

Enunciado

Una empresa que construye ascensores nos solicita desarrollar un programa que controle el funcionamiento de todos los ascensores de un edificio.

El **Edificio** tiene como atributos la cantidad de pisos y una colección con sus ascensores.

De cada **Ascensor** conocemos su número identificador, el piso en donde se encuentra parado (inicia en 0, que representa Planta Baja) y un modo de funcionamiento que indica a qué pisos puede acudir (TODOS, SOLO_PARES, SOLO_IMPARES). Es decir que si un ascensor tiene su modo de funcionamiento en SOLO_IMPARES sólo podrá frenar en los pisos impares.

Se nos pide crear lo siguiente en donde corresponda:

- **Constructor de Edificio:** Recibe el número del último piso del edificio al que llegan los ascensores y la cantidad total de ascensores. Debe dejar los ascensores “listos para funcionar”.
- Método **obtenerMasCercano()**: recibe el número de piso y resuelve y devuelve el ascensor que más cerca se encuentra del piso solicitado. Si un ascensor que “saltea” pisos puede parar en el piso deseado y está tan cerca como uno que acude a todos, el que debe acudir es el primero (el que saltea).
- Método **llamarAscensor()**: recibe por parámetro el número de piso al que hay que acudir y llama al ascensor que se encuentra más cerca. Actualiza el número de piso del ascensor que acudió al llamado y devuelve el número de este ascensor.
- Método **setModoSegunNumeroDeAscensor()**: setea el modo de funcionamiento del ascensor en base a su número identificador. Los primeros dos ascensores del edificio siempre llegan a todos los pisos. A partir del tercero, los que tienen identificador impar se setean en modo SOLO_IMPARES, y los de número par llegan a solo a los pares.
- Método **puedoParar()**: recibe un número de piso y devuelve un booleano indicando si dicho ascensor puede detenerse en el piso indicado.

Recordá: Hay ascensores que pueden llegar a todos los pisos.

Recordá que podés crear todos los métodos que consideres necesarios para mantener el correcto manejo de responsabilidades entre los objetos.

En la clase Test, creá una instancia de Edificio, con 30 pisos y 4 ascensores, y luego ejecutá las instrucciones respetando el siguiente escenario:

1. Llamar ascensor desde piso 14 → Debe mostrar “ASCENSOR 4”
2. Llamar ascensor desde piso 13 → Debe mostrar “ASCENSOR 3”
3. Llamar ascensor desde piso 2 → Debe mostrar “ASCENSOR 2”
4. Llamar ascensor desde piso 16 → Debe mostrar “ASCENSOR 4”
5. Llamar ascensor desde piso 5 → Debe mostrar “ASCENSOR 2”
6. Llamar ascensor desde piso 30 → Debe mostrar “ASCENSOR 4”
7. Llamar ascensor desde piso 1 → Debe mostrar “ASCENSOR 1”

IMPORTANTE: Incluir además, el Diagrama UML en la resolución, en el cual se debe visualizar las relaciones entre los elementos del modelo.

Cuando termines y/o antes de que expire el tiempo del examen exportá el proyecto tal como se pide al principio. Luego subí el archivo al Aula Virtual como entrega del examen.

Asegurate de haber adjuntado lo correcto y finalizá el examen pulsando el botón "Enviar y Terminar"