Instituto de Tecnología ORT
Carrera de Analista de Sistemas
Materia: Programación 1

EXAMEN PARCIAL 1

Leé con cuidado el enunciado y por lo menos dos veces para resolver lo pedido. Pensá bien la estrategia de resolución antes de comenzar el desarrollo de lo que te solicitan. El objetivo de este examen es **evaluar la correcta aplicación de los conceptos y técnicas** vistos hasta el momento:

- *Correcta implementación de constructores.*
- *Modularización reutilizable y mantenible con uso de métodos con correcta parametrización y correcto encapsulamiento, publicando setters y getters sólo cuando corresponda.*
- *Manejo de clases y colecciones utilizando Herencia y Polimorfismo.*
- *Generación de diagramas UML y Nassi-Shneiderman.*

Enunciado

La estación espacial Atlantis dispone de puntos de atraque que alquila a naves espaciales de distinto tipo. Cada uno de estos puntos de atraque puede “enganchar” solo una nave, de las cuales por ahora hay dos tipos principales:

Las naves de tipo Cargo, dedicadas al transporte de mercadería, guardan la información de su carga: descripción y volumen en metros cuadrados. Las naves de tipo Cruiser, que transportan pasajeros, guardan la cantidad actual de pasajeros. Ambas naves, por ser comerciales, guardan el nombre de la compañía naviera.

Antes de atracar, todas las naves deben informar un “manifiesto” que en todos los casos consta de datos que hay guardados en cada una de ellas:

- El nombre del planeta de origen de la nave.
- El nombre del último planeta visitado.
- La cantidad de tripulantes.

Cada tipo de nave agrega datos particulares al manifiesto:

- Cargo: el detalle de la carga con formato: “`descripcion (volumen m³)`”.
- Cruiser: El porcentaje de seres a bordo que son pasajeros, respecto al total (pasajeros + tripulantes).

Se pide:

- Diseñar el diagrama UML completo que resuelva el problema propuesto agregando lo que consideres necesario a lo previamente enunciado.
- Desarrollar en NS+ los siguientes puntos (incluyendo los métodos derivados que utilices):
    - **`imprimirManifiestos`**(), que muestre por consola los manifiestos de todas las naves que haya estacionadas.
    - **`estacionarNave`**(...), que reciba una nave y la intente estacionar en el primer punto de atraque libre, siempre que la nave esté en condiciones de estacionarse:
        - Las naves Cargo pueden estacionarse si y sólo si el volumen de carga no supera 100 m^3.
        - Las naves Cruiser pueden estacionarse si y sólo si hay pasajeros.

      Devuelve si se pudo estacionar o no.

Criterios

Para considerar aprobado el ejercicio del examen éste 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 uso de las herramientas de modelado UML y desarrollo en Nassi-Shneiderman.

IMPORTANTE:

Cuando termines y/o antes de que expire el tiempo del examen juntá los archivos Nassi-Shneiderman de NS+ (formato .nsplus) con ambos archivos de UMLetino y adjuntalos al examen en un archivo .zip cuyo nombre será SEDE_CURSO_APELLIDO_NOMBRE.zip (reemplazando cada parte por lo que corresponda, por ejemplo BE_PR1A_PEREZ_JUAN.zip).

La no entrega de alguno de los archivos invalida el examen. Asegurate de haber adjuntado lo correcto y finalizá el examen pulsando primero "Terminar intento" y luego el botón "Enviar todo y Terminar".