Ejercicios integradores

1. Nos contratan de las Fuerzas Armadas para desarrollar una aplicación que les permita planificar las batallas que se pueden librar en las distintas zonas de conflicto.
   Para eso nos comentan que no enviarán hombres al frente sino Vehículos con distintas características.
   Algunos vehículos pueden desempeñarse por tierra, otros por agua y otros por aire. Sin embargo, las últimas tecnologías les permitieron a las fuerzas incorporar nuevos vehículos:
   - Anfibios: Pueden desempeñarse tanto por tierra como por agua
   - Hidroaviones: Son un tipo de avión especial que puede (además de volar) realizar operaciones por agua.

Como información adicional para la construcción del software nos presentan los siguientes requerimientos:

1. Los vehículos de guerra se identifican unívocamente por su código.
2. Las batallas se identifican unívocamente por su nombre.
3. El convoy es el conjunto de Vehículos que disponen (no importa el tipo de vehículo).
4. Para enviar un Vehículo a una batalla se debe confirmar que el mismo forme parte de la fuerza.
5. El Vehículo que se envía a una batalla determinada debe ser apto para esta:
   a. En las batallas aeras, sólo se podrán utilizar Vehículos Voladores.
   b. En las batallas por tierra se podrán utilizar Vehículos Terrestre.
   c. En las batallas por agua sólo se podrán utilizar Vehículos Acuáticos.

Para resolver el ejercicio planteado se recomienda tomar como base el siguiente proyecto (https://github.com/jmonteagudo/recuperatorio), logrando que todos los test se ejecuten de manera exitosa, e incorporando los test adicionales que crea conveniente.

2. Nos solicitan desarrollar el software para un club que se dedica a la organización de eventos deportivos del siguiente tipo:
   a. Natación
   b. Ciclismo
   c. Pedestrismo (Running)
   d. Acuatlon (Combinación entre natación y pedestrismo)
   e. Duatlon (Combinación entre pedestrismo y ciclismo)
   f. Triattlon (Combinación de las tres diciplinas, natación, ciclismo y pedestrismo)

   Para poder participar de los eventos deportivos, primero se tiene que ser socio. Los socios se identifican unívocamente por el número de socio, y, según la disciplina que realizan, se requiere conocer determinada información:
   - Nadadores: Se desea conocer su estilo preferido (Los estilos son Croll, Espalda, Pecho y Mariposa).

- Ciclistas: Se desea conocer el tipo de bicicleta que utilizan (De montaña, de ruta o de triatlón)
- Corredores: Se desea conocer su distancia preferida (5 Km, 10 Km, 21 Km o la Maratón de 42 Km).

Para este club la seguridad de sus socios es muy importante, en consecuencia, nos encargan custodiar que el sistema no permita que un deportista no pueda inscribirse en una competencia para la cual no esté preparado, por tal motivo, para el caso que se intente realizar eso, el sistema tiene que generar una Excepción (NoEstaPreparado). Un deportista "no está preparado" si por ejemplo desea participar de un torneo de natación, pero no sabe nadar.

Para resolver el ejercicio planteado se recomienda tomar como base el siguiente proyecto (https://github.com/jmonteagudo/eva03), logrando que todos los test se ejecuten de manera exitosa, e incorporando los tests adicionales que crea conveniente.

3. Se desea desarrollar un software de monitoreo, el cual sirva para detectar colisiones entre los vehículos que circulan por un área determinada. Sabiendo que se cuenta con distintos tipos de vehículos (medios de transporte), se desea conocer constantemente la ubicación (coordenadas) de cada uno, para poder evaluar si se produce una colisión (las coordenadas de dos vehículos es la misma).
   Para esto se solicita desarrollar el método "actualizarCoordenadas", el cual, ante la ocurrencia de una colisión debe generar una excepción "CollitionException".

   Para resolver el ejercicio planteado se recomienda tomar como base el siguiente proyecto (https://github.com/jmonteagudo/PBIIEjemploSegundoParcial), logrando que todos los test se ejecuten de manera exitosa, e incorporando los tests adicionales que crea conveniente.