

Indice

Indice	1
Sistema de Estacionamiento	2
Clases	2
Car	2
Parking	3
Plaza	4
ThreadUtils	4
Ejecución del Programa	4

Sistema de Estacionamiento

Este programa simula un sistema de estacionamiento con múltiples coches intentando entrar y salir de un estacionamiento limitado.

Clases

Car

La clase Car representa un coche en el sistema de estacionamiento. Cada coche es un hilo (Thread) que intenta entrar al estacionamiento, permanece estacionado por un tiempo aleatorio y luego sale. Los métodos principales son:

- Constructor: Car(Parking parking, int offsetId)
 - Inicializa un coche con una referencia al estacionamiento al que pertenece.
- entrar(): boolean
 - Intenta entrar al estacionamiento.
 - Devuelve true si el coche entra con éxito, false si el estacionamiento está lleno.
- salir(): void
 - Sale del estacionamiento.
- run(): void
 - Método principal que se ejecuta cuando inicia el hilo del coche. El coche intenta entrar, espera un tiempo aleatorio, sale y espera otro tiempo antes de intentar entrar nuevamente.
- getPlaza(): Plaza
 - Obtiene la plaza actual del coche.

Parking

La clase Parking representa el estacionamiento. Contiene una lista de plazas y gestiona la entrada y salida de coches. Algunos métodos clave son:

- Constructor: `Parking(int maxCap)`
 - Inicializa el estacionamiento con una capacidad máxima.
- `init(int cantCoches): void`
 - "Abre" el estacionamiento creando coches y asignándoles un offset de ID.
- `addCar(Car c): boolean`
 - Agrega un coche al estacionamiento de manera sincronizada.
 - Devuelve true si el coche se agregó con éxito, false si el estacionamiento está lleno.
- `removeCar(Car c): void`
 - Remueve un coche del estacionamiento de manera sincronizada.
- `getCarByPlazaNum(int n): Car`
 - Obtiene un coche por el número de plaza.
- `getPlazaByCar(Car c): Plaza`
 - Obtiene la plaza ocupada por un coche.
- `getNextPlazaLibre(): Plaza`
 - Obtiene la próxima plaza libre en el estacionamiento.
- `getPlazas(): List<Plaza>`
 - Obtiene la lista de plazas en el estacionamiento.
- `hayHueco(): boolean`
 - Verifica si hay al menos una plaza libre en el estacionamiento.
- `getPlazasLibres(): List<Plaza>`
 - Obtiene la lista de plazas libres en el estacionamiento.
- `existsCar(Car c): boolean`
 - Verifica si un coche existe en el estacionamiento.

Plaza

La clase Plaza representa una plaza en el estacionamiento. Contiene información sobre el número de la plaza, el coche estacionado en ella y métodos para verificar la ocupación de la plaza.

ThreadUtils

Una clase de utilidad ThreadUtils que proporciona un método estático para esperar un número de segundos.

Ejecución del Programa

El programa principal se encuentra en la clase Main. Puedes ajustar el número de plazas y coches modificando las variables `numeroPlazas` y `numeroDeCoches` en el método `main`. Luego, ejecuta el programa para simular el sistema de estacionamiento.