

PRIMER EXAMEN FINAL

03/12/2020

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

Enunciado

El juego "AmongUs" se hizo muy popular en esta segunda mitad de este año de pandemia.

Una partida del juego posee varios jugadores denominados tripulantes y varios jugadores denominados impostores.

De cada tripulante se conoce su nombre, su color (String), un flag que indica si está vivo o muerto y una lista de tareas que debe ir realizando.

De cada impostor se conoce su nombre, su color (String), un flag que indica si está vivo o muerto y una lista de aquellos tripulantes que haya asesinado.

Cada tarea tiene un nombre y su estado (PENDIENTE, EN_CURSO o REALIZADA).

La partida termina cuando se cumplen las condiciones de victoria del bando de los tripulantes o del bando de los impostores.

Para que ganen los tripulantes, debe pasar alguna de las siguientes condiciones:

- ✓ Todos los impostores de la partida deben estar muertos.
- ✓ Todos los tripulantes deben tener todas sus tareas realizadas.

Para que ganen los impostores, debe pasar lo siguiente:

- ✓ La cantidad de impostores vivos debe ser mayor o igual a la cantidad de tripulantes vivos.

Se pide desarrollar:

- Completar el Diagrama de Clases UML **ya provisto**.

Importante: Las clases, atributos, métodos y relaciones ya provistas no pueden cambiarse, pero, si lo considera necesario, puede agregar elementos al diagrama.

- La explotación de los siguientes métodos en NS:
 - El método **obtenerBandoGanador(...)**:
Debe devolver uno de estos posibles resultados:
 - TRIPULANTES: se cumplen las condiciones de victoria del bando de los tripulantes.
 - IMPOSTORES: se cumplen las condiciones de victoria del bando de los impostores.
 - INDEFINIDO: este caso es cuando aún no se cumplen las condiciones de victoria de ninguno de los dos bandos.

Criterios

Para considerar aprobado el examen, el mismo debe resolver lo pedido y aplicar los siguientes conceptos de la programación orientada a objetos:

- Detección de clases, atributos, métodos y relaciones (asociativas y de uso).
- Modularización reutilizable y mantenible usando métodos con correcta parametrización.
- Asignación de responsabilidades a cada clase y correcto encapsulamiento.
- Manejo del concepto de instancia y de la interacción entre objetos.
- Manipulación de listas de objetos (ArrayList) y su uso en ciclos condicionales y for-each.
- Manejo de diagramas Nassi-Schneiderman y UML de 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