

PRIMER EXAMEN PARCIAL

29/04/2021

😊 Leé por lo menos dos veces el enunciado antes de resolver.

Enunciado

La sede de licencias de conducir de una jurisdicción nos pide un prototipo para generar algunas estadísticas de los exámenes tomados.

Todo examen consta de una fecha (**String**) y la persona que lo hizo (DNI, nombre y apellido). Como parte del examen, se debe realizar un circuito en caso de concursar para licencia de auto y varios circuitos en caso de concursar para licencia de moto.

Los exámenes de moto guardan, además de los circuitos hechos, información sobre la moto con la que se hicieron las pruebas (patente, marca, cilindrada y si llevaba todos los espejos).

Los exámenes de auto guardan, además del único circuito realizado, información del auto con el que se hizo (patente, marca, cilindrada y cantidad de puertas).

De los circuitos (para cualquier tipo de examen) se sabe el tiempo en segundos que tomó y el número de faltas cometidas. Los circuitos y los exámenes son evaluables, ya que deben dar un resultado: **aprobó o no**.

Condiciones para aprobar:

- ★ Circuitos: Tener 2 faltas o menos.
- ★ Exámenes de auto: Que su circuito se haya aprobado en un minuto y medio o menos.
- ★ Exámenes de moto:
 - Si la cilindrada de la moto es superior a 150cc, todos los circuitos deben haber sido aprobados (sin importar el tiempo).
 - Si la cilindrada de la moto es igual o inferior a 150cc, se puede tener hasta un solo circuito desaprobado y el tiempo total entre todos no debe superar los 2 minutos.

Basado en el enunciado descripto, realizá:

- A) El diagrama de clases que lo modelice, con sus relaciones, atributos y métodos.
- B) El método **obtenerInforme()** que debe **devolver** (no mostrar por consola) los siguientes datos:
- Porcentaje de exámenes reprobados.
 - Promedio de tiempo de los exámenes de auto aprobados.

Criterios

Para considerar aprobado el trabajo práctico, el mismo debe demostrar la correcta aplicación de los siguientes conceptos de la programación orientada a objetos:

- Correcta definición de clases y asignación adecuada de sus responsabilidades.
- Encapsulamiento, ocultamiento de información y uso de getters y setters sólo cuando corresponda.
- Modularización reutilizable y mantenible con uso de métodos con correcta parametrización.
- Correcta aplicación de miembros de instancia y de clase.
- Correcta aplicación de herencia y polimorfismo.
- Correcta aplicación conceptual de las relaciones entre clases.

% Correcto	0 a 20	25 a 45	50 a 55	60	65 a 70	75	80	85 a 90	95	100
Nota	1	2	¿4?	4	5	6	7	8	9	10