|  |
| --- |
| Implement class named MedicationTreatmentPlan.java and SurgeryTreatmentPlan.java that implements TreatmentPlan.java. These classes are responsible for generating treatment plan for the patients. |

**Task Explanation**

Your task involves creating two classes, MedicationTreatmentPlan.java and SurgeryTreatmentPlan.java, both of which implement the TreatmentPlan interface. These classes will be responsible for generating treatment plans for patients.

Let’s go step by step:

**1. Interface: TreatmentPlan.java**

This interface will define a method for creating a treatment plan. Every treatment plan (whether it’s surgery or medication) should implement this method.

**java**

package com.algonquin.cst8288.fall24.assignment1.treatment;

import com.algonquin.cst8288.fall24.assignment1.patient.Patient;

public interface TreatmentPlan {

String createTreatmentPlan(Patient patient);

}

**2. MedicationTreatmentPlan.java**

The MedicationTreatmentPlan class will implement the TreatmentPlan interface. The business logic here is based on the rules provided:

* **CHILD**: 1 dose per day.
* **YOUTH**: 2 doses per day.
* **ADULT**: 3 doses per day.
* Different durations based on the type of illness: **Acute**, **Infection**, **Chronic**.

java

package com.algonquin.cst8288.fall24.assignment1.treatment;

import com.algonquin.cst8288.fall24.assignment1.patient.Patient;

public class MedicationTreatmentPlan implements TreatmentPlan {

@Override

public String createTreatmentPlan(Patient patient) {

String lifeStage = patient.getLifeStage();

int dosesPerDay;

// Determine doses per day based on life stage

switch (lifeStage) {

case "CHILD":

dosesPerDay = 1;

break;

case "YOUTH":

dosesPerDay = 2;

break;

case "ADULT":

default:

dosesPerDay = 3;

break;

}

// Determine duration based on illness

String illnessType = patient.getPlannedTreatment();

int duration;

String medicationType;

switch (illnessType) {

case "Acute":

duration = 7;

medicationType = "No Medication";

break;

case "Infection":

duration = 14;

medicationType = "Antibiotics";

break;

case "Chronic":

default:

duration = 180;

medicationType = "Specialized Medication";

break;

}

// Return the treatment plan details as a string

return String.format("Medication Treatment Plan:\n- Doses per day: %d\n- Duration: %d days\n- Medication: %s",

dosesPerDay, duration, medicationType);

}

}

**3. SurgeryTreatmentPlan.java**

The SurgeryTreatmentPlan class will also implement the TreatmentPlan interface. The business logic is based on the surgery-specific details:

* Different fasting times based on age.
* Follow-up schedules based on age.

**java**

package com.algonquin.cst8288.fall24.assignment1.treatment;

import com.algonquin.cst8288.fall24.assignment1.patient.Patient;

public class SurgeryTreatmentPlan implements TreatmentPlan {

@Override

public String createTreatmentPlan(Patient patient) {

long age = patient.getAge();

String fastingRequirement;

String followUpSchedule;

// Determine fasting requirements and follow-up based on age

if (age < 6) {

fastingRequirement = "No Fasting Required";

followUpSchedule = "Next Day";

} else if (age <= 18) {

fastingRequirement = "2 Hours Before Surgery";

followUpSchedule = "In a Week";

} else {

fastingRequirement = "8 Hours Before Surgery";

followUpSchedule = "In a Month";

}

// Return the treatment plan details as a string

return String.format("Surgery Treatment Plan:\n- Fasting Requirement: %s\n- Follow-up Schedule: %s",

fastingRequirement, followUpSchedule);

}

}

**Explanation of the Code**

1. **Interface (TreatmentPlan.java)**:
   * This is a contract that defines the method createTreatmentPlan(Patient patient). Both types of treatment plans (Medication and Surgery) must implement this method.
2. **MedicationTreatmentPlan.java**:
   * This class implements the TreatmentPlan interface and provides a medication-specific implementation of createTreatmentPlan().
   * It determines the dosage and duration of the treatment based on the patient's life stage and illness type.
3. **SurgeryTreatmentPlan.java**:
   * This class implements the TreatmentPlan interface and provides a surgery-specific implementation of createTreatmentPlan().
   * It determines fasting requirements and follow-up schedules based on the patient’s age.

**How This Fits into Your Overall System**

1. **PatientManagement.java**:
   * This class will use either MedicationTreatmentPlan or SurgeryTreatmentPlan to generate a treatment plan based on the type of patient.
2. **Physician.java**:
   * The Physician class will create instances of patients and then apply either a MedicationTreatmentPlan or SurgeryTreatmentPlan, as explained earlier.

**Next Steps**

* If you have specific rules or conditions that are different from what I described, you can modify the logic in the createTreatmentPlan() method accordingly.
* Test the system by running your Physician.java class and ensure that the treatment plans are generated and printed as expected.