

Se desea realizar un programa para gestionar los parkings que dispone una empresa. Se debe crear la clase Parking que tiene como atributos el nombre, la categoría, el número total de plazas y el número de plazas ocupadas. También tiene un atributo código que será un valor numérico que debe generarse de forma automática para cada uno de los parkings.

Por último, el parking también tendrá un atributo llamado recaudación que debe guardar todo el dinero que se va cobrando en el parking.

El atributo categoría es una clase que tiene como atributos el nombre de la categoría y el precio por minuto aplicable a los parkings de esa categoría.

Se pide:

1. Crear el constructor de la clase Parking lanzando la excepciones que corresponda, es decir, si el número total de plazas es negativo. El constructor debe recibir el nombre, la categoría y el número total de plazas. Se supone que cuando se crea el parking no hay ningún coche y por supuesto no tiene ningún dinero.
2. Crear los métodos getter y setter necesarios teniendo en cuenta que el número total de plazas y el dinero recaudado no se puede modificar directamente.
3. Crear los siguientes métodos en la clase Parking:
 - a. **cobrarYsalirDelParking**: Se le pasará los minutos que estuvo dentro del parking y devolverá el precio que se le cargará. También debe actualizar el estado del parking, es decir, deberá registrar que ha salido un coche.
 - b. Un método **entrarParking**: que no recibe nada, ni devuelve nada, sólo cuenta que ha entrado un coche en el parking. Si el parking está lleno se debe generar una exception.
 - c. Un método **clonarParking**. Se le pasará una cadena con el nombre del nuevo parking y devolverá un nuevo objeto parking con las mismas características que él y todas las plazas libres.

Se debe crear un programa principal que debe crear un parking y mostrar el siguiente menú:

- 1.- Entrar al parking.
- 2.- Salir de parking.
- 3.- Mostrar información del parking.
- 4.- Salir del programa.

Se deben gestionar las excepciones en el programa general.

2.- Realizar un método para comprobar si un password cumple con las condiciones exigidas, que son:

- La contraseña debe tener una longitud mínima de 8.
- La contraseña debe tener al menos 2 dígitos.
- La contraseña debe tener al menos 2 letras.
- La contraseña debe tener al menos uno de los siguientes caracteres “*-[”

El método debe devolver verdadero si la cadena que recibe como argumento es un password válido.