

Assignment III Report

Number :21627485

In this assignment what we need hospital system and we need to solve this problem with inheritance and polymorphism methods. We are expected to our program work with data access object pattern and design pattern.

```

classDiagram
    class Admission {
        <<Java Class>>
        +String examinationType
        +String operation
        +String admissionID
        +int patientID
        +Admission(int, int)
        +Admission(String, String)
        +getAdmissionID() int
        +setAdmissionID(int) void
        +getPatientID() int
        +setPatientID(int) void
        +getExaminationType() String
        +setExaminationType(String) void
        +getOperation() String
        +setOperation(String) void
        +read(Scanner, ArrayList<Admission>) void
    }
    class Examination {
        <<Java Interface>>
        +Examination
        +cost() int
    }
    class Patient {
        <<Java Class>>
        +Patient
        +int patientId
        +String patientName
        +String patientSurname
        +String patientAddress
        +String patientPhone
        +Patient(int, String, String, String, String)
        +getPatientId() int
        +setPatientId(int) void
        +getPatientName() String
        +setPatientName(String) void
        +getPatientSurname() String
        +setPatientSurname(String) void
        +getPatientAddress() String
        +setPatientAddress(String) void
        +getPatientPhone() String
        +setPatientPhone(String) void
        +read(Scanner, ArrayList<Patient>) void
    }
    class Operation {
        <<Java Class>>
        +Operation
        +Operation()
    }
    class Inpatient {
        <<Java Class>>
        +Inpatient
        +Inpatient()
        +cost() int
    }
    class Outpatient {
        <<Java Class>>
        +Outpatient
        +Outpatient()
        +cost() int
    }
    class Measurement {
        <<Java Class>>
        +Measurement
        +Measurement()
        +cost() int
    }
    class Test {
        <<Java Class>>
        +Test
        +Test()
        +cost() int
    }
    class Imaging {
        <<Java Class>>
        +Imaging
        +Imaging()
        +cost() int
    }
    class Program {
        <<Java Class>>
        +Program
        +temp: String
        +temp2: String
        +Program(String)
        +Program(String, String)
        +getTemp2() String
        +setTemp2(String) void
        +getTemp() String
        +setTemp(String) void
        +work(String) void
        +read(Scanner, ArrayList<Program>) void
    }
    class Total {
        <<Java Class>>
        +Total
        +patientId: int
        +examination: String
        +operation: String
        +total: int
        +Total(int, String, String, int)
        +getPatientId() int
        +setPatientId(int) void
        +getExamination() String
        +setExamination(String) void
        +getOperation() String
        +setOperation(String) void
        +getTotal() int
        +setTotal(int) void
        +calculate(String, String, int) int
    }
    class Main {
        <<Java Class>>
        +Main
        +Main()
        +main(String[]) void
    }

    Admission <|-- Examination
    Patient <|-- Operation
    Examination <|-- Inpatient
    Examination <|-- Outpatient
    Operation <|-- Measurement
    Measurement <|-- Test
    Test <|-- Imaging
    Examination ..> Patient
    Operation ..> Patient
    Measurement ..> Patient
    Test ..> Patient
    Imaging ..> Patient
    Program --> Admission
    Total --> Examination
    Main --> Total
  
```

```

if(emk.getTemp().equals("RemovePatient")) { //remove patient function delete patient
    int index=0;
    for(Patient emj:array) {
        if(emk.getTemp2().equals(Integer.toString(emj.getPatientId()))){
            System.out.println(emj.getPatientId());
            System.out.println("Patient "+emj.getPatientId()+" "+emj.getPatientName()+" removed");
            break;
        }
        index++;
    }
    array.remove(index);
}

```

1.Admission class

This class is read and hold admission.txt elements.

2.Patient class

This class just read and hold patient.txt elements.

3.Examination class

This class is my interface class.

4.Operation class

This class is my abstract class.

5.Imaging class

This class is return imaging fee.

6.Test class

This class is return test fee.

7.Measurement class

This class is return measurement fee.

8.Outpatient class

This class is return outpatient fee.

9.Inpatient class

This class is return inpatient fee.

10.Main class

This class is just run work function in Program class.

11.Total class

This class is calculate total fee.

12.Program class

This class is read input file, run fee function, list patient, print text and run other function what we need.