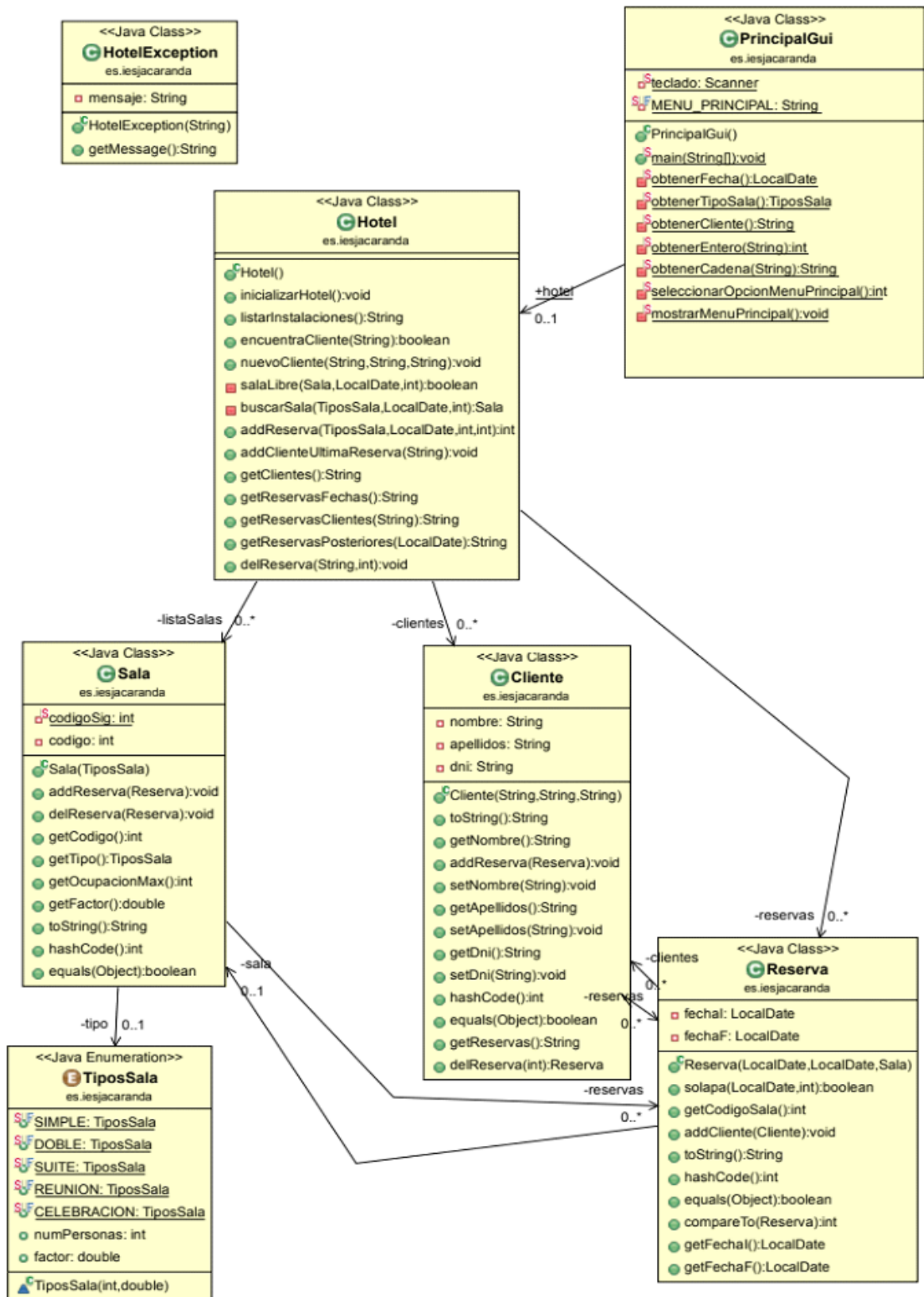


HOTEL JACARANDÁ



(quedan por añadir las opciones de la 3 a la 7)

HOTEL JACARANDÁ

Vamos a realizar un programa un poco nuestro hotel, el menú principal será:

1. Listar todas las estancias.
2. Reservar una estancia.
3. Mostrar todas las reservas ordenadas por fecha.
4. Mostrar todos los clientes del hotel.
5. Mostrar todas las reservas de un cliente.
6. Mostrar las reservas posteriores a una fecha.
7. Anular una reserva
8. Salir

Vamos a explicar un poco como funciona la opción 2 y la opción 7:

Si queremos hacer una reserva nos pedirá la fecha inicial de la reserva, el número de días, el tipo de habitación y el número de personas, si es necesario. Una vez que vemos que hay disponibilidad le pediremos el dni de la persona responsable o de las personas si son dos en habitación doble o suite. Si el cliente no existe, se deberá solicitar el nombre y el apellido.

Vamos a crear un enumerado con el número máximo de Personas y el factor de cada tipo.

Las salas o estancias del hotel tendrán un código que se asignará de forma automática. A continuación se explican los métodos que no son básicos:

- Dos salas son iguales cuando tienen el mismo código.
- El toString de sala deberá devolver la sala en el siguiente formato:
Sala: codigoSala Tipo: tipoSala Ocupación Máxima: ocupacionMaxima
Reservas: listadeReservas.
- delReserva(Reserva): debe borrar la reserva que se la pasa por parámetro de la lista de reservas de las salas

Cliente. Dos clientes serán iguales cuando tengan el mismo dni.

- El toString de cliente deberá devolver el siguiente formato:
Cliente: Dni: dni, nombre apellidos.
- delReserva(int): borra la reserva que se encuentra en la posición int -1. Devolverá la reserva borrada y si no existe lanzará una exception.
- El método getReserva devolverá un String con las reservas con el siguiente formato:
1.- reserva.toString
2.- reserva.toString()
.....

Reservas. Dos reservas son iguales cuando tengan la misma sala y la misma fecha inicial y final.

- solapa(LocalDate, int): boolean
Este método devuelve un booleano que será true, si la reserva que llama a este método solapa con la fechas que se la pasa como argumento, teniendo en cuenta que el LocalDate es la fecha inicial y el entero es el número de días. Si no solapa devuelve false.
- El método toString tendrá el siguiente formato.

HOTEL JACARANDÁ

Reserva: Fecha Inicial: fechaI Fecha Final: fechaF Cliente: cliente.toString()
Sala: sala.getCodigo()

Hotel.

- encuentraCliente(String): boolean
Recibirá el dni del cliente y devolverá true si el cliente está en la lista de clientes del hotel, si no se encuentra en la lista de clientes devolverá false.
- nuevoCliente(String dni, String nombre, String apellidos): void
Recibe los datos del cliente y crea un nuevo cliente y lo almacena en la lista de clientes del hotel.
- addReserva(TiposSala, LocalDate fechaI, int numDias, int numeroPersonas) : int
Recibe los datos y tiene que añadir una reserva de ese tipo de sala. Deberá buscar una sala de ese tipo libre para esos días. Si no encuentra sala deberá lanzar una exception. Devolverá el número de dni que se deben pedir de los clientes.
- addClienteUltimaReserva(String dni): void
Añade el cliente con dni que recibe como argumento a la última reserva realizada.
- buscarSala(TiposSala, LocalDate fecha, in numDias): Sala
Método que devuelve una sala libre para las fechas que se piden, del tipo que se pide. Si no existe una sala libre lanza una exception.
- salaLibre(Sala, LocalDate, int): booleano
Devuelve true si la sala está libre en esas fechas
- getCientes(): devolverá una cadena con la lista de todos los clientes con el siguiente formato:
cliente().toString()
cliente().toString()

- getReservaClientes(String dni) devolverá una cadena con la lista de las reservas de ese cliente con el siguiente formato.
1.- reserva.toString
2.- reserva.toString()
.....
- getReservasPosteriores(LocalDate) devolverá una cadena con la lista de las reservas posteriores a la fecha que se le pasa como argumento. El listado será de la forma y deben salir ordenados por la fecha inicial.
reserva.toString
reserva.toString()
.....
- delReserva(String dni, int opc) borrará la reserva del cliente que tiene el dni que recibe como argumento que ocupa la posición opc -1. Si no existe la reserva se deberá lanzar una exception.