DOCUMENTACION PARCIAL PROGRAMACION II

ALUMNO: Erlan Gabrelach Mauro Leonel

Descripción del Ejercicio

El ejercicio propone la implementación de un sistema informático para administrar una clínica médica local. El sistema debe gestionar información sobre profesionales médicos, pacientes, especialidades médicas y recursos de la clínica. Además, se requieren diversas funcionalidades para realizar consultas y obtener información específica.

Clases

Persona (Super-Clase "Clase Abstracta")

La clase Persona es una superclase abstracta que representa a las personas en el sistema médico. Contiene atributos comunes como nombre, apellido, edad y estado de activo/inactivo.

* nombre: El nombre de la persona.
* apellido: El apellido de la persona.
* edad: La edad de la persona.
* activo: Indica si la persona está activa o inactiva en el sistema.

Profesional (Sub-Clase de Persona)

La clase Profesional es una subclase de Persona que representa a los profesionales médicos que trabajan en la clínica. Además de los atributos heredados de Persona, tiene atributos adicionales como la especialidad médica y el nombre de la secretaria que agendó el turno.

* especialidad: La especialidad médica del profesional.
* secretaria: El nombre de la secretaria que agendó el turno para el profesional.

Paciente (Sub-Clase de Persona)

La clase Paciente es una subclase de Persona que representa a los pacientes atendidos en la clínica. Además de los atributos heredados de Persona, tiene un atributo adicional que indica la obra social a la que está afiliado el paciente.

* obraSocial: La obra social a la que está afiliado el paciente.
* Especialidad

La clase Especialidad describe las especialidades médicas disponibles en la clínica. Cada especialidad tiene un código y un nombre.

* codigo: El código de identificación de la especialidad.
* nombre: El nombre de la especialidad médica.

Recurso

La clase Recurso representa los recursos disponibles en la clínica que deben ser registrados, cuantificados y clasificados. Los recursos pueden incluir computadoras, muebles de oficina, televisores, sillas, camas de una plaza y camas de dos plazas. Cada recurso tiene un nombre, una cantidad y una categoría.

* nombre: El nombre del recurso.
* cantidad: La cantidad disponible de ese recurso.
* categoria: La categoría a la que pertenece el recurso (por ejemplo, computadoras, camas, televisores, etc.).

Funcionalidades

El ejercicio se divide en varios pasos (Ejercicio/Paso 1, Ejercicio/Paso 2, etc.) que corresponden a las diferentes funcionalidades que se deben implementar en el sistema. A continuación, se describe cada paso con su funcionalidad correspondiente:

Ejercicio/Paso 1

En este paso, se definen las clases que se utilizarán en los siguientes ejercicios/pasos. Se crean las clases Persona, Profesional, Paciente, Especialidad y Recurso con sus respectivos atributos.

Ejercicio/Paso 2

En este paso, se solicita listar los pacientes del día martes 12 de septiembre de 2023 en la especialidad de Radiología. Los pacientes deben mostrarse ordenados por orden de llegada, y se debe incluir el número de turno, el apellido y nombre del paciente, y el nombre de la secretaria que agendó el turno.

Ejercicio/Paso 3

En este paso, se solicita listar los turnos por Profesional y Especialidad. Los turnos deben ordenarse dentro de cada especialidad por orden de llegada, y se debe mostrar el nombre del secretario que agendó el turno.

Ejercicio/Paso 4

En este paso, se solicita listar los pacientes activos mayores de 65 años que están afiliados a la obra social INSSSEP. Para considerar a un paciente como activo o inactivo, se utiliza un marcador. Se deben mostrar los nombres de los pacientes, sus edades y la obra social a la que están afiliados.

Ejercicio/Paso 5

En este paso, se solicita determinar cuántas camas de una plaza y televisores hay en el edificio de la clínica. La cantidad de estos recursos se obtiene contando los recursos disponibles en la categoría correspondiente.

Implementación

El código del ejercicio muestra una implementación básica en Java de las clases y métodos necesarios para llevar a cabo las funcionalidades descritas en cada paso.

Se utilizan las clases List y stream() para filtrar y procesar datos de manera eficiente. Las funciones filter() y collect() se utilizan para filtrar y recopilar los resultados de las consultas.

Cada paso incluye un ejemplo de cómo se debe realizar la consulta o acción solicitada y cómo se debe mostrar la información resultante.