

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. Project Content and Requirement

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 just only one elevator to conserve energy.

Now, you are a professional engineer asked to design the new elevator system by using the techniques of the object-oriented software. Your goal is to implement a working smart elevator simulator program that runs according to these specifications to achieve the following:

1. Improve the elevator scheduling and minimize the number of total moving floors to successfully meet the anticipated traffic requirements in the Building V.
2. Execute each floor program correctly and minimize the execution time of all the floor programs.

The elevator, which has a capacity of 10 persons, 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 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 elevator is a special elevator, that is, it is a program elevator, which means that if one person wants to 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 before leaving the elevator.

If there are “n” persons have the same destination floor f1, then you need to run the floor f1 program with n different data set, and each data set m times. The group with correct answer and less execution time will win this floor score (check answer first, and then compare the execution time).

These processes will be handled by the fair and smart “Judge”. The “Judge” component of the elevator simulator provides one set of conditions:

1. the number of persons on each floor.

2. and their destination floors.
3. The floor number which the elevator is parked on.

When your elevator simulator accept these conditions:

1. The smart elevator simulator “creates” the persons for the specified floor, and places the person on that floor. The person’s destination floor is never equal to the floor on which that the person arrives.
2. Update the status data of the “Elevator”, and these status data will show on the windows GUI panel.
3. According to conditions provided by the “Judge”, the “Scheduler” need to make the optimized decision:
 - (a). Create the schedule to transport these persons with minimum number of total moving floors.
 - (b). The Scheduler handshakes and communicates with “Judge” to complete its whole schedule.
 - (c). Once these persons reach their destination floor:
 - (I). They have to solve the corresponding floor program.
 - (II). The “Judge” will monitor and determine the score of each floor for your team, and the corresponding run time messages will be displayed on windows GUI panel.
 - (III). The “Judge” will monitor the whole process of executing floor programs and determine which team win the total floor scores.

The conclusion:

Your elevator simulator needs to

1. meet the anticipated traffic requirements,
2. solve each corresponding floor program within minimum time.
3. and optimize the traffic time.

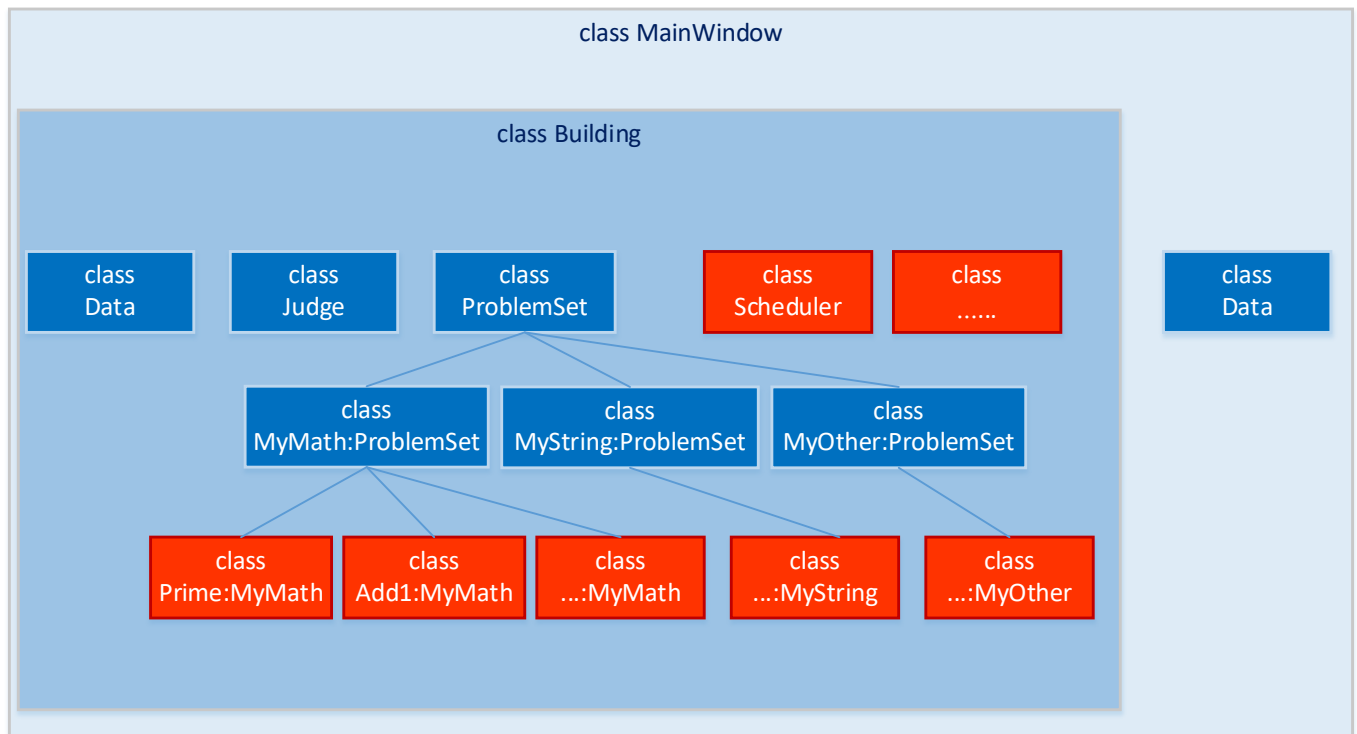
➤ Basic Requirement

In this elevator simulator program for the management part, you need to provide the following classes:

1. class Building,
2. class Scheduler
3. class Judge
4. class Floor
5. class Elevator
6. class Data.

According these specification, you need to add five more classes in your class hierarchy.

3. Overall System



4. Project Schedule

Week	Date	Content
7	2019.10/30.10/31	Project Introduction
9	2019.11/13.11/14	1. Introduction to UML Class Diagram 2. Announcement of Project System Specification 3. Announce Parts of Floor Programs 4. Floor Programs Using Polymorphism (Problem1) 5. Floor Programs (Wedx3, Thux3)
10	2019.11/20.11/21	1. Introduction to Database and MySQL (Problem1) 2. Floor Program (Wedx2, Thux2)
11	2019.11/27.11/28	1. Class Diagram of Elevator Hierarchy Demo (One Page word) 2. Announcement of Double-Elimination Tournament Table 3. Lobby Database Program (Problem1) 4. Floor Program (Wedx2, Thux2)
12	2019.12/04.12/05	1. Definition of Hand Shake Flow (Input/Output of Scheduling/Floor Programs) 2. Elevator Scheduling Simulation (Problem1) 3. Floor Program (Wedx3, Thux3)
13	2019.12/11.12/12	1. Elevator Scheduling Demo (Complete the System) 2. Floor Program (Wedx3, Thux3)
15	2019.12/25.12/26	Preliminary - Qualifying

5. 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。