

**Universidad Tecnológica Nacional
Facultad Regional Avellaneda**




Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos										
Materia: Laboratorio de Programación II										
Apellido:				Fecha:	19/11/2019					
Nombre:				Docente ⁽²⁾ :	Dávila/Boullon					
División:				Nota ⁽²⁾ :						
Legajo:				Firma ⁽²⁾ :						
Instancia ⁽¹⁾ :	PP		RPP		SP	x	RSP		FIN	

(1) Las instancias válidas son: 1^{er} Parcial (**PP**), Recuperatorio 1^{er} Parcial (**RPP**), 2^{do} Parcial (**SP**), Recuperatorio 2^{do} Parcial (**RSP**), Final (**FIN**). Marque con una cruz.
 (2) Campos a ser completados por el docente.

IMPORTANTE:

- Guardar el proyecto en el **disco D:**. Ante un corte de energía o problema con el archivo de corrección, el alumno será responsable de que el proyecto sea recuperable.
- **2 (dos) errores en el mismo tema anulan su puntaje.**
- **Errores de conceptos de POO anulan el punto.**
- **Cada tema vale 1 (un) punto (Herencia, Generics, Test Unitarios, etc.). La correcta documentación también será evaluada.**
- **Se deberán tener al menos el 60% bien de los temas a evaluar según la instancia para lograr la aprobación.**
- Colocar sus datos personales en el nombre del proyecto principal, colocando: Apellido.Nombre.AñoCursada. Ej: Pérez.Juan.2019. No se corregirán proyectos que no sea identificable su autor.
- **Salvo que se indique lo contrario, TODAS** las clases deberán ir en una Biblioteca de Clases llamada Entidades.
- No se corregirán exámenes que no compilen.
- **Reutilizar** tanto código como crean necesario.

Al finalizar, colocar la carpeta de la Solución completa en un archivo ZIP que deberá tener como nombre

Apellido.Nombre.AñoCursada.zip y dejar este último en el Escritorio de la máquina. Luego presionar el botón  de la barra superior, **colocar un mensaje** y presionar *Aceptar*. **Aguardar a que el profesor indique que el examen fue copiado de forma correcta.** Luego retirarse del aula.

TIEMPO MÁXIMO PARA RESOLVER EL EXAMEN 110 MINUTOS.

Utilizar la cáscara proporcionada por los profesores (Esta es similar al primer parcial). Y actualizarla según los próximos puntos.

AstroDelegate:

1. Retorna void y recibe un string.

Astro:

2. La clase se debe poder serializar en XML.
 - a. Tener en cuenta que tanto los hilos como los eventos no pueden serializarse y necesitan el tag "XmlIgnore".
 - b. Hacer todo lo necesario para que esto sea posible (pueden hacerse cambios que **no estén en el diagrama de clases**).
3. Agregar los atributos privados hiloRotacoin e hiloOrbita con propiedades de lectura y escritura.
4. Agregar el evento AstroEvent del tipo AstroDelegate.

AstroExtension:

5. GuardarPlanTxt: Método que extiende *Planeta<Satelite>* permite guardar en un archivo **planeta.txt** de texto con el método **Mostrar**.
6. GuardarXML: Extiende *Satélite* y guarda al mismo en un archivo **satelites.xml**.

IAstro:

7. Debe respetar los métodos que están en el diagrama de clases.

Planeta:

8. Planeta debe ser genérico y debe presentar la siguiente restricción de que el tipo hereda de Astro e implementa IAstro.
9. RunEvtRotar: debe tener un bucle (while, for o cualquiera), en el cual se duerme el hilo durante 3000 milisegundos y debe iterar una cantidad de veces definida por el atributo DuraRotacion. Cuando termina de iterar debe llamar al evento AstroEvent, se le debe pasar el resultado del método Rotar().
10. RunEvtOrb: hace lo mismo que el metodo de arriba pero duerme el hilo 1000 milisegundos y toma como referencia el duraOrbita. Le pasa al evento el resultado del método Orbitar().

Satélite:

11. Debe ser serializable.
12. RunEvtRotar: debe tener un bucle (while, for o cualquiera), en el cual se duerme el hilo durante 800 milisegundos y debe iterar una cantidad de veces definida por el atributo DuraRotacion. Cuando termina de iterar debe llamar al evento AstroEvent pasandole el resultado de Rotar().
13. RunEvtOrb: hace lo mismo que el metodo de arriba pero duerme el hilo 400 milisegundos y toma como referencia el duraOrbita pasandole el resultado de Orbiar().

SateliteDB:

14. Leer: debe leer de la base de datos y traer los satélites que haya.
15. Guardar: debe guardar un satélite.

SateliteException:

16. Esta excepción se debe lanzar cuando se intenta crear un satélite que ya se encuentra en la lista. Utilizar la sobrecarga del == que ya está desarrollada.
17. Debe tener un constructor sin parámetros y siempre debe retornar el mensaje: **"No se puede crear el satélite"**.

Excepciones:

18. Manejar todos los escenarios en donde pueda haber excepciones, especialmente con base de datos y archivos.

Test Unitarios:

19. Testear que se lance la excepción al querer cargar 2 satélites iguales
20. Guardar y leer un satélite. comparar que sus atributos nombre, duraOrbita y duraRotacion sean iguales

FormParcial:

21. Declarar un atributo Planeta del tipo Satélite. Instanciarlo en el constructor con los valores { Nombre: "Júpiter", rotación: 20, órbita: 20 }.
22. Crear Satélite: crea una instancia de satélite y lo guarda en la lista de planetas. Además agrega este a la base de datos.
 - a. No validar los campos pero se debe controlar la excepciones que puedan darse.

23. Botón Rotar Planeta:

- Si AstroEvent no tiene manejadores agrega el método MostrarMensajeRotar como manejador.
- Previamente se debe validar si el atributo HiloRotacion es null, si es null se debe instanciar un hilo y asignarlo, posteriormente se corre el hilo junto con el método RunEvtRotar. Caso contrario no se hace nada ya que el planeta ya tendría su hilo.

24. Botón Orbitar Satélites: por cada satélite se debe:

- Chequear si el evento del satélite tiene manejadores asignados.
- Si no tiene agregarlo.
- Si el HiloOrbita es null o no está corriendo se debe crear un hilo nuevo asignarlo y correrlo junto con el método RunEvtOrb.
- Si está corriendo mostrar un messagebox indicando que no se pudo iniciar la Orbita.

25. MostrarMensaje: de recibir un elemento del tipo string y muestra el mismo en un messagebox.

26. Cuando se inicia el form se debe instanciar el planeta y traer los satélites que haya en la base de datos.

Clases:

