

EXAMEN PARCIAL ALGORITMOS

UML

El objetivo de este ejercicio es diseñar un diagrama UML para un sistema de gestión de hotel. Este sistema está diseñado para ayudar a los hoteles a gestionar sus habitaciones, empleados y servicios de manera eficiente.

Una vez completado el diagrama UML, guarda tu trabajo como un archivo .drawio y un archivo de imagen (PNG o JPEG).

El sistema de gestión hotelera está compuesto por tres entidades principales: Room (Habitación), Employee (Empleado) y Hotel. Cada una de estas entidades tiene sus propios atributos y métodos que permiten realizar operaciones específicas dentro del sistema.

Class Room:

Atributos

- Room_type
- Room_number
- Room_state
- Room_price

Métodos

- Constructor
- Getter y setters
- Is_occupied()
- Check_in()
- Check_out()

Class Employee:

Atributos

- Emp_id
- Name
- Position
- Salary

Métodos

- Constructor
- Getters y Setters

Class Hotel

Atributos

- Name
- Rooms (list)
- Employees (list)
- Reservations (dictionary)

Métodos

- Add_room
- Remove_room
- Add_employee
- Remove_employee
- Check_in
- Check_out
- Find_room

PROGRAMA

En el ámbito de la hospitalidad, los hoteles juegan un papel crucial en proporcionar alojamiento y servicios a los viajeros. Para mejorar la experiencia del huésped y optimizar la operación del hotel, se emplean sistemas de gestión hotelera.

En este ejercicio, vamos a desarrollar una versión programada en Python de un sistema de gestión de hotel. Este sistema permitirá a los usuarios gestionar habitaciones, empleados, y servicios ofrecidos por el hotel.

La aplicación programada en Python permitirá que los usuarios introduzcan la información de las habitaciones, empleados, y servicios que ofrece el hotel. Además, deberá poder gestionar reservaciones y el check-in/check-out de los huéspedes.

No es necesario implementar el docString correspondiente a las funciones y métodos desarrollados, aunque se recomienda hacer el diagrama de flujo de los métodos en papel de forma previa a su resolución.

Especificaciones:

- a) Programe la clase **Room**. Cada habitación estará caracterizada por:

- Tipo de habitación. En esta versión del sistema, solamente existirán 3 tipos de habitación (Individual, Doble, Suite).
- Número de habitación. Cada habitación tiene un número único.
- Estado de ocupación. Indica si la habitación está ocupada o no.
- Precio por noche. Un valor numérico que representa el costo de la habitación por noche.

Nota: Es obligatorio el uso de enumerados para los atributos del tipo de habitación.

Incluya los atributos de esta clase y establezca la visibilidad adecuada (público, privado, protegido). Añada cualquier atributo y/o método que considere necesario.

Programa un constructor que reciba los datos necesarios para crear una habitación. El método debe verificar el tipo y valor de cada uno de los parámetros y lanzar la excepción correspondiente cuando no se cumplan los requisitos.

Programa los métodos setters y getters para la clase en función de lo que necesite.

Programa el método `is_occupied(self)` de la clase `Room`. Este método sirve para saber si la habitación está ocupada.

Programa el método `check_in(self)` y `check_out(self)` para gestionar la ocupación de la habitación.

b) Programa una clase **Employee**. Cada empleado va a estar caracterizado por:

- Un identificador único (por ejemplo, número de empleado).
- Nombre del empleado.
- Posición que ocupa en el hotel (Recepcionista, Limpiador, Gerente, etc.).
- Lista de tareas asignadas.

Incluya los atributos de esta clase y establezca la visibilidad adecuada (público, privado, protegido). Añada cualquier atributo y/o método que considere necesario.

Programa un constructor que reciba los datos necesarios para crear un empleado. El método debe verificar el tipo y valor de cada uno de los parámetros y lanzar la excepción correspondiente cuando no se cumplan los requisitos.

Programa los métodos setters y getters para la clase en función de lo que necesite.

c) Programa una clase **Hotel**. El hotel estará caracterizado por:

- Nombre del hotel.

- Dirección del hotel.
- Lista de habitaciones.
- Lista de empleados.
- Servicios ofrecidos (piscina, restaurante, gimnasio, etc.).

Programa los métodos necesarios para añadir/eliminar habitaciones y empleados, y para gestionar las reservaciones y el check-in/check-out de los huéspedes.

Para ayudar en el desarrollo de este ejercicio, se facilitan los archivos vacíos donde deberían estar implementadas las clases que se piden y que tienen que ser completadas por el alumno. En dichas clases, están ya añadidos los casos de tests de cada una de ellas.

La clase main es el punto de entrada de la aplicación del sistema de gestión de hotel programado en Python. Esta clase será responsable de interactuar con el usuario, procesar las entradas y mostrar las salidas basadas en las operaciones realizadas en las clases Room, Employee, y Hotel. La implementación de esta clase debe permitir al usuario realizar las siguientes operaciones:

- Agregar habitaciones al hotel: Permite al usuario introducir la información de nuevas habitaciones y añadirlas a la lista de habitaciones del hotel.
- Agregar empleados al hotel: Permite al usuario introducir la información de nuevos empleados y añadirlos a la lista de empleados del hotel.
- Añadir servicios al hotel: El usuario puede introducir nuevos servicios que el hotel ofrece a sus huéspedes.
- Realizar reservas: El usuario puede reservar habitaciones para los huéspedes, especificando el tipo de habitación y las fechas de la estancia.
- Realizar check-in: Permite al usuario realizar el check-in de un huésped, asignando una habitación disponible según la reserva.
- Realizar check-out: Permite al usuario realizar el check-out de un huésped, liberando la habitación y calculando el costo de la estancia basado en el precio por noche y la duración de la estancia.

Para facilitar la interacción del usuario con el sistema, la clase main debería implementar un menú de consola con opciones para cada una de las operaciones descritas anteriormente. Cada opción del menú deberá llamar a los métodos correspondientes de las clases Room, Employee, y Hotel para realizar las operaciones requeridas.