

Enunciado Tema III - Hotel

El hotel "**Estancia Feliz**" necesita desarrollar un sistema para gestionar su inventario de habitaciones. Se requiere implementar un programa en Java que utilice conceptos de colecciones, polimorfismo, ordenamiento y manejo de excepciones para administrar los diferentes tipos de habitaciones que ofrece el hotel.

Crea una clase abstracta llamada "**Habitacion**" con los siguientes atributos privados:

- numero (int): el número de la habitación.
- precio (double): el precio por noche de la habitación.
- capacidad (int): la capacidad máxima de personas en la habitación.
- reservada (boolean): indica si la habitación está reservada o no.

Esta clase debe tener un constructor que reciba los valores para los atributos numero, precio y capacidad, ya que reservada inicialmente estará en false. Además, implementa los siguientes métodos:

- getNumero(): devuelve el número de la habitación.
- getPrecio(): devuelve el precio por noche de la habitación.
- setPrecio(double precio): actualiza el precio por noche.
- getCapacidad(): devuelve la capacidad máxima de personas en la habitación.
- isReservada(): devuelve true si la habitación está reservada, false en caso contrario.
- setReservada(boolean reservada): actualiza el estado de reserva de la habitación.
- mostrarInformacion(): muestra en pantalla el número, precio, capacidad y estado de reserva de la habitación.

Crea tres clases derivadas de "Habitacion": "**HabitacionSimple**", "**HabitacionDoble**" y "**Suite**". Cada clase derivada debe tener atributos adicionales que sean específicos del tipo de habitación. Por ejemplo, "HabitacionSimple" puede tener un atributo "categoria", "HabitacionDoble" puede tener un atributo "vista", y "Suite" puede tener un atributo "servicioAdicional". Asegúrate de que cada clase derivada tenga un constructor que reciba los valores para los atributos y los inicialice correctamente. Además, cada clase derivada debe sobrescribir el método "mostrarInformacion()" para mostrar la información específica de la habitación.

Crear una clase "**Hotel**" que declare una colección (ArrayList) llamada "disponibles" para almacenar las habitaciones disponibles en el hotel. Se debe incluir el método **void agregarHabitacion(Habitacion habitacion)**, también debe incluir un método

"void **mostrarDisponibles()**" que muestre la información de todas las habitaciones en la colección utilizando un bucle "for each".

Agrega un método llamado **ordenarPorPrecio()** en la clase Hotel que ordene la colección disponibles en base al número de las habitación, de forma ascendente.

Crear una **interfaz Oferta** que incluya el método **void calcularDescuento(int porcentaje)** que setea el descuento en precio según el porcentaje. Hacer que las habitaciones tipo Suite implementen esta interfaz. (Nota:tenga en cuenta que las clases envoltorio implementan compareTo)

Tenga en cuenta la siguiente **clase Main**

```
public class Main {
    public static Habitacion pedirDatos(String tipo){ //propagar excepciones
        Habitacion habitacion=null;
        Scanner leer=new Scanner (System.in);
        System.out.println("Ingrese número: ");
        int numero=leer.nextInt();
        leer.nextLine();
        System.out.println("Ingrese precio: ");
        double precio=leer.nextDouble();
        leer.nextLine();
        System.out.println("Ingrese capacidad: ");
        int capacidad=leer.nextInt();
        leer.nextLine();
        switch (tipo){
            case "Simple": System.out.println("Ingrese categoría: ");
                String categoría=leer.nextLine();
                habitacion=new HabitacionSimple(numero,precio,capacidad,categoria);
                break;
            case "Doble": System.out.println("Ingrese dimension: ");
                String vista=leer.nextLine();
                leer.nextLine();
                habitacion=new HabitacionDoble(numero,precio,capacidad,vista);
                break;
            case "Suite": System.out.println("Ingrese servicioAdicional: ");
                String servicioAdicional=leer.nextLine();
                habitacion=new Suite(numero,precio,capacidad,servicioAdicional);
                break; }
        return habitacion; }
    public static void main(String[] args) {
        Hotel hotel = new Hotel();
        //controlar excepciones como dice el enunciado
        Habitacion simple = pedirDatos("Simple");
        //agregar simple a la coleccion
        Habitacion doble = pedirDatos("Doble");
        //agregar doble a la coleccion
        Habitacion suite = pedirDatos("Suite");
        //agregar suite a la coleccion
        //calcular 15% de descuento a suite
        //ordenar colección
        //mostrar coleccion
    }
}
```

```
}
```

se pide modificar la clase Main para que:

- Modificar el método `static Habitacion pedirDatos(String tipo)` para que **propague excepciones** del tipo `InputMismatchException` y `Exception`.

1- Dentro del método `main()`, se debe agregar un bloque `try-catch-finally` al crear una nueva Habitación. en caso de que se produzca una excepción y proporcionar información adicional sobre el problema:

-Si ocurre una excepción tipo `InputMismatchException` se debe mostrar el mensaje "Error de tipo de datos", si ocurre una excepción `Exception`, se debe mostrar un cartel "Error general". El bloque `finally` debe mostrar un cartel "En bloque finally"

2- agregar las habitaciones creadas a la colección.

3- calcular descuento del objeto suite.

3- ordenar la colección.

4- Mostrar la información de la colección disponibles usando el método de la clase Hotel .