

Segundo Parcial

Leé con cuidado el enunciado y por lo menos dos veces para resolver lo pedido. Pensá bien la estrategia de resolución antes de comenzar el desarrollo de lo que te solicitan. El objetivo de este examen es **evaluar la correcta aplicación de los conceptos y técnicas** vistos hasta el momento:

- Correcta definición de clases y asignación adecuada de sus responsabilidades.
- Encapsulamiento, ocultamiento de información y uso de getters y setters sólo cuando corresponda.
- Modularizarían reutilizable y mantenible con uso de métodos con correcta parametrización.
- Correcta aplicación de miembros de instancia y de clase.
- Correcto uso de colecciones.
- Correcta aplicación conceptual de las relaciones entre clases.

Enunciado

CowORTking, una empresa dedicada al coworking con múltiples sedes, desea implementar un sistema para gestionar la asistencia y las actividades de sus usuarios registrados.

De los usuarios se sabe el dni, su nombre completo y si pagó la cuota del mes.

Hay varias sedes distribuidas por la ciudad. Los usuarios registrados pueden acceder libremente a cualquiera de las sedes para hacer uso de sus servicios por un valor base mensual de \$15.000.

Cada sede tiene un nombre y ofrece diversos servicios. Cada uno de estos servicios tienen, a su vez, un nombre, un precio adicional individual (por ejemplo: Sala de reuniones: \$30.000) y, de estar reservado, quien lo reservó ó está usando (si está disponible será null).

Tener en cuenta que en cada sede cada servicio puede aparecer más de una vez (por ejemplo, una sede puede contar con tres salas con proyector).

Para reservar el servicio el usuario informa su dni y los nombres de la sede y el servicio que requiere. De ser posible se realiza la reserva y se emite el ticket con la información de la misma: sede, servicio y el valor del importe a abonar. Ante cualquier error o inconveniente al momento de realizar la reserva (por ejemplo que el usuario no exista o que todos los servicios de ese tipo ya estén reservados) debe emitirse el respectivo mensaje informando la dificultad.

Se pide:

- Diseñar el diagrama de clase que represente lo que se describe y requiere.
- Desarrollar en Nassi-Shneiderman las siguientes funcionalidades (cada punto puede necesitar uno o más métodos en distintas clases para completarse):
 1. Constructor de las clases Sede y Ticket.

2. Registrar un nuevo usuario. No retorna ningún valor y debe verificar que el usuario no esté registrado previamente. En caso de estarlo, emitir una notificación.
3. Calcular y devolver el importe a pagar por el usuario. Este importe incluye el valor del recurso que se está registrando y, en caso de que el usuario no haya abonado la cuota mensual, suma el valor de la misma.
4. Reservar un servicio. Si la reserva se efectúa debe actualizar los datos del usuario y devolver el ticket correspondiente (con sede, servicio y el valor del importe a abonar); si la reserva no se efectiviza, debe mostrarse el mensaje indicando el inconveniente y no devuelve nada.