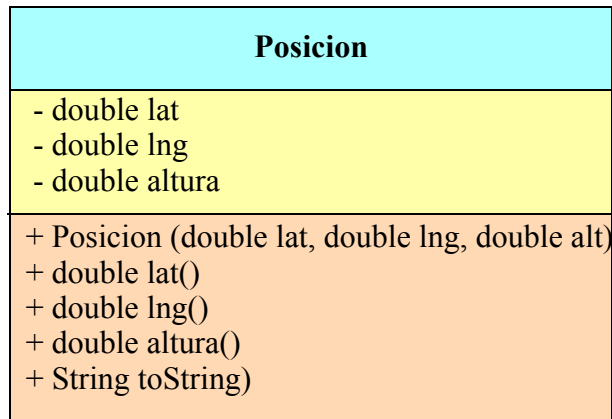


Examen de Prácticas de Introducción al Software (Ingeniería Informática)

Septiembre 2012

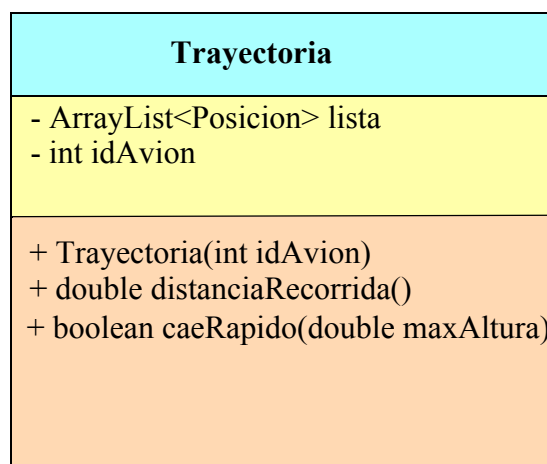
Se desea realizar una parte del software perteneciente a un sistema de control de tráfico aéreo. Para ello, se dispone de la siguientes clase, ya realizada, que funciona de acuerdo al siguiente diagrama de clases:



La clase `Posicion` almacena la latitud y longitud que definen las coordenadas en la superficie de la tierra de un avión, así como su altitud. Sus métodos son:

- *Constructor*, al que se le pasan la latitud y longitud, en grados, y la altura en metros.
- `lat()`: observador de la latitud; la devuelve en grados. La latitud aumenta de abajo a arriba en el mapa.
- `lng()`: observador de la longitud; la devuelve en grados. La longitud aumenta de izquierda a derecha en el mapa.
- `altura()`: observador de la altura; la devuelve en metros.
- `toString()`: retorna las coordenadas convertidas a texto.

Por otro lado, queremos completar la implementación de la clase `Trayectoria` que almacena la trayectoria de un avión, y hace operaciones con ella. La clase tiene el siguiente diagrama de clases:



La clase `Trayectorias` almacena en `lista` la trayectoria que ha ido siguiendo el avión, es decir, las posiciones de los puntos por los que ha pasado, obtenidas de un radar. Asimismo debe almacenar el identificador del avión en `idAvion`. Sus métodos son:

- *Constructor*: Almacena el identificador del avión `idAvion` que se pasa como parámetro. Por otra parte crea la `lista` y la rellena con datos obtenidos de un radar (en la implementación actual, para pruebas, rellena la lista con datos sintéticos).
- `distanciaRecorrida()`: Esta operación calcula y retorna la distancia total recorrida por el avión en millas náuticas. La distancia es la suma de las distancias entre cada punto de la lista y el siguiente. La distancia entre dos puntos $(lat_1, long_1)$ y $(lat_2, long_2)$ en millas náuticas se calcula con la fórmula:

$$60 \cdot \text{acos}(\sin(lat_1)\sin(lat_2) + \cos(lat_1)\cos(lat_2)\cos(long_1 - long_2))$$

donde el arcocoseno está expresado en grados.

- `caeRapido()`: Hace una búsqueda para comprobar si alguna de las posiciones almacenadas en `lista` tiene una diferencia de altura demasiado alta con respecto a la posición inmediatamente anterior. Retorna `true` en caso afirmativo o `false` en caso negativo. La diferencia de altura se considera demasiado alta si es superior al parámetro `maxAltura` en valor absoluto.

El constructor se ofrece ya realizado. Lo que se pide es completar el resto de los métodos de la clase `Trayectoria`.