AOOP Final Project – Program Elevator Fighter

1. Introduction

In the future, Engineering Building V is torn down and rebuilt because it is too old. The new building has 30 floors and there is only one elevator. Now, you are a professional engineer and asked to design the new elevator system. Firstly, you have to implement an elevator simulator to simulate the elevator system. The elevator is a special elevator, that is, it is a program elevator, which means that if one person wants to enter or leave the elevator, he/she has to solve a program. Each floor has its corresponding program. For example, you have to solve a prime problem on the first floor, solve a string problem on the second floor, and so on. In order to save everyone's time to take the elevator, you have two things to do. One is to improve the elevator scheduling, the other is to improve the programs performance on each floor. For finding out which systems designed in this class is the best one, we will have a competition.

2. Case Study of Elevator Simulation

A company intends to build a **two-story office** building and equip it with the latest elevator technology. The company wants you to develop an object-oriented software simulator that models the operation of the elevator to determine of this elevator will meet their needs.

The elevator, which has a capacity of **one person**, is designed to conserve energy, so it only moves when necessary. The elevator starts the day waiting with its door shut on **floor 1** of the building. The elevator, of course, alternates directions----first up, then down.

Your simulator includes a **clock** that begins **the day set to time 0 and that "ticks" once per second**. The "**scheduler**" component of the simulator randomly schedules the arrival of the first person on each floor. When the clock's time becomes equal to the time of the first arrival, the simulator "creates" a new person for the specified floor, and places the person on that floor. The person then presses the button on that floor to summon the elevator. The person's destination floor in never equal to the floor on which that the person arrives.

If the first person of the day arrives at floor 1, the person can immediately get on the elevator (after pressing the button and waiting for the elevator's door to open, of course). If the first person arrives at floor 2, the elevator proceeds to floor 2 to pick up that person. The elevator requires **five ticks** of the clock to **travel between floors**.

The elevator signals its arrival at a floor by turning on a light above the elevator door on that floor and by sounding a bell inside the elevator. The button on the floor and the button in the elevator for that floor are reset, the elevator opens its door, the passenger---if there is one whose destination is that floor----gets out of the elevator, another passenger---if there is one waiting on that floor---gets into the elevator and press a destination button, and the elevator closes its door. For simplicity, assume that all of the events that happen once the elevator reaches a floor and until the elevator close its door on that door, take zero time. The elevator always knows what floor it is on and what floor it is going to.

At most, one person can be waiting on each floor at any time, so if a floor is occupied when a new

person (i.e., not a person already on the elevator) is due to arrive at that floor. The new arrival is rescheduled for one second later. Assume that people arrive at random on each floor every 5 to 20 seconds.

Your goal is to implement a working soft simulator program that runs according to these specifications. Your program should simulate several minutes of the elevator's operation and determine if the elevator will successfully meet the anticipated traffic requirements in the office building.

