Yaneth Mejía Rendón
PROGRAMACIÓN II
Universidad de Caldas - I SEM 2024

TALLER CALIFICABLE. SÓLO SE RECIBE HASTA EL 02 DE ABRIL MEDIODÍA

1. **Objetivo: Implementar un programa que gestione el registro de estudiantes en una universidad con funcionalidades de búsqueda, filtrado y eliminación.**

Se debe crear una clase Estudiante con los siguientes atributos:

nombre (String): Nombre completo del estudiante.
codigo (String): Código único de identificación del estudiante.
carrera (String): Carrera que estudia el estudiante.
promedio (double): Promedio académico del estudiante.

Se debe crear una clase RegistroEstudiantes con los siguientes métodos:

a. agregarEstudiante(Estudiante estudiante): Agrega un estudiante al registro. Este método debe realizar las mismas validaciones que el ejercicio anterior.
b. eliminarEstudiante(String codigo): Elimina el estudiante que coincida con el código especificado.
c. listarEstudiantes(): Imprime en la consola la información de todos los estudiantes registrados.
d. buscarEstudiantes(String filtro): Busca y retorna una lista de estudiantes que coincidan con el filtro especificado. El filtro puede ser una cadena que se compara con el nombre, el código o la carrera del estudiante. Cualquiera de los 3 filtros se deben permitir.

2. **Objetivo: Implementar un sistema de gestión de biblioteca que permita registrar libros, autores, préstamos y devoluciones.**

Se debe crear una clase Libro con los siguientes atributos:

titulo (String): Título del libro.
autor (Autor): Autor del libro.
isbn (String): ISBN del libro.
categoria (String): Categoría del libro.
ejemplares (int): Número de ejemplares disponibles del libro.

Se debe crear una clase Autor con los siguientes atributos:

nombre (String): Nombre completo del autor.
nacionalidad (String): Nacionalidad del autor.

Se debe crear una clase Prestamo con los siguientes atributos:

libro (Libro): Libro prestado.
usuario (Usuario): Usuario que ha prestado el libro.
fechaPrestamo (Date): Fecha en la que se realizó el préstamo.
fechaDevolucion (Date): Fecha en la que se debe devolver el libro.

Se debe crear una clase Usuario con los siguientes atributos:

nombre (String): Nombre completo del usuario.
cedula (String): Cédula de identidad del usuario.

Se debe crear una clase RegistroBiblioteca con los siguientes métodos:

a. registrarLibro(Libro libro): Agrega un libro al registro. Este método debe realizar las mismas validaciones que el ejercicio anterior.
b. registrarAutor(Autor autor): Agrega un autor al registro.
c. registrarPrestamo(Prestamo prestamo): Registra un préstamo de un libro a un usuario. Este método debe realizar las siguientes validaciones:

El libro debe tener ejemplares disponibles.
El usuario no debe tener préstamos vencidos.

d. registrarDevolucion(Prestamo prestamo): Registra la devolución de un libro.

e. listarLibros(): Imprime en la consola la información de todos los libros registrados.
f. listarAutores(): Imprime en la consola la información de todos los autores registrados.
g. listarPrestamos(): Imprime en la consola la información de todos los préstamos activos.
h. buscarLibros(String filtro): Busca y retorna una lista de libros que coincidan con el filtro especificado. El filtro puede ser una cadena que se compara con el título, el autor, el ISBN o la categoría del libro.

3. Objetivo: Implementar un sistema de gestión de inventario para una tienda que permita registrar productos, categorías, proveedores y ventas.

Se debe crear una clase Producto con los siguientes atributos:
nombre (String): Nombre del producto.
categoria (Categoria): Categoría del producto.
precio (double): Precio de venta del producto.
stock (int): Cantidad de unidades del producto en stock.

Se debe crear una clase Categoria con los siguientes atributos:
nombre (String): Nombre de la categoría.
descripcion (String): Descripción de la categoría.

Se debe crear una clase Proveedor con los siguientes atributos:
nombre (String): Nombre del proveedor.
telefono (String): Teléfono del proveedor.
direccion (String): Dirección del proveedor.

Se debe crear una clase Venta con los siguientes atributos:
producto (Producto): Producto vendido.
cantidad (int): Cantidad de unidades del producto vendidas.
fechaVenta (Date): Fecha en la que se realizó la venta.

Se debe crear una clase RegistroInventario con los siguientes métodos:
a. registrarProducto(Producto producto): Agrega un producto al registro. Este método debe realizar las mismas validaciones que el ejercicio anterior.
b. registrarCategoria(Categoria categoria): Agrega una categoría al registro.
c. registrarProveedor(Proveedor proveedor): Agrega un proveedor al registro.
d. registrarVenta(Venta venta): Registra una venta de un producto. Este método debe realizar las siguientes validaciones:

El producto debe tener stock disponible.
a. listarProductos(): Imprime en la consola la información de todos los productos registrados.
b. listarCategorias(): Imprime en la consola la información de todas las categorías registradas
c. listarProveedores(): Imprime en la consola la información de todos los proveedores registrados.
d. listarVentas(): Imprime en la consola la información de todas las ventas registradas.
e. buscarProductos(String filtro): Busca y retorna una lista de productos que coincidan con el filtro especificado. El filtro puede ser una cadena que se compara con el nombre, la categoría o el precio del producto.

4. Objetivo: Implementar un sistema de gestión de citas médicas para una clínica que permita registrar pacientes, médicos, especialidades y citas.

Se debe crear una clase Paciente con los siguientes atributos:
nombre (String): Nombre completo del paciente.
cedula (String): Cédula de identidad del paciente.
telefono (String): Teléfono del paciente.
direccion (String): Dirección del paciente.

Yaneth Mejía Rendón

PROGRAMACIÓN II

Universidad de Caldas - I SEM 2024

Se debe crear una clase Medico con los siguientes atributos:

nombre (String): Nombre completo del médico.
especialidad (Especialidad): Especialidad del médico.
codigoMedico (String): Código único de identificación del médico.

Se debe crear una clase Especialidad con los siguientes atributos:

nombre (String): Nombre de la especialidad.
descripcion (String): Descripción de la especialidad.

Se debe crear una clase Cita con los siguientes atributos:

paciente (Paciente): Paciente que tiene la cita.
medico (Medico): Médico que atiende la cita.
fechaCita (Date): Fecha de la cita.
horaCita (String): Hora de la cita.
estado (String): Estado de la cita (pendiente, confirmada, cancelada).

Se debe crear una clase RegistroCitas con los siguientes métodos:

a. registrarPaciente(Paciente paciente): Agrega un paciente al registro. Este método debe realizar las mismas validaciones que el ejercicio anterior.
b. registrarMedico(Medico medico): Agrega un médico al registro.
c. registrarEspecialidad(Especialidad especialidad): Agrega una especialidad al registro.
d. registrarCita(Cita cita): Registra una cita médica. Este método debe realizar las siguientes validaciones:
    i. El médico debe tener disponibilidad en la fecha y hora de la cita.
    ii. El paciente no debe tener otra cita en la misma fecha y hora.
e. listarPacientes(): Imprime en la consola la información de todos los pacientes registrados.
f. listarMedicos(): Imprime en la consola la información de todos los médicos registrados.
g. listarEspecialidades(): Imprime en la consola la información de todas las especialidades registradas
h. listarCitas(): Imprime en la consola la información de todas las citas registradas.
i. buscarCitas(String filtro): Busca y retorna una lista de citas que coincidan con el filtro especificado. El filtro puede ser una cadena que se compara con el nombre del paciente, el médico, la especialidad o la fecha de la cita.