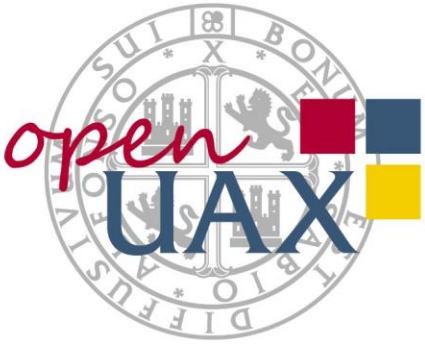
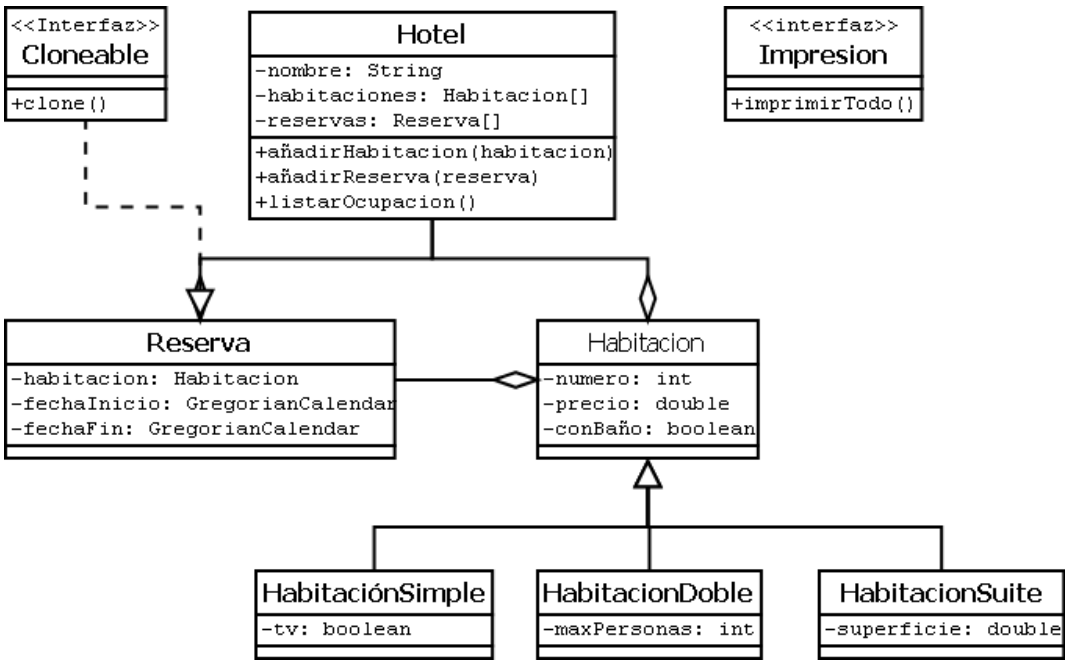


3. Ejercicio Feedback

ENUNCIADO

Se desea realizar un programa para llevar la gestión de las reservas de un hotel. Como parte del análisis de este programa se ha decidido escribir las siguientes clases, cuyo diagrama es el siguiente:



Se recomienda realizar las clases en el orden en están descritas en el enunciado.

Si necesita métodos o atributos adicionales, puede añadirlos con un comentario que describa qué función tendrá en la clase.

Ejercicio 1

Interfaz Impresión:

- Debe ser implementado por todas las clases.
- Las clases abstractas deben delegar la implementación.
- Las clases que implementan el método deben escribir toda la información.

Ejercico 2

Interfaz Cloneable:

- Es la interfaz del Sistema. No se debe crear.

Ejercicio 3

Habitación:

- Es una clase abstracta
- No debe escribir el código de Impresión, delegándolo en los distintos tipos de habitaciones.
- Tendrá un constructor con parámetros para todos sus atributos.
- Tendrá métodos get y set para todos sus atributos.

Ejercicio 4

HabitaciónSimle:

- Hereda de Habitación.
- Tendrá un constructor con todos sus atributos.
- Tendrá métodos get y set para todos sus atributos.

Ejercicio 5

HabitaciónDoble:

- Hereda de Habitación.
- Tendrá un constructor con todos sus atributos.
- Tendrá métodos get y set para todos sus atributos.

Ejercicio 6

HabitacionSuite:

- Hereda de Habitación.
- Tendrá un constructor con todos sus atributos.
- Tendrá métodos get y set para todos sus atributos.

Ejercicio 7

Reserva:

- Tendrá un constructor con todos sus atributos.

- La habitación es una de las habitaciones del hotel.
- La fecha de inicio siempre debe ser posterior a la de finalización. Si esto no se cumple en alguna operación, incluido el constructor, debe lanzar la excepción IllegalArgumentException.
- Tendrá métodos get y set para todos sus atributos.
- Debe implementar la interfaz Cloneable para hacer una copia **profunda** del objeto.

Ejercicio 8

Hotel:

- El constructor recibe como parámetro un nombre y un número de habitaciones. Crea las estructuras para esas habitaciones y un máximo de 200 reservas.
- El método añadirHabitación recibe una habitación y la añade al array de habitaciones.
- El método añadirReserva recibe una habitación y las fechas de inicio y fin. Comprueba que la habitación existe en el hotel, y que no está ya reservada para las fechas indicadas. En este caso crea una reserva y la añade al array de reservas. Si no puede crear la reserva debe generar la excepción HabitacionOcupadaException.
- El método ListarOcupación escribe por pantalla todas las reservar cuya fecha de inicio es anterior a la fecha actual y fecha de fin es posterior a la fecha actual. Use compareTo de GregorianCalendar.

Ejercicio 9

Programa principal:

- Cree un hotel
- Cree 3 habitaciones simples, 3 habitaciones dobles y 2 suites, con los datos que considere.
- Añada todas las habitaciones creadas al hotel
- Reserve una habitación simple desde el 21/12/2012 hasta el 24/12/2012.
- Reserve una habitación doble del 26/11/2012 al 30/11/2012.
- Reserve una suite del 01/01/2013 al 3/1/2013.
- Reserve la misma suite del 2/1/2013 al 4/1/2013. Esta reserva debe generar una excepción. Capturar y escribir un mensaje
- Escribir el listado de ocupación del hotel.
- Llamar a imprimirCompleto del hotel.

Ejercicio 10

En un proyecto distinto:

Un pájaro puede tener las siguientes características:

- Puede volar
- Puede nadar
- Puede bucear

Los murciélagos pueden volar. Los patos vuelan y nadan. Los avestruces no hacen nada de lo anterior.

Usando interfaces, diseñar e implementar el modelo adecuado, teniendo en cuenta que no tiene sentido tener un “pájaro” en general sin especificar la especie. La implementación de los métodos se dejará vacía.

CRITERIOS DE CORRECCIÓN

Los criterios de corrección serán los siguientes:

Ejercicio 1	5%
Ejercicio 2	0%
Ejercicio 3	10%
Ejercicio 4	5%
Ejercicio 5	5%
Ejercicio 6	5%
Ejercicio 7	15%
Ejercicio 8	20%
Ejercicio 9	15%
Ejercicio 10	20%