Grupo 8

Integrantes

Brian Ramirez
Luis Salas

Corrige: Fabrizio Britez

UML:
Modelado incompleto
Relación entre AlertListener CellPhone es al reves segun lo que esta en el código.
Cardinalidades: ok
Nomenclatura: ok

Desplazamiento:
No está implementada la lógica de la detección de desplazamiento. Los mensajes walking y driving solo inician o detiene un estacionamiento.
Falta detección de desplazamiento

Modos de inicio y fin de estacionamiento
Bien detectado el patrón y correcta implementación

Monitoreo de Estacionamientos:
No hay modelado en el UML ni implementación en código.

Falta.
Fin de estacionamiento por fin del día.

AppSEM:
No utilizar singleton como patrón para tener acceso “rápido” a los objetos desde cualquier punto del código. El singleton es un patrón para tener control de las instancias de una clase.
AppSEM>>startParking no utilizar System.out.println es preferible devolver un string o utilizar un objeto que “almacene” esos strings.
Las horas hardcodeadas en la validación de tiempo hace que el modelo no pueda escalar y viola Open Close Principle.

ParkingSystem:
Singleton debería de usarse como un patrón de control de instancias.
Valores estáticos hardcodeados hacen que el modelo no escale. Ejemplo: Precio por hora.

ShoppingRecord
Singleton debería de usarse como un patrón de control de instancias.

ParkingArea
ParkingArea>>existParking se puede implementar utilizando stream como hicieron en otros mensajes de la misma clase.

Clases EndParkingResponse y StartParkingResponse no tiene comportamiento. Cual seria la responsabilidad de estos objetos?

Cellphone
Cellphone>>buyCredit Rompen encapsulamiento al acceder a los puntos de venta del área. Deberían delegarle la responsabilidad al área de definir si el punto de venta le pertenece o no.

Clase Car no es necesaria, tampoco tiene responsabilidades ni comportamiento.
Clase TimeUtil no tiene comportaminto. Solo wrapea el coportamiento de la clase LocalTime

La implementación de las sublcases de Parking>>inForce es exactamente igual.

Tests:
No se puede chequear la cobertura.
Faltan clases de tests unitarios.
No utiliza test doubles.
Tests que no compilan.

Faltaría agregar documentación de qué patrones detectaron, porque los identificaron como tal. Qué rol cumple cada clase dentro del modelo según el patrón. Porque se inclinaron por ese patrón y no por otro.

Condicion: **REENTREGA**