

Examen Final I Término Período 2013-2014

Fundamentos de Programación Orientada a Objetos

Codificar las siguientes clases:

Clase Pista			
	Atributos	Nombre: String	Validar que no sea null
		longitudPista: int	Debe validar que esté entre 10 y 20 kilómetros
	Operaciones	aumentarKmPista (int)	Aumenta la longitud de la pista de carreras, validar que no se exceda del límite (20Km).

Clase Piloto			
	Atributos	Nombre: String	Validar que no sea null
		carrerasRealizadas: int	Inicia con cero cuando se crea el objeto
		carrerasGanadas: int	Inicia con cero cuando se crea el objeto
	Operaciones	incrementarCarrerasRealizadas()	Incrementa en 1 las carreras realizadas
		incrementarCarrerasGanadas()	Incrementa en 1 las carreras ganadas

Clase Auto			
	Atributos	numero : int	Validar que no sea cero
		kilometraje: int	Debe ser cero al instanciarlo, aumenta por cada vuelta recorrida en las carreras
		nivelGasolina: double	Debe validar que esté entre 1 y 30
		conductor: Piloto	Asignar un objeto tipo Piloto
	Operaciones	aumentarKilometraje (int)	Recibe como parámetro el número de kilómetros a aumentar.

Clase Carrera			
	Atributos	numero	Validar entre 1 y 100
		pistaUsada: Pista	Objeto de tipo Pista, validar que no sea null
		numeroVueltas	Validar entre 10 y 30
		Auto [] Competidores	Vector de autos
		numeroAutoGanador	Es el número del auto que ganó
	Operaciones	asignarCompetidor (auto)	Recibe como parámetro el auto que va a competir. Debe validar que ya tenga asignado un piloto, caso contrario debe mostrar un mensaje de error. Debe aumentar el número de carreras realizadas por el piloto
		eliminarCompetidor (auto)	
		registrarGanador (int)	Recibe como parámetro el número del auto ganador. Debe incrementar el número de carreras ganadas del piloto asignado
		finalizar ()	Finaliza la carrera. Debe aumentar el kilometraje recorrido de los autos. Debe multiplicar el número de vueltas de la carrera por la longitud de la pista, el resultado lo incrementa al kilometraje de los autos que compitieron. También debe bajar el nivel de gasolina, baja un nivel por cada 50km de recorrido

- Debe realizar los métodos GET y SET respectivos y sobrecargar la operación toString para todas las clases.

Instrucciones que deben ejecutarse en el main()

Instrucción	Comentario
Pista objPista1 = new Pista ("SanCiro", 14);	
objPista1.aumentarKmPista (3);	
objPista1.aumentarKmPista (6);	// Envía error, se excede del límite (1 pto)
Piloto objPiloto1 = new Piloto ("Roberto Alonso");	// Inicializa con cero las carreras realizadas, ganadas y no terminadas
Auto objAuto1 = new Auto (1, 1000, 5, objPiloto1);	// Asigna 1 al número de auto, 1000 al kilometraje, 5 al nivel (galones) de gasolina, y al objeto del piloto Roberto Alonso. (1 pto)
Carrera objCarrera1 = new Carrera (3, objPista1, 10);	// Asigna 3 al número de competencia, asigna el objeto tipo Pista cuyo nombre es SanCiro, el 10 es el número de vueltas que deberán realizar los autos en la carrera (1 pto)
objCarrera1. asignarCompetidor (objAuto1); objCarrera1. asignarCompetidor (objAuto2); objCarrera1. asignarCompetidor (objAuto3); objCarrera1. asignarCompetidor (objAuto4); objCarrera1. asignarCompetidor (objAuto5);	// Agrega el auto al vector (1 pto)
objCarrera1. retirarCompetidor (1);	// Eliminar el auto del vector (2 pts)
objCarrera1. registrarGanador (2);	// Debe validar que el número de auto esté registrado en la carrera, sino debería mostrar un mensaje de error. Se debe aumentar el número de carreras ganadas al ganador. También debe actualizar el kilometraje de todos los autos que están en la carrera y el número de carreras realizadas a todos los pilotos. (5 pts)
objCarrera1.imprimirListado():	Imprime lista de competidores de la carrera (2 pts)