|  |  |
| --- | --- |
| **Programming II**  Diploma in IT / DS / CSF  Year 1 (2021/22) Semester 2 | Week  **8** |
| 1 hour |
| **Practical 8: Class Associations (Part 2)** | |

**Objectives**

At the end of this practical, the students should be able to:

* implement Class associations

|  |
| --- |
| **IMPORTANT**   * Create a folder, **week9.** * Create a new Console App (.NET Core) project, **Snnnnnnnn\_HospitalApp**, in the **Week9** folder created above *(note:* ***Snnnnnnnn*** *is your Student Number)*. * At the end of the session, copy the folder **Week08** folder (which contains all your work) to PRG2 network folder: [**\\ictspace.ict.np.edu.sg\PRG2**](file://ictspace.ict.np.edu.sg/PRG2) |

1. Implement all the classes in the following diagram, and the associations between the **Doctor**, **Patient**, and **Room** classes.

|  |
| --- |
| ***Person*** |
| -nric: string  -name: string |
| +Person(string,string)  +ToString():string |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Doctor** |  | **Patient** |  |  |
| -department: string  inChargeOf  1 0..\*  -patientList: List<Patient> |  | -wardedAt: Room |  |  |
| +Doctor(string ,string, string)  +AddPatient(Patient)  +RemovePatient(Patient)  +ToString():string |  | +Patient(string ,string, Room)  +ToString():string |  |  |

staysIn

1 1

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **Room** |
|  |  |  | -location: string  -wardClass: string |
|  |  |  | +Room(string, string)  +ToString(): string |

1. In the *Program.cs* file, do the following:
2. Create 3 Lists, **roomList**, **patientList**, and **doctorList**, to store the respective objects.
3. Initialise rooms and doctors.
   1. Write the **InitData()** method to:

* Create 3 room objects with the information given below and add them into **roomList**, which stores a list of rooms.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Location:  Ward class: | #01-01  C | #02-02  B | #03-03  A | #04-04  A |

* + - Create doctor objects with the information given below and add them into the **doctorList**, which stores a list of doctors.

|  |  |  |  |
| --- | --- | --- | --- |
| Nric:  Name:  Department: | S1234567A  Tom  Pediatrics | S2345678A  Champ  Oncology | S3456789B  Terry  Cardiology |

* + - Method signature is

static void InitData(List<Room> roomList, List<Doctor> doctorList)

* 1. Call the **InitData()** method in the *Main()* method. Display the list.

1. Create patients records from text file.
2. Write the **CreatePatients()**method to perform the following:
   * + The data file *Patients.csv* available in MeL contains the information of a list of patients and the room that he/she stay: NRIC, name, and room location. Download the file from MeL.
     + The **CreatePatients()** method will read all the data from the file. Create the Patient objects with the room that they are staying in and add them to the **patientList**.

Note: You may like to write another method **SearchRoom()** to search and return the **Room** object with the given location from the **roomList**.

* + - Method signature is

static void CreatePatients(List<Patient> patientList, List<Room> roomList)

* 1. Call the **CreatePatients()** method in the *Main()* method. Display the patients.

1. Assign patients to doctors.
   1. Write **AssignPatientsToDoctors()**method to perform the following:
      * The data file *PatientsToDoctor.csv* available in MeL contains the information of the patients who have been assigned to doctors: patient NRIC, doctor NRIC. Download the file from MeL.

* The **AssignPatientsToDoctors()** method will read all the data from the file, add the patient to the doctor’s patient list by using the **AddPatient()** method.

Note: You may like to write the method **SearchDoctor()** to search and return the doctor object with the given NRIC in the **doctorList**, likewise for the patients.

* Method signature is

static void AssignPatientsToDoctors(List<Patient> patientList, List<Doctor> doctorList)

* 1. Call the **AssignPatientsToDoctors()** method in the *Main()* method. Display the names of all the patients under the care of each doctor.

1. Remove patient from doctor.
2. Write **RemovePatientFromDoctor()** method to perform the following:
   * Allow user to enter the NRIC of a patient who has been assigned to a doctor and remove the patient from the doctor’s patient list by using the **RemovePatient()** method.
   * Method signature is

static void RemovePatientFromDoctor(List<Doctor> doctorList)

1. Call the method **RemovePatientFromDoctor()** in the *Main()* method. Display the names of all the patients under the care of each doctor.
2. Add new patient
3. Write **AddNewPatient()** method to perform the following:
   * Allow user to enter the information of a new patient (NRIC, name, and location of room warded).
   * Create the patient object and add into the **patientList**.
   * Append the data of the patient to the data file “Patients.csv”.
   * Method signature is

static void AddNewPatient(List<Patient> patientList, List<Room> roomList)

1. Call the **AddNewPatient()** method in the *Main()* method.
2. Open your data file, ensure that the new patient is added to the file.

**Plagiarism Warning:**

**If a student is found to have submitted work not done by him/her, he/she will not be awarded any marks for this practical. Disciplinary action may also be taken.**

**Similar action will be taken for student who allows other student(s) to copy his/her work.**