Consignas

Proyecto

Deberá crear un proyecto Java que admita pruebas (tests) con JUnit 4 y deberá nombrarse

- PB2-C02-3300-Q1-2024-TM-Recuperatorio-**ApellidosNombres** (debiendo reemplazar "**ApellidosNombres**" por su/s apellido/s y nombre/s).

Agregue los paquetes (packages) necesarios según la convención de Java y desarrolle un proyecto aplicando TDD (Desarrollo guiado por pruebas) para el siguiente enunciado.

Enunciado

Atención a pacientes de hospital

Debemos desarrollar un producto software que permita gestionar doctores, pacientes y atenciones para un hospital.
Dicho hospital cuenta con doctores de los cuales conocemos su DNI, nombres, apellidos, la fecha de nacimiento y su matrícula (conocemos que las matrículas actualmente empiezan en 90000). Los doctores deben mostrarse ordenados por su matrícula de manera ascendente y sin duplicados.
Además de poseer doctores, el hospital debe contar con un registro de pacientes. Para ellos solicitamos: legajo (número de uso interno), nombres, apellidos, DNI y fecha de nacimiento.
Cada paciente puede recibir asistencia por obra social, por su afiliación directa al hospital, o de manera particular. A los pacientes que llegan de manera particular se les cobra el valor completo de la atención: $1000. Los pacientes que asisten por obra social, deben presentar su número de tarjeta (Ejemplo: 1234-4567-8901) y se les descuenta $200 del valor de la atención. Respecto a los pacientes afiliados, cuentan con un porcentaje de descuento, el cual inicialmente es del 50%.
Se debe permitir ajustar estos valores de bono, porcentaje y precio de la asistencia en un futuro.
No podemos permitir tener pacientes con el mismo DNI.
En el hospital se brindarán atenciones. Cada atención debe contar con la información del doctor que atendió, el paciente (según aplique), quién nos deberá indicar el motivo de su visita, y un precio (calculado según lo visto anteriormente). Tener en cuenta que el valor de la atención lo establece el hospital y que se le asigna un número a cada atención.
Los doctores pueden diagnosticar atenciones, si y sólo si, esa atención le pertenece. Es necesario controlar esta tarea para no permitir que se realice.
El sistema debe permitir consultar atenciones realizadas a pacientes de obra social, según un doctor en específico. Esto nos ayuda a revisar las asignaciones de atenciones a cada doctor. También debe permitir obtener un reporte de facturación total de atenciones para cada doctor debiendo ordenarse por la matrícula descendente. Ejemplo:

- Doctor3 (Matrícula 96231): 5000.

- Doctor1 (Matrícula 86231): 9000.
- Doctor2 (Matrícula 76231): 11000.

Tests a desarrollar

1. **Test:** dadoQueExisteUnHospitalAlAgregarUnDoctorObtengoVerdadero().
2. **Test:** dadoQueExisteUnHospitalConDoctoresAlAgregarUnDoctorConMatriculaDuplicadaSeLanzaUnaMatriculaDuplicadaException().
3. **Test:** dadoQueExisteUnHospitalConDoctoresCuandoLosConosultoObtengoUnaColeccionDeDoctoresOrdenadasPorMatriculaAscendente().
4. **Test:** dadoQueExisteUnHospitalAlAgregarUnPacienteAfiliadoObtengoVerdadero().
5. **Test:** dadoQueExisteUnHospitalConDoctoresYPacientesCuandoQuieroDiagnosticarUnaAtencionDeUnDoctorObtengoVerdaderoYElDiagnosticoEnLaAtencionEstaActualizado()
6. **Test:** dadoQueExisteUnHospitalConDoctoresYPacientesCuandoQuieroDiagnosticarUnaAtencionQueNoEsDelDoctorSeLanzaUnaTareaNoPermitidaException().
7. **Test:** dadoQueExisteUnHospitalConDoctoresPacientesYAtencionesCuandoQuieroVerLasAtencionesAPacientesConObraSocialPorDoctorObtengoLasAtencionesParaEseDoctorDePacientesConObraSocial().
8. **Test:** dadoQueExisteUnHospitalCuandoConsultoElPrecioDeUnaAtencionParaUnPacienteAfiliadoObtengoElTotal500().
9. **Test:** dadoQueExisteUnHospitalCuandoConsultoLaFacturacionDeCadaDoctorObtengoUnMapaPorDoctorYTotalOrdenadoPorMatriculaDeDoctorDescendente()

Deberá incluir los siguientes métodos:

- boolean diagnosticarAtencion(Long idAtencion, Long matriculaDoctor, String diagnostico)
- List<Atencion> obtenerAtencionesDePacientesConObraSocialAtendidasPorUnDoctor(Long matriculaDoctor).
- Double obtenerPrecioDeLaAtencion(Long legajo).
- Map<Doctor, Double> obtenerFacturacionDeAtencionesPorDoctor().

Puede agregar los métodos que considere necesarios para cumplir con los tests y métodos indicados.

Entrega

Se deberá comprimir el proyecto y entregarlo en la plataforma MIeL, en la práctica determinada para tal fin.