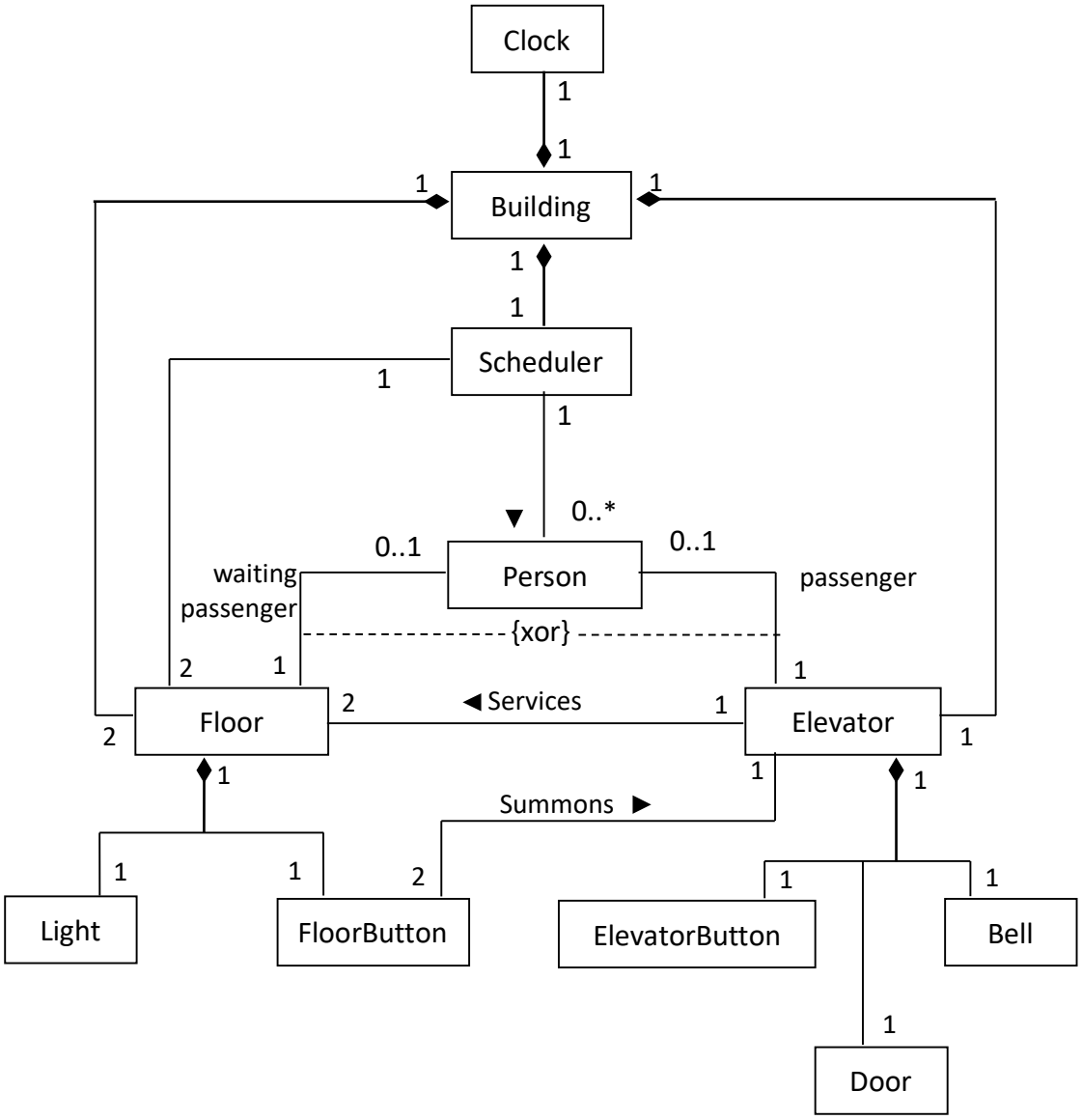
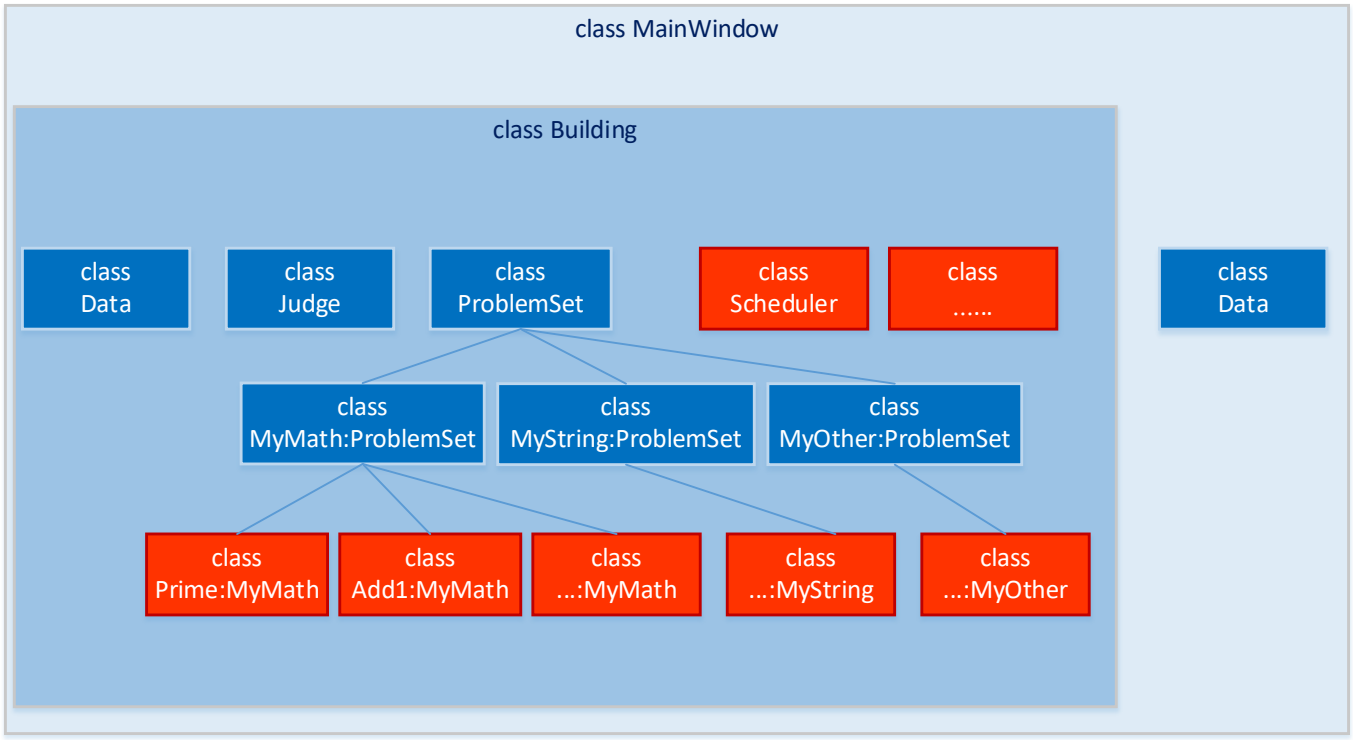


