Nurse Scheduling Project Data Structure

Generated by Doxygen 1.12.0

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

Class Documentation

3.1 Nurse Struct Reference

Structure representing a nurse's data.

`#include <NurseList.h>`

Public Member Functions

- bool **operator==** (const Nurse &other) const

Public Attributes

- std::string **fullName**
  Full name of the nurse.
- int **nurseNumber**
  Unique identifier for the nurse.
- std::string **nurseType**
  Type of nurse (e.g., "RN", "LPN")
- std::string **department**
  Department of the nurse (e.g., "Oncology")
- std::vector< int > **shiftPreferences**
  Stores preferences for 42 shifts (0, 1, 2)
- std::vector< int > **scheduledShifts**
  Stores what shifts a nurse is scheduled for.

3.1.1 Detailed Description

Structure representing a nurse's data.

This structure holds information about a nurse, including their full name, nurse number, type, department, and shift preferences.

The documentation for this struct was generated from the following file:

- NurseList.h

Chapter 4

File Documentation

4.1 CSVParser.h

```
00001 #ifndef CSV_PARSER_H
00002 #define CSV_PARSER_H
00003
00004 #include <string>
00005
00011 void parseNursesCSV(const std::string& fileName);
00012
00018 void parseConstraintsCSV(const std::string& fileName);
00019
00020 #endif // CSV_PARSER_H
```

4.2 NurseFunctions.h

```
00001 #ifndef NURSE_FUNCTIONS_H
00002 #define NURSE_FUNCTIONS_H
00003
00004 #include <string>
00005 #include "json.hpp"
00006 #include "NurseList.h"
00007 #include <fstream>
00008 using json = nlohmann::json;
00009
00018 void viewNursesByDepartmentAndType(const std::string& department, const std::string& type);
00019
00020 void viewNursesByDepartmentAndTypeJSON(const std::string& department, const std::string& type);
00021
00022 using ShiftSchedule = std::vector<std::vector<Nurse»;
00023
00033 Nurse getRandomNurseFromShift(const ShiftSchedule& schedule, int shift, const std::string& nurseType);
00034
00035 void printShiftSchedule(const ShiftSchedule& schedule);
00036
00047 void remove(ShiftSchedule& schedule, int shift, const Nurse& nurse);
00048
00059 void add(ShiftSchedule& schedule, int shift, const Nurse& nurse);
00060
00072 void shiftScheduleToJSON(const ShiftSchedule& schedule, const std::string& filename);
00073 #endif // NURSE_FUNCTIONS_H
00074
00081 void printNursesForShift(const ShiftSchedule& schedule, int shift);
00082
00096 void returnBestSatisfactionScores(
00097     int scoreGeneticAlgorithm, const ShiftSchedule& scheduleGeneticAlgorithm,
00098     int scoreBruteForce, const ShiftSchedule& scheduleBruteForce,
00099     int scoreLinearProgramming, const ShiftSchedule& scheduleLinearProgramming
00100 );
00101
00108 int calculateTotalShiftPreferences(const ShiftSchedule& shiftSchedule);
```

4.3 NurseList.h

```
00001 #ifndef NURSE_LIST_H
00002 #define NURSE_LIST_H
00003
00004 #include <string>
00005 #include <vector>
00006 #include <algorithm>  // For std::remove
00007 #include <iostream>
00008 #include <unordered_map>
00009
00016 struct Nurse {
00017     std::string fullName;
00018     int nurseNumber;
00019     std::string nurseType;
00020     std::string department;
00021     std::vector<int> shiftPreferences;
00022     std::vector<int> scheduledShifts;
00023
00024     // Equality operator to compare nurses by nurseNumber for removal
00025     bool operator==(const Nurse& other) const {
00026         return nurseNumber == other.nurseNumber;
00027     }
00028
00029 };
00030
00031 // Fake nurse for a shift unable to be scheduled
00032 // Declare a global instance of Nurse
00033 extern Nurse fakeNurse;
00034
00035 // Alias for the shift schedule: Vector of 42 vectors of Nurses
00036 using ShiftSchedule = std::vector<std::vector<Nurse»;
00037
00038 void add(ShiftSchedule& schedule, int shift, const Nurse& nurse);
00039 void remove(ShiftSchedule& schedule, int shift, const Nurse& nurse);
00040
00041 // Global variables
00042 extern std::unordered_map<std::string, std::unordered_map<std::string, std::vector<Nurse»>
      departmentNursesMap;
00043 extern std::unordered_map<int, std::unordered_map<std::string, std::unordered_map<std::string, int»>
      constraintsMap;
00044
00045 // Satisfaction scores for different algorithms
00046 extern int satisfactionScoreGeneticAlgorithm;
00047 extern int satisfactionScoreBruteForce;
00048 extern int satisfactionScoreLinearProgramming;
00049
00050 #endif // NURSE_LIST_H
```

Index