

Proyecto Estructuras de Datos (Grupo B): Batalla Naval por Consola

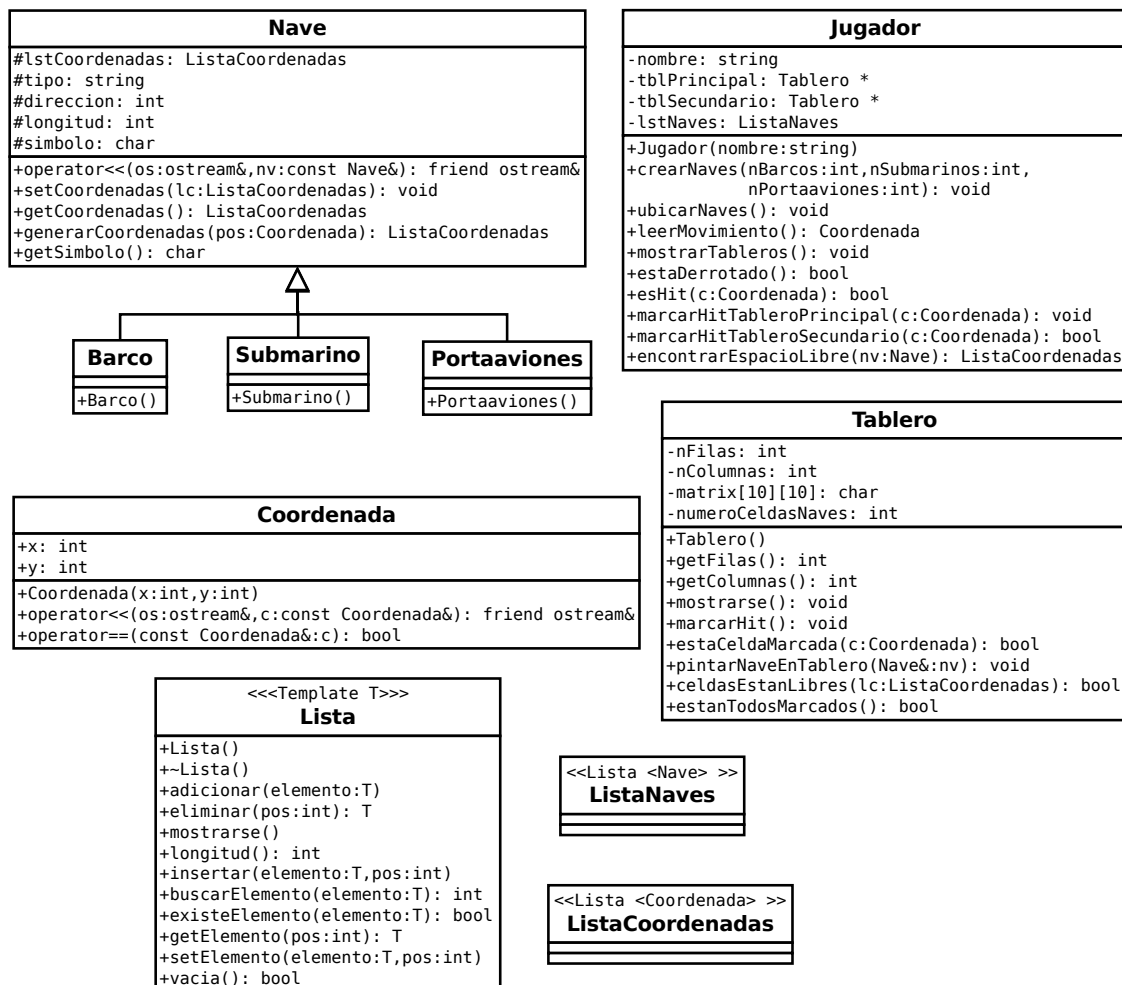
Prof. Luis E. Garreta U.

November 5, 2017

Pontificia Universidad Javeriana - Cali

1 Introducción

Usted debe implementar el juego batalla naval para consola. La estructura básica que va a utilizar son las listas simplemente enlazadas con características polimórficas, es decir la lista puede contener objetos de diferentes tipos de datos, en este caso va a existir una lista de naves donde va a almacenar objetos de la clase *Barco*, *Submarino*, y de la clase *Portaaviones*. El diagrama de las clases involucradas en el juego se muestra a continuación:



2 Implementación

La implementación del juego se debe realizar incremental, como en los anteriores proyectos, es decir primero implementa la lógica del juego en el programa principal (*mainbn.cpp*), luego implementa la clase *Lista* con apuntadores y templates, y finalmente implementa las demás clases. Igualmente, la implementación de cada clase se hace incremental, es decir, deriva su propia clase. por ejemplo de la clase *Tablero* deriva la clase *MiTablero* y empieza a sobrescribir los métodos uno por uno para irlos probando a medida que los termine.

Para todo esto usted tiene los archivos de encabezado (.h) y los archivos objetos (.o) de todas las clases de la librería. También tiene una implementación de listas (listas.h) que usa la clase *list* de STL. Usted debe reemplazarla por su propia lista con apuntadores y que debe usar templates.

El orden sugerido para la implementación del proyecto es el siguiente

1. Implementar el programa principal (*mainbn.cpp*) usando la librería de clases de batalla naval.
2. Implementar la clase *Lista* con listas simplemente enlazadas usando nodos y crear las clases *ListaCoordenadas* y *ListaNaves* usando templates.
3. Derivar de la clase *Nave* sus propias subclases *MiBarco*, *MiSubmarino*, y *MiPortaaviones*, sobrescribir el constructor de cada subclase. Después reemplazar estas nuevas clases por las viejas donde se utilicen.
4. Derivar de la clase *Nave* su propia clase *MiNave* y derivar de esta las subclases creadas en el punto anterior. Después sobrescribir los métodos.
5. Derivar de la clase *Tablero* su clase *MiTablero* y sobrescribir los métodos.
6. Derivar las clases restantes.

3 Documentación y Recursos

En los encabezados de cada clase (.h) está la descripción de lo realiza cada función, junto con lo que le ingresa como parámetros y el valor de retorno. Cualquier otra inquietud, la resolvemos en clase o me preguntan el correo.

- Los archivos cabecera (.h) y objetos (.o) están en el repositorio <http://github.com/lgarreta/puj-bnaval>. Allí también está el ejecutable *mainnb.exe* para que pruebe la funcionalidad del juego.
- Existen dos versiones: una para linux y otra para windows. Cada versión trae un Makefile listo para que compile los programas con la utilidad *make* (tanto para linux como para windows).
- Si quiere usted puede crear el proyecto en devcpp adicionando creando un proyecto, adicionando los encabezados (.h) y adicionando los objetos (.o) dentro de las opciones de proyecto, en la parte de parámetros.

4 Entregas

- Martes 7: Nov: Implementación lógica del juego (*mainbn.cpp*).
- Jueves 9: Nov: Implementación de Listas con Nodos sin Templates
- Martes 14 Nov: Implementación de Listas con Nodos con Templates.
- Jueves 16 Nov: Implementación Juego usando las Listas con Nodos.
- Martes 21 Nov: Implementación Final