# Final PBII 2022

Apellido:

Nombre:

DNI:

1. ¿Cuál es la diferencia entre clases abstractas e interfaces? ¿Para que utilizaría cada una? Ejemplificar.

Las interfaces son un conjunto de métodos que pueden cambiar

1. La ejecución del método actualizarBaseDeDatos(), puede provocar la aparición de las siguientes excepciones:
   1. EOFException
   2. FileNotFoundException
   3. SQLException

Sabiendo que el método devuelve un tipo de dato void, y no recibe parámetros, escriba la firma o declaración del mismo, de manera que, ante la aparición de alguna de las excepciones mencionadas, las lance (throw) para que éstas sean tratadas desde el la clase llamadora.

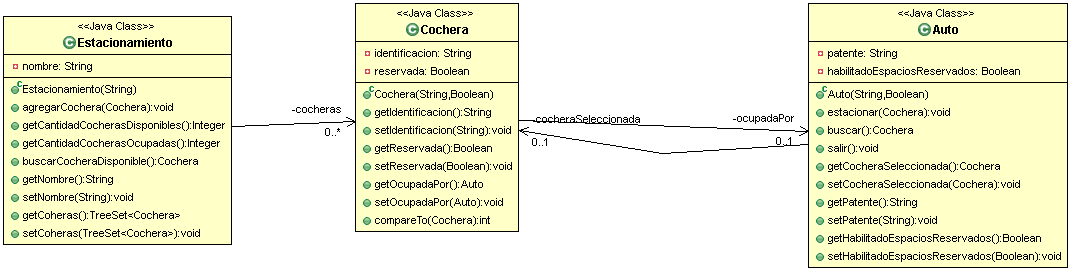
1. Indicar en qué se diferencian las colecciones List (listas) y Set (conjuntos). Ejemplifique con una clase que implemente cada una indicando en qué caso las utilizaría.

Una admite repetidos y otra no

1. Establecer la diferencia entre una excepción y un error.

Una excepción se puede manejar, un error no. Un error puede ser estar corto de memoria

1. Considerando el siguiente diagrama, se pide:



* 1. Codificar las clases Estacionamiento, Cochera y Auto:
     + El constructor de cada clase debe respetar la firma descripta
     + Los métodos get y set se deben generar automáticamente a través del Eclipse
     + Las cocheras del estacionamiento se deben guardar en un TreeSet<Cochera>
  2. Desarrollar los siguientes métodos para la clase Estacionamiento:
     1. agregarCochera: Agrega una cochera al conjunto de cocheras.
     2. getCantidadCocherasDisponibles: Recorre el conjunto de cocheras, suma aquellas que NO TIENEN ningún auto estacionado y retorna el resultado.
     3. getCantidadCocherasOcupadas: Recorre el conjunto de cocheras, suma aquellas que TIENEN un auto estacionado y retorna el resultado.
     4. buscarCocheraDisponible: Recorre el conjunto de cocheras y devuelve la primera que esté disponible (no tiene un auto estacionado).
  3. Desarrollar los siguientes métodos para la clase Auto:
     1. estacionar: Intenta ocupar una chochera. Se debe validar que el auto está habilitado a estacionar en dicha cochera. Un auto está habilitado para estacionar en una cochera cuando se cumplen las siguientes condiciones:
* La cochera está disponible y no es reservada
* La cochera está disponible, es reservada y el auto está habilitado para ocupar cocheras reservadas.

Si el auto intenta estacionar en una cochera no habilitada, se debe generar una excepción con el mensaje correspondiente:

throw new Exception(mensaje);

* + 1. buscar: Devuelve la cochera donde se encuentra el auto.
    2. salir: Debe liberar la cochera ocupada por este (this) auto.

1. Desarrollar la clase TestEstacionamiento, incluyendo los test que considere necesarios para garantizar el correcto funcionamiento de las clases propuestas. Al menos se deben incluir los siguientes:
   * 1. testQueAlComenzarLaCantidadDeCocherasOcupadasEsCero
     2. testQueAlIngresarUnCocheLaCantidadDeCocherasDisponiblesEs29
     3. testQueAlIntentarEstacionarUnAutoNoHabilitadoEnUnaCocheraReservadaLaCantidadDeCocherasDisponiblesNoCambia
     4. testQueCuandoElEstacionamientoEstaCompletoNoPermiteEstacionarUnNuevoAuto