Fig 1. Elevator class Hierarchy

3. Overall System



4. Competition Rules

1) 兩人一組。
2) 比賽時採用的 Judge System 為助教提供，比賽當下會請各組將自己的檔案加進助教提供的 Project Files，因此請在比賽前熟悉好如何做這件事。
3) 初賽：初賽規制採 F1 排位賽制，最後一堂會讓各組在課堂上各自跑自己的程式，在該堂課結束前取最好的成績做排位，並依排位填入對應的雙淘汰制賽程表中。另外會取前 8 名(暫定)做為種子選手(可以少比最底層一輪)。
4) 決賽：決賽採用雙淘汰制，依雙淘汰賽程表進行比賽，兩組同時在兩台電腦跑程式並比較積分結果，一場比 5 局，每一局輪流換電腦，第五局以丟銅板決定電腦。
5) 計分方式
   小程式： 比相對秒數，秒數少的組別贏得該題分數。
   電梯路徑規劃： 計算電梯移動的距離。
6) 決賽每一場賽前會給各組 3 分鐘時間檢視對方的程式，若對對方程式有疑義可以向該場裁判助教提出，若經由助教判定程式有問題即判定失去資格。
7) 請不要作弊，請將心力放在增進程式效能，不要花心思想奧步。
8) 比賽時請不要賴皮。
9) 以和為貴。
10) 專題成績會依據比賽最終排名結果分配分數，ex: 第一名 100 分、第二名 98 分、...最後 5 名 60 分(暫定)。
11) 上述規則到比賽前皆會視狀況做合理修訂，若同學有任何疑問或覺得規則有所遺漏隨時皆可提出來給助教做合理修訂。