

ESCUELA COLOMBIANA DE INGENIERÍA

PROGRAMACIÓN ORIENTADA A OBJETOS

HERENCIA - EXCEPCIONES

S07- S09 : 2024-2

Memoria

Presente el mapa de memoria correspondiente a:

- En el juego se tienen dos flotas “LA GRAN FLOTA BLANCA” y “LA GRAN ARMADA DE CASTILLA”
- “LA GRAN ARMADA DE CASTILLA” tienen un barco (número 900); un porta avión (identificado con 333) y un avión (identificado con 'chkgood'). El avión corresponde al porta avión 333 y actualmente está en el aire.

Algunos métodos. Diseñando y codificando

Implementen algunos de los siguientes métodos de la clase Flota

Method Summary	
void	alNorte() Mueve la flota una posición al norte. El mundo tablero es circular. Longitud [0,180] Latitud [-90,90]. Coordenadas.
void	avance(int dLon, int dLat int) Mueve todas las máquinas la distancia definida El mundo tablero es circular. Longitud [0,180] Latitud [-90,90]. Coordenadas. Parameters: dLon - avance en longitud dLat - avance en latitud
ArrayList<Maquina>	seranDestruídas(int longitud, int latitud) Consulta las máquinas que pueden afectarse por una explosión en la posición dada. En una coordenada pueden estar muchas máquinas. Los aviones en aire no se destruyen.
ArrayList<Maquina>	maquinasDebiles() Consulta las máquinas débiles de una flota Un barco es débil si tienen menos de cinco marinos; un avión, si no tiene piloto principal; y un portaaviones, si es un barco débil o alguno de sus aviones en aire es débil. Returns: Las máquinas debiles
public boolean	esBuenAtaque(int longitud, int latitud) Verifica si una ubicación de ataque es adecuada (destruye elementos enemigos sin ocasionar bajas propias) Parameters: longitud - longitud de la explosion latitud - latitud de la explosion
void	ataquen(int lon, int lat) Mueve todas las máquinas que no son débiles paso a paso (uno a uno) hacia la posición a atacar indicada por (lon, lat) Parameters: lon - longitud lat - latitud

Nueva máquina. Diseñando y codificando.

Las flotas ahora van a tener cápsulas submarinas, sin tripulantes, que piden instrucción a una máquina nodriza. Las máquinas nodrizas pueden ser un barco u otra cápsula. Las capsulas alcanzan profundidades superiores a (8.000 mts) que las hace inmunes a ataques y NUNCA son débiles.

- Presente el diseño estructural
- Analice si debe modificar el diseño de los métodos desarrollados.
- Diseñe los métodos que requieren cambios.
- Realice la codificación completa de esta nueva clase.

Nuevo comportamiento. Diseñando y codificando

Los marinos, aviones y cápsulas están preparados para auto-destruirse. Los marinos y barcos se auto-destruyen si reciben la instrucción. Las cápsulas se auto-destruyen si su nodriza es destruida. Todos deben poder informar la causa por la que tomaron la decisión. Las flotas conocen todos los elementos que se han auto destruido.

- Presente el diseño estructural
- Realice la codificación completa del nuevo elemento.

Otros métodos. Diseñando y codificando.

Implementen los métodos de la clase Flota y Tablero

Method Summary Flota	
void	alNorte() throws BatallaNavalExcepcion Mueve la flota una posición al norte. En el momento que una máquina no se puede mover se para el movimiento de toda la flota. Throws: BatallaNavalExcepcion - si alguna de las máquinas no se pudo mover al norte.
ArrayList<Marino>	pilotos() throws BatallaNavalExcepcion Consulta los pilotos de la flota Returns: Los pilotos asignados a los aviones de las flotas Throws: BatallaNavalExcepcion - si un piloto no es marino de la flota, un piloto de un avión asignado a un portavion no es marino del portaavión o un piloto está asignado a más de un avión
int	potencia() throws BatallaNavalExcepcion Consulta la potencia de la flota. La potencia es el número de máquinas no débiles. Returns: La potencia de las flotas Throws: BatallaNavalExcepcion - si existen menos marinos que máquinas en la flota

Method Summary Tablero	
int	alNorte() Consulta el número de flotas que lograron un movimiento completo
ArrayList<Flota>	infiltrados() throws BatallaNavalExcepcion Consulta las flotas que tienen pilotos infiltrados Returns: Las flotas con pilotos infiltrados Throws: BatallaNavalExcepcion - si alguna flota no tiene marinos asignados
int	potencia() throws BatallaNavalExcepcion Consulta la potencia del tablero La potencia es la suma de la potencia de las flotas Returns: La potencia del tablero Throws: BatallaNavalExcepcion - si más de la mitad de las flotas tienen problemas de potencia.

