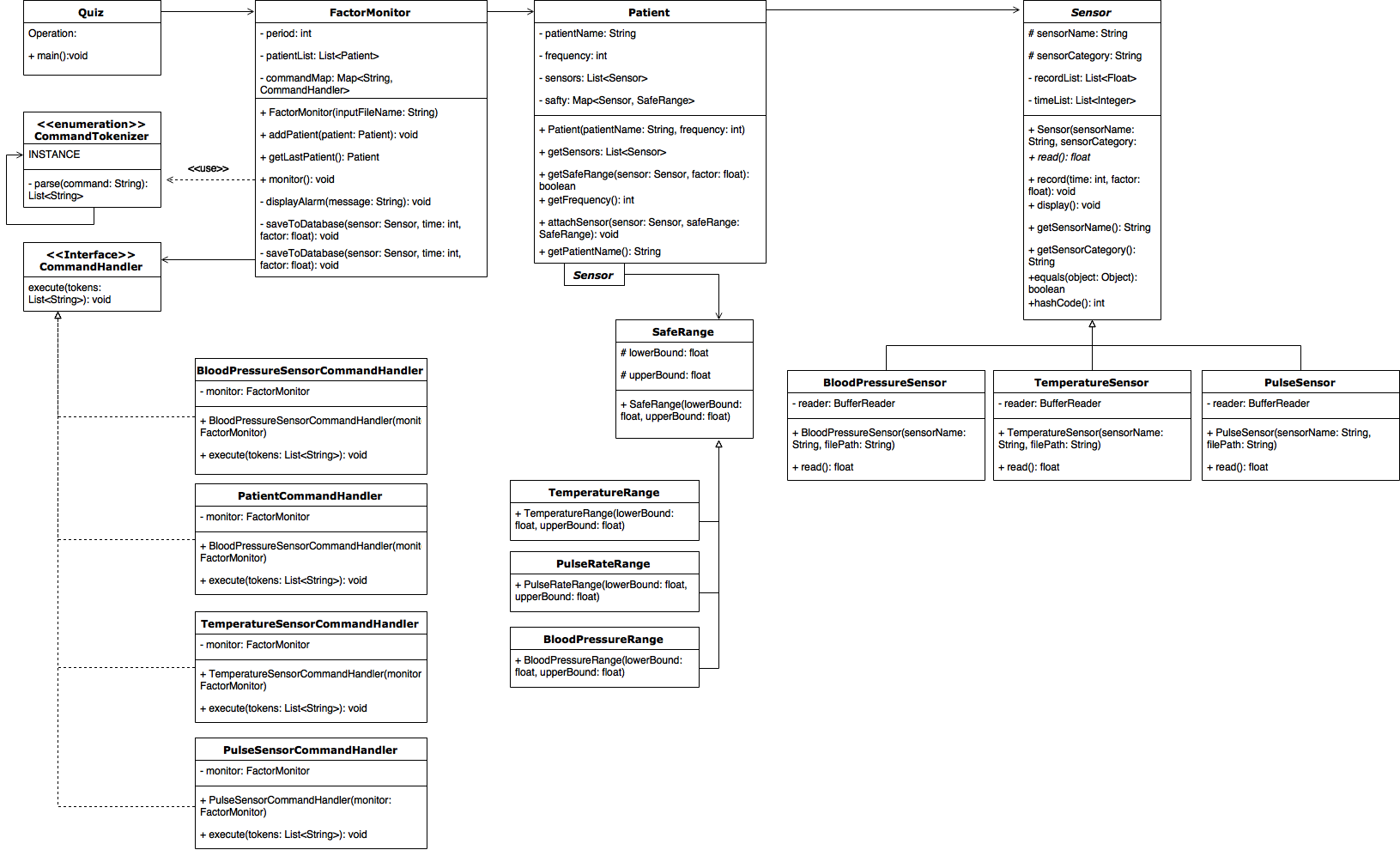
1. 簡介

在這兩次的作業中，我接採取一樣的做法。先想好怎麼寫後先創建好每一個class檔案，並且在每個檔案中都用TODO的方式先寫好每個class應該做些什麼事情後再開始時做，在第一次作業耗費了四個小時寫完，第二次作業耗費兩個小時完成。

1. 變動說明：
   1. 在第一次作業中，我把所有sensor category皆視為同一個class，僅利用其中一個attribute存取不同sensor之category name以及name。且第一次作業我將所有的database在收到input後隨即把所有的factor讀取出來存在list之中，之後再做monitor的時候才一個一個叫出來處理。
   2. 在第二次作業改進中，我將不同sensor category皆創立其專屬之class，且我也將sensor讀取database factor的方式改為monitor時才一行一行讀取。另一個比較不同的地方是saferange也創立其專屬之abstract class，並由不同sensor category之saferange class去繼承它。
   3. 這次改動要求我們照著老師所提供的class diagram寫程式，然而我發現其中有些部分並沒有寫得很清楚，因此造成在implement之中有些困難，必須再根據自己的想法揣測哪些class需要增加哪些operation，或是一些參數。最後的結果如圖一所示。
2. 心得

兩次的架構其實變化不多，也因為再第一次架構的設計時有多想一下，因此在接下來的變動中並不會造成太大的困難。這是我第二次利用所學的design pattern做一些實際例題，我所應用的包涵command pattern以及singleton pattern。我利用command pattern處理所有input之command type，例如:patient, BloodPressureSensor…等等，並且利用singleton創立一個專門拿來處理command tokenizer之部分。

在這次作業中我更加深刻的體悟到利用一些design pattern撰寫code時，可以更多元的考慮一些over design的部分，例如未來要是增加一些command只要再多建立幾個class即可。然而這邊的tokenizer我有點不確定寫成singleton是否是對的，因為最後只有在FactorMonitor這個class中有使用到，除非未來的改動中有造成其他class也會用到，否則我認為我這樣的設計是不好的。



圖一、程式改進後之Class Diagram