1. Overview

Program is designed following the MCV design pattern. There is two main classes in package Model, one class in package View and one class in package Controller.

The Model which consists two class **Day** and **Worker** presented for the weekdays and the workers.
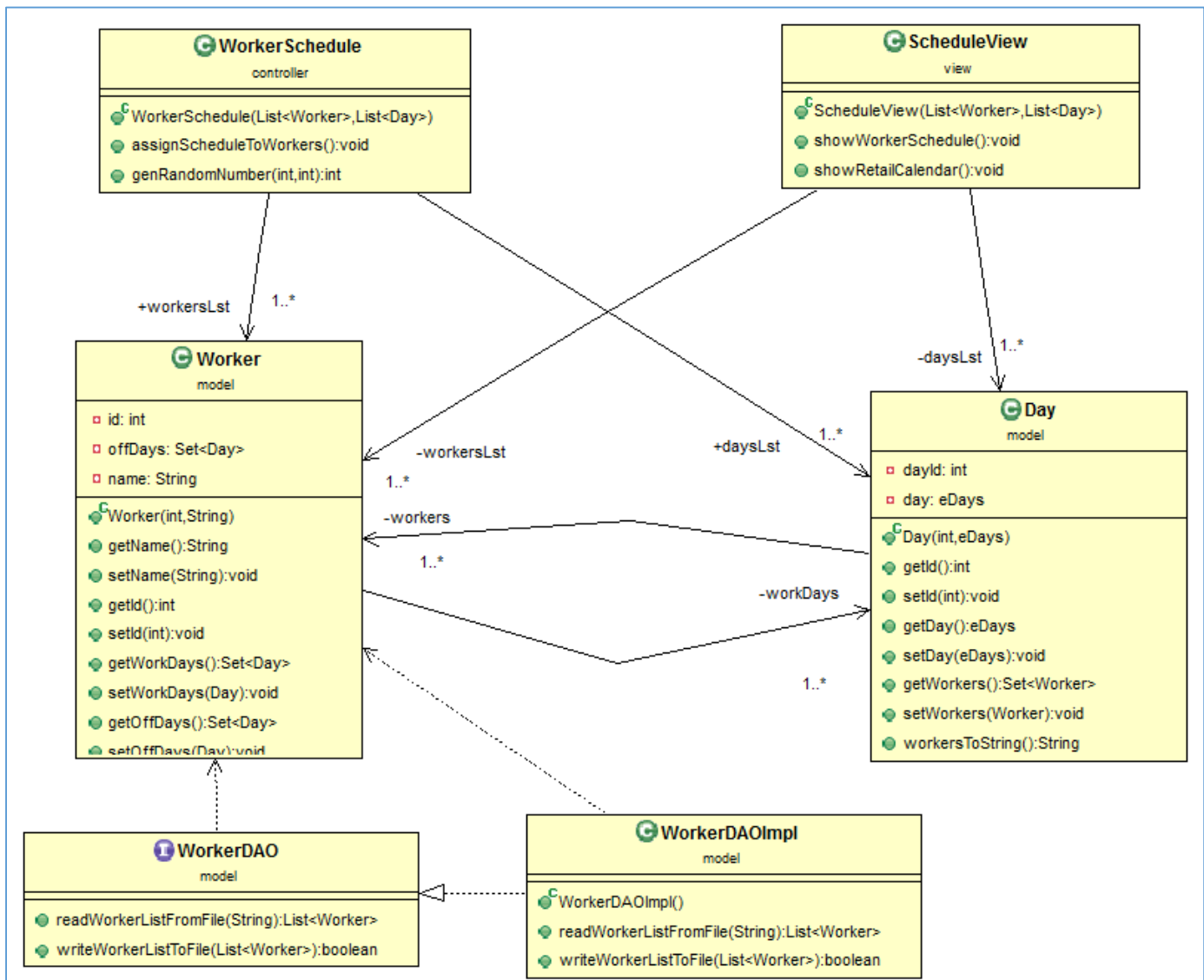
The Controller consists class **WorkerSchedule** which processes the data in models. Its main method is *assignScheduleToWorkers* which calculates and assign tasks for each worker. This method is main algorithm of program.

The View consists class **ScheduleView** just for display information of the models.

Please note that the program is designed emphasizing to software structure design than focus on algorithm with expectation that the software is Stability, Flexibility, Reusability and Scalability.

However, we can improving the algorithm by implementation a better algorithm in method *assignScheduleToWorkers* of class **WorkerSchedule**.

Below is class diagram of program:

The program will start from function *main* of class Main to initializing resource. Initialization the models, controller and view then to execute controller to calculation the schedule and in the end is to execute the view to display the results calculated by the controller.

2. How to Run The Program

       a.   Import project to Eclipse EE
       b.   Go to class …/main/Main.java and Run as "Java Application"

```
19  public static void main(String[] args) {
20
21      List<Worker> workersLst ;
22      List<Day> daysLst = new ArrayList<Day>();
23
24      WorkerSchedule workerSchedule;
25      ScheduleView scheduleView;
26
27      // 1. Initialize the resource file path
28      String filePath = "resource/Workers_1.txt";
29
30      // 2. initialize workersLst by reading workers from input text file
31      WorkerDAO workerDAO = new WorkerDAOImpl();
32      workersLst = workerDAO.readWorkerListFromFile(filePath);
33
34      // 3. initialize list of weekdays, from Mon to Sun. Put all weekdays
35      for (int i = 0; i <= 6; i++) {
36          daysLst.add(new Day(i, Util.numToDay(i)));
```