ALGONQUIN COLLEGE

Assignment #1 – 24F

In this task, your objective is to apply **UML** and **SOLID principles** to a coding exercise. Your task is to apply SOLID principles to enhance the design and maintainability of the code. This may involve refactoring, implementing interfaces, or creating new classes. Your goal is to improve the codebase by making it more modular, extensible, and in accordance with SOLID principles. The provided code is consisting of several classes.

1. **Patient.java**: A base class representing a generic patient.
2. **Inpatient.java**: A subclass of Patient representing patients who are admitted to the hospital.
3. **Outpatient.java**: A subclass of Patient representing patients who visit the hospital but are not admitted.
4. **TreatmentPlan.java**: Interface defining the treatment plan for patient.
5. **PrescriptionService.java:** Interface defining prescription generation mechanism
6. **PatientManagement.java**: The controller class of the application.
7. **Physician**: The main program to execute the application

Part 1:

1. Examine the **PatientManagement.java** class to identify any violations of the **Single Responsibility Principle**. Refactor the class to adhere to this principle, considering the potential introduction of additional classes to maintain compliance.

2. Evaluate the **Patient.java, Inpatient.java** and **Outpatient.java** to identify any violation of the **Liskov Substitution Principle.** Refactor the corresponding class in order to comply with principle.

3. Evaluate **createTreatmentPlan()** and **generatePrescription()** method under **PatientManagement.java**. Identify how the high level modules are directly dependent on the low level module violating **Dependency Inversion Principle**. Refactor the codes using **TreatmentPlan.java** and **PrescriptionService.java** to align with the **Dependency Inversion Principle**.

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

ALGONQUIN COLLEGE

Assignment #1 – 24F

Following business rules are associated to the different treatment plans

Medication Treatment

a. CHILD – 1, YOUTH – 2, and ADULT – 3 times per day
b. Duration of doses
    i. Acute– 7 days (No Medication)
    ii. Infection – 14 days (Antibiotics)
    iii. Chronic– 180 days (Specialized Medication)

Surgery Treatment

a. Date and Time of the Surgery
b.

Age	Risk Factor	Fasting	Follow Up
Age < 6	High	No Fasting	Next Day
6 =< Age =< 18	Medium	2 Hours Before	In a Week
Age > 18	Low	8 Hours Before	In a Month

6. Implement classes named **OnlinePrescriptionSevice.java** and **PrintablePrescriptionService.java** that implements the **PrescriptionService.java**. These classes are responsible for generating prescription in different formats (.html and .txt)

7. Implement **Physician.java** class to simulate complete solution. In order to do so, instantiate two **Patient** objects and Utilize **PatientManagement.java** to create treatment plan, generate prescriptions and other required data. Save the prescriptions in both online and printable formats in the files named **patient_name_presciption.html** and **patient_name_prescription.txt**.

ALGONQUIN COLLEGE

Assignment #1 – 24F

Part 2: Develop JUnit tests to verify the business rules you implemented in different classes

Part 3: Finally, generate a report (limited to 2 pages) comprising a UML Class Diagram, outlining the purpose of each newly introduced class and detailing the application of SOLID principles in your solution.

Deliverables

- This task constitutes 7.5% of your overall grade and is due by **September 22, 2024 at 11:59 PM.**
- Deliver a comprehensive coding solution.
- Provide the report outlined in part 3.
- Demo is mandatory. No demo results 0 for your submission.
- Your code should include proper commenting and coding practice according to Java standards.
- You must demo your solution during the lab session and submit your code and report on **Brightspace**
- Violating academic integrity or missing the deadline will result in a grade of 0 for your submission. No exception to this policy.