

EXAMEN FINAL

08/07/2021

🧐 Leé por lo menos dos veces el enunciado antes de resolverlo.

Enunciado

Hace mucho tiempo, en una galaxia muy lejana... había muchos droides que formaban parte de la existencia cotidiana de todos.

Algo particular unía a todos esos droides: la capacidad que tenían de autorrepararse. Ante algún desperfecto, podían reemplazar la o las piezas que no estaban funcionando por otras (con el mismo nombre) para seguir operando.

Para esto, cada droide llevaba un registro detallado de cada una de las piezas que lo componían y de cada una de ellas, manejaban un estado indicando si la pieza estaba operativa o no. Cada una de estas piezas tiene un nombre (**String**) que es el mismo en todos los droides (por ejemplo, "Batería de litio", "Sensor de proximidad", "Visor nocturno", etc).

Siempre que un droide encontraba a otro droide fuera de servicio (luego de batallas o simplemente por el uso) lo registraba para así, de ser necesario, usar las piezas sanas de éste para repararse. Al necesitar una pieza buscaba entre los droides rotos que tenía registrados, chequeando si encontraba entre ellos las piezas sanas que necesitaba. De encontrarlas, reemplazaba sus piezas no operativas por las operativas encontradas en los otros droides.

Se pide:

- Confeccionar el diagrama UML que describe el escenario del enunciado incluyendo los atributos de cada clase y los métodos a desarrollar y aquellos que creas conveniente.
- Desarrollar el método **autoRepararse()** de la clase Droide, que no recibe parámetros. Debe intentar reemplazar sus piezas no operativas por las piezas operativas que pudiera encontrar en alguno de los otros droides. Este método devuelve la lista de piezas no operativas que pudieron ser reemplazadas.

Importante: la clase Droide cuenta con un método llamado **"reemplazarPieza"**. **No hay necesidad de desarrollarlo, solo tenés que utilizarlo** donde consideres necesario.

Pieza **reemplazarPieza(int, Pieza)** recibe como parámetro la posición de la pieza a reemplazar y la pieza de reemplazo. El método devuelve la pieza reemplazada, o *null* si la posición indicada por el parámetro está fuera de rango (no es válida para la cantidad de elementos de la lista).

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 reusable y mantenible usando métodos con correcta parametrización.

EXAMEN FINAL

08/07/2021

- 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, eligiendo siempre el más adecuado en cada situación.
- 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