UNIVERSIDAD POPULAR DEL CESAR

GUÍA DE ACTIVIDADES PARA LA CONSTRUCCIÓN DE CONOCIMIENTO

NOMBRE DE LA ASIGNATURA	Programación de Computadores II							
NOMBRE DE LA ACTIVIDAD	Parcial 1							
TIPO DE ACTIVIDAD	Sincrónica		Asincrónica	x	Individual	x	Grupal	

TEMÁTICA REQUERIDA PARA LA ACTIVIDAD	**OBJETIVOS**
Unidad 1. Introducción a la POO Unidad 2. Programación basada en objetos	Desarrollo etapa de análisis, diseño e implementación de programa básico

COMPETENCIAS	**INSUMOS PARA EL DESARROLLO DE LA ACTIVIDAD / REFERENCIAS BIBLIOGRÁFICAS**
• Identificación de clases con sus atributos requeridas en el dominio de solución de un problema • Representación de clases mediante UML • Identificación de relaciones de asociación (agregación y composición) entre clases • Representación de relaciones entre clases mediante UML • Diseño de diagramas de clases • Implementación de clases mediante Java	• Material educativo y material complementario de la asignatura "**Unidad 1.**" • Material educativo y material complementario de la asignatura "**Unidad 2.**" • Talleres y codigo elaborado en el desarrollo de la asignatura.

CONOCIMIENTOS PREVIOS REQUERIDOS

Conceptos fundamentales de POO - Estructura básica de clases – Relaciones entre clases y UML

ESPECIFICACIONES DE LA ACTIVIDAD

Problema planteado:

Por motivo de las fiestas patronales del municipio, se va a realizar el concurso de lanzamiento de llantas, y se requiere un programa que permita calcular al ganador.

Las reglas del concurso son las siguientes:

- Solos pueden participar tres concursantes (A, B, C), de los cuales se debe conocer su nombre, apellido, su número y la Llanta con la que participa.
- Cada concursante puede llevar la llanta de su preferencia, pero es necesario conocer de ella su diámetro (en metros).
- Cada concursante lanzara su llanta de manera horizontal en un terreno plano, un delegado del concurso anotara el número de giros (vueltas) que dio la llanta de cada concursante. Este valor es muy importante, porque con él, el sistema deberá calcular el resultado de su lanzamiento, es decir, la distancia recorrida por cada llanta.
- El concursante ganador será aquel que su llanta tenga el mayor trayecto recorrido.

Un equipo de diseño sugiere que para la construcción del programa es requerido contemplar las clases Concurso, Participante y Llanta.

A usted, como parte del equipo, le asignan la tarea de identificar los atributos y métodos que considere necesario para cada una de estas clases y la implementación del programa.

Así mismo, se le indica que para su diseño deberá tener en cuenta el siguiente ejemplo de clase principal, como entrada y salida del programa:

Entrada:

```
Participante A = new Participante("Luis", "Perez", 1, new Llanta(10));
Participante B = new Participante("Carlos", "Lopez", 2, new Llanta(5));
Participante C = new Participante("Gorge", "Pinto", 3, new Llanta(7));

Concurso concurso = new Concurso(A,B);
concurso.setC(C);

System.out.println("Resultado lanzamientos:");
System.out.println("-------------------------------------");
System.out.println("Resultado A: " + concurso.getA().resultadoLanzamiento(3));
System.out.println("Resultado B: " + concurso.getB().resultadoLanzamiento(6.001));
System.out.println("Resultado C: " + concurso.getC().resultadoLanzamiento(4));
System.out.println("");

Participante ganador = concurso.getGanador();
System.out.println("Ganador: " + ganador);
```

Salida:

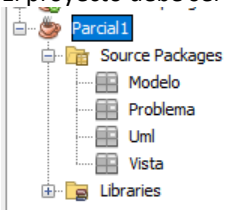
```
Resultado lanzamientos:
-------------------------------------
Resultado A: 94.24777960769379
Resultado B: 94.26348757096174
Resultado C: 87.96459430051421

Ganador: Nombre=Carlos, Apellido=Lopez, Numero=2
BUILD SUCCESSFUL (total time: 0 seconds)
```

Requerimiento de la evaluación:

- Se debe diseñar el respectivo diagrama de clases, el cual debe incluir Clases, atributos, y relaciones entre clases (indicando tipo de relación, nombre, navegabilidad y cardinalidad).

- El proyecto debe ser desarrollado en NetBeans, y debe mantener la siguiente estructura:



El código fuente debe cumplir con las siguientes prestaciones:

- El código fuente debe ser legible, bien identado y sin errores de sintaxis. Debe seguir las convenciones de Java en lo que se refiere a nombres de las clases, nombres de métodos, atributos y constantes.
- Código que no compile, no es calificado, es decir, debe ser funcional el aplicativo desarrollado.

Otras consideraciones:

- El proyecto es individual.
- El código implementado debe ser consistente con el diagrama de clases diseñado (con los atributos definidos, métodos y relaciones entre clases.)

- **Modo de entrega**

Se debe montar en aula web enlace a repositorio GitHub del código elaborado.

RECOMENDACIONES / OBSERVACIONES	Sin observaciones