El general MacMapkerson ha decidido encargarnos el desarrollo de una parte del sistema de defensa, se necesita un sistema para poder gestionar los ejércitos.

Estos ejércitos estarán compuestos por tres cuerpos o divisiones principales:

* Caballería: Compuesta por vehículos de ataque rápido y de transporte de tropas.
* Infantería: Compuesto por soldados a pie con diferente tipo de armamento.
* Artillería: Compuesto por cañones de diferente tipo y funcionalidad.

Mac, nos ha encargado la creación de una estructura base de programación con el fin de poder saber diferentes características de los ejercicitos compuestos por esos tres grandes tipos de elementos militares.

 Mac quiere evaluar lo que el llama la Capacidad Militar o CM sobre cada uno de los ejércitos de los que pueda disponer,

Interfaz de usuario gráfica, Texto, Aplicación, Word, Correo electrónico

Descripción generada automáticamente

Siendo:

* CM: Capacidad Militar: Fuerza del ejército al enfrentarse a otro oponente.
* Pf: Potencia de fuego: Capacidad de destrucción que puede llevar a cabo el ejército.
* Cm: Capacidad de movimiento: Capacidad de movilidad del ejército.
* Bl: Blindaje: Resistencia al fuego enemigo, capacidad de defensa o sanación.

Por otro lado, hay que tener en cuenta que cada ejército dispone de un fondo limitado para crear sus componentes, eso implica que cada uno de los elementos militares va a tener una propiedad denominada precio. y cada ejercito podrá tener un fondo limitado

Hemos decidido estructurar nuestra jerarquía mediante la creación de unas interfaces para poder determinar que capacidades tiene cada uno de los elementos militares que conformarán el ejército.

**IDestructor** (Las clases que implementan esta interface estarán indicando su capacidad de poder tener una fuerza destructiva determinada)

Public interface IDestructor.

Public double capacidadDeDestruccion();

**IMovil (**Las clases que implementan esta interface están indicando su capacidad de poder tener una velocidad de movimiento)

Public interface IMovil

Public doublé capacidadDeMovimiento();

**IBlindado** (Las clases que implementan esta interface están indicando su capacidad de poder tener una capacidad de resistencia al ataque del ejército enemigo)

Public interface IBlindado

Public doublé resistenciaAlAtaque();

El general nos ha proporcionado una serie de elementos de ejemplo, con el fin de que nosotros podamos crear la primera estructura de clases para su proyecto.

CABALLERIA

|  |  |
| --- | --- |
|  | Transporte: MX-7899.  Velocidad: 4.5  Blindaje: 1.4  Potencia de Fuego: No  Precio: 4.200 € |
|  | Tanque de ataque Sombras-VB98  Velocidad: 7.3  Blindaje: 4.8  Potencia de Fuego: 9.8  Precio: 15.600 € |
|  | Transporte rápido TAXIN-66  Velocidad: 12  Blindaje: No  Potencia de Fuego: No.  Precio: 1.600 € |

 INFANTERIA

|  |  |
| --- | --- |
|  | Infantería Básica  Velocidad: 6  Blindaje: No  Potencia de Fuego: 7  Precio: 250 € |
|  | Ametrallador  Velocidad