Refactor Report

Cheng Suk Yin 53035683

Schedule.java

From

|  |
| --- |
| ArrayList<Timeslot> timeslots = new ArrayList<Timeslot>();  readTimeslots(timeslots, "CS3343\_data2.txt"); //Extract method |

To

|  |
| --- |
| ArrayList<Timeslot> timeslots = new ArrayList<Timeslot>();  readTimeslots(timeslots, "CS3343\_data2.txt"); //Extract method  System.out.println(timeslots.size());    // validate input  validateInput(ArrayList<Timeslot> timeslots)    // input and validate constraints  parseInputFile(inputFile, timeConstraint, timeGapConstraint, requiredConstraint, buildingConstraint); |

From

|  |
| --- |
| for (String i : uniqueCourses) {  ArrayList<Timeslot> t = extractTimeslotsByCode(timeslots, i);  ArrayList<Timeslot> allLectures = extractTimeslotsByType(t, "Lecture");  ArrayList<Timeslot> allTutorials = extractTimeslotsByType(t, "Tutorial");  // filter Timeslot according to constraints  HashMap<String,ArrayList<Timeslot>> slot = new HashMap<String,ArrayList<Timeslot>>();  if (allLectures.size()>0)  slot.put("Lecture", allLectures);  if (allTutorials.size()>0)  slot.put("Tutorial", allTutorials);  uniqueCourseTimeslots.put(i, slot);  } |

To

|  |
| --- |
| for (String i : uniqueCourses) {  ArrayList<Timeslot> t = extractTimeslotsByCode(timeslots, i);  ArrayList<Timeslot> allLectures = extractTimeslotsByType(t, "Lecture");  ArrayList<Timeslot> allTutorials = extractTimeslotsByType(t, "Tutorial");  // filter Timeslot according to constraints  System.out.println(allLectures.size());  System.out.println(allTutorials.size());  HashMap<String,ArrayList<Timeslot>> slot = new HashMap<String,ArrayList<Timeslot>>();  if (allLectures.size()>0)  slot.put("Lecture", allLectures);  if (allTutorials.size()>0)  slot.put("Tutorial", allTutorials);  uniqueCourseTimeslots.put(i, slot);  } |

Add

|  |
| --- |
| for (String i : uniqueCourses) {  System.out.println(i + " - " + permutatedUniqueCourseTimeslots.get(i).size());  }  System.out.println(permutatedUniqueCourseTimeslots);  System.out.println(permutatedUniqueCourseTimeslotsList);    System.out.println("========="); |

Add

|  |
| --- |
| System.out.println(allPerm.get(0).size());  for (ArrayList<Timeslot> i : allPerm.get(0))  /System.out.println(i); |

From

|  |
| --- |
| for (int i=0; i < validPermutatedUniqueCourseTimeslotsList.size(); i++) {  ArrayList<Timeslot> l = validPermutatedUniqueCourseTimeslotsList.get(i);  RequiredConstraint rc = new RequiredConstraint(l, listOfCrns);  if (rc.isFulfilled()) {printSchedule(validPermutatedUniqueCourseTimeslotsList.get(i)); break; }//System.out.println(); count++;}  } |

To

|  |
| --- |
| for (int i=0; i < validPermutatedUniqueCourseTimeslotsList.size(); i++) {  ArrayList<Timeslot> l = validPermutatedUniqueCourseTimeslotsList.get(i);  RequiredConstraint rc = new RequiredConstraint(l, listOfCrns);  System.out.println("slot 0 fulfilled: " + rc.isFulfilled());  if (rc.isFulfilled()) {printSchedule(validPermutatedUniqueCourseTimeslotsList.get(i)); break; }//System.out.println(); count++;}  }  System.out.println(count); |

Add

|  |
| --- |
| if(depth == allTimeslots.size())  {  result.add(current);  return;  }    for(int i = 0; i < Lists.get(depth).size(); ++i)  {  GeneratePermutations(Lists, result, depth + 1, current + Lists.get(depth).get(i));  } |

From

|  |
| --- |
| System.out.println(" |-------------------------| ");  System.out.println(" | Visualized timetable | ");  System.out.println("|------------|------------|------------|------------|------------|------------|------------|------------|");  System.out.println("|Time |Monday |Tuesday |Wednesday |Thursday |Friday |Saturday |Sunday |"); |

To

|  |
| --- |
| System.out.println(" |-------------------------| ");  System.out.println(" | Visualized timetable | ");  System.out.println("|------------|------------|------------|------------|------------|------------|------------|------------|");  System.out.println("|Time |Monday |Tuesday |Wednesday |Thursday |Friday |Saturday |Sunday |");  System.out.println("|------------|------------|------------|------------|------------|------------|------------|------------|"); |

From

|  |
| --- |
| if (j == timeslots.get(k).getDay() && timeslots.get(k).getStartTime() <= startTime && timeslots.get(k).getFinishTime() >= startTime+1)  {  System.out.print(timeslots.get(k).getCode() + "-" + timeslots.get(k).getSession() + " |");  filled = true;  } |

To

|  |
| --- |
| if (j == timeslots.get(k).getDay() && timeslots.get(k).getStartTime() <= startTime && timeslots.get(k).getFinishTime() >= startTime+1)  {  System.out.print(timeslots.get(k).getCrn() + " |");  System.out.print(timeslots.get(k).getCode() + "-" + timeslots.get(k).getSession() + " |");  filled = true;  } |

Timeslot.java

Add

|  |
| --- |
| public boolean finishBefore(Timeslot a) {  return this.finishTime < a.finishTime;  } |