ÍNDICE

ACTIVIDAD 7.6 Barcos 1/2

Enunciado

Se pretende desarrollar un conjunto de clases que represente la jerarquía de barcos de una flota. A continuación, se describen las características básicas de cada uno de los tipos de barcos:

Barco. Clase general que describe a un barco. Incluye sus datos básicos que son **nombre, matrícula**, **año de construcción**, así como la **fecha en la que fue adquirido por la flota** (*utiliza la clase* `Data`) y el **número de horas de mantenimiento** que se le han dedicado al barco desde que fue construido.

Tendrá, al menos, las siguientes funciones miembro:

- **Constructor** para definir correctamente un barco a partir de su nombre, matrícula y año de construcción. Las horas de mantenimiento serán 0 de inicio. Se establecerá la fecha en la que fue adquirido al día actual (*usando la clase Data*).
- Se sobreescribirá el **método `toString`** para que muestre la siguiente información del barco (tipo de barco, nombre, matrícula, año de construcción y horas de mantenimiento, en ese orden). Fíjate en el ejemplo de ejecución para más detalles acerca del formato de la cadena que debes retornar.
- **Método `realizarMantenimiento`**: Este método aumentará las horas de mantenimiento en 100 horas cada vez que sea llamado.

Barco de guerra. Es un barco que tendrá, además de las características de un barco estándar, un **listado de armamento** (es decir, un conjunto de armas; no es necesario que crees una clase Arma, simplemente se guarda la descripción del arma) disponible que el barco lleva a bordo, el **número de tripulantes máximo,** así como el **número de tripulantes actual**. En este caso, al ser un barco tan especial, se invierte en cada mantenimiento un total de 300 horas.

Tendrá, al menos, las siguientes funciones miembro:

- **Constructores** (debe rellenar la información básica y los datos concretos de este tipo de barco; aunque en el caso del armamento, se debe permitir crear barcos tanto con dos armas de inicio, como con tres).
  - Si se proporciona un número de tripulantes actual superior al máximo, se establecerá como número de tripulantes actual el valor máximo de ellos.
  - Un barco no tiene límite de armas a llevar a bordo.
  - Si alguna de las armas que se proporcionan son iguales, solo se añadirá la primera ocurrencia.
- Método **`aumentarTripulantes`**, que deberá aumentar el número de tripulantes actual en un número proporcionado. Se impone la misma lógica que al crear el barco en caso de que los actuales superen el máximo.
- Método **`aumentarMaximoDeTripulantes`**. Se proporcionará en cuántas personas se quiere aumentar el máximo de tripulantes del barco.
- Método **`anyadirArmamento`**. El método debe recibir una lista de armamento que deberá añadirse al armamento actual (si alguna de las armas ya existe, no se añadirá ya que no puede haber armas repetidas).
- Deberás sobreescribir el **`método toString`** para que imprima sus datos básicos (los de Barco), el tipo de barco, el listado de armas a bordo, así como el número de tripulantes actual y los concretos).
- También deberás sobreescribir el método **`realizarMantenimiento`**.

Barco de pesca. Es un barco del que se registra su **capacidad de carga** (en metros cúbicos), la **lista de redes disponibles** para realizar la faena pesquera, así como el **porcentaje de capturas exitosas**. A estos barcos se les dedica en cada revisión el **doble de las horas** que se le dedica a un barco estándar.

Tendrá, al menos, las siguientes funciones:

- **Constructor** (debe rellenar la información básica por ser un barco, así como los datos referentes a la capacidad y el porcentaje de capturas exitosas). De inicio un barco no dispone de redes de pesca.
- Deberás **sobreescribir el `método toString`** para que imprima sus datos básicos, la lista de redes que lleva encima y el porcentaje de capturas exitosas.
- Al igual que el barco anterior, también deberá sobreescribir el método **`realizarMantenimiento`**.
- Dar de **alta una nueva red de pesca (método `anyadirNuevaRed`)**: Las redes de pesca vendrán representadas por una clase `Red`. De ellas necesitamos almacenar el tamaño (pequeña, mediana o grande), material de la red (pueden ser de poliamida, poliéster o polietileno) y tipo de pesca para la que se utiliza esa red (que puede ser

pesca spinning, submarina o carpfishing). Se permite que un barco tenga redes de características iguales.

- Dar de **baja una red (método `eliminarRed`)**: Recibirá una red y si existe esa red en el listado de redes la eliminará (*para simplificar la solución, se pondrá esa red a null*). Se considera que una red es igual a otra si todas sus características coinciden.

A continuación, implementa un programa (en un fichero `TestBarcos`) que muestre cómo funciona. Deberás crear, al menos, un barco de cada tipo y probar que el método `realizarMantenimiento` funciona correctamente según el tipo de barco. Además, deberás probar el alta y baja de redes de pesca, así como la inclusión de armas en los barcos de guerra.

Organiza tu jerarquía de clases, **introduciendo todas ellas en un paquete** (por ejemplo, lo puedes llamar `tiposbarcos`) para evitar el acceso a elementos de visibilidad `protected` desde otras clases (en este caso, desde `TestBarcos`, la cual quedará fuera de ese paquete).

Evita en la medida de lo posible la creación de getters y setters si hay una alternativa a su creación, así como la utilización de visibilidad protected y/o public cuando éstas no son estrictamente necesarias.

Posible ejemplo de ejecución

```
---- Se crean tres barcos, uno de cada tipo. En el barco de guerra,
comprueba que funciona tanto la eliminación de armas repetidas, como la
creación de un barco de guerra con más tripulantes actuales que máximo.
También deberás comprobar que el barco pesquero no tiene ninguna red dada
de alta ----

Tipo de barco: BARCO, nombre: La molinera, matrícula: XXXX, año de
construcción: 2005, horas de mantenimiento: 0

Tipo de barco: BARCO DE GUERRA, nombre: Arrasator, matrícula: YYYYY, año
de construcción: 2017, horas de mantenimiento: 0, armas a bordo: [Lanza
torpedos], número de tripulantes: 0

Tipo de barco: BARCO DE PESCA, nombre: Faenero, matrícula: ZZZZZ, año de
construcción: 2001, horas de mantenimiento: 0, redes: [], porcentaje de
éxito en captura: 40%


---- Incrementa las horas de mantenimiento de los tres barcos, y del
pesquero una vez más ----

Tipo de barco: BARCO, nombre: La molinera, matrícula: XXXX, año de
```

C/ Societat Unió Musical, 8 - 03802 Alcoi
Tel. 966 52 76 60 - Fax 966 52 76 61
03012165.secret@gva.es
www.cipfpbatoi.es


```
construcción: 2005, horas de mantenimiento: 100

Tipo de barco: BARCO DE GUERRA, nombre: Arrasator, matrícula: YYYYY, año
de construcción: 2017, horas de mantenimiento: 300, armas a bordo: [Lanza
torpedos], número de tripulantes: 0

Tipo de barco: BARCO DE PESCA, nombre: Faenero, matrícula: ZZZZZ, año de
construcción: 2001, horas de mantenimiento: 400, redes: [], porcentaje de
éxito en captura: 40%


---- Añade 2 redes diferentes al barco pesquero y un armamento compuesto
por 2 armas iguales al barco de guerra, pero diferente a la que ya tiene
registrada ----

Tipo de barco: BARCO DE GUERRA, nombre: Arrasator, matrícula: YYYYY, año
de construcción: 2017, horas de mantenimiento: 300, armas a bordo: [Lanza
torpedos, Misil de crucero], número de tripulantes: 0

Tipo de barco: BARCO DE PESCA, nombre: Faenero, matrícula: ZZZZZ, año de
construcción: 2001, horas de mantenimiento: 400, redes: [Red de tamaño
mediano hecha de poliamida para pesca carpfishing, Red de tamaño pequeño
hecha de poliéster para pesca submarina], porcentaje de éxito en captura:
40%

---- Barco de pesca después de eliminar su primera red ----

Tipo de barco: BARCO DE PESCA, nombre: Faenero, matrícula: ZZZZZ, año
de construcción: 2001, horas de mantenimiento: 400, redes: [null, Red
de tamaño pequeño hecha de poliéster para pesca submarina], porcentaje
de éxito en captura: 40%

-- Prueba también que la gestión del número de tripulantes de los
barcos de guerra funciona, siguiendo la misma metodología ---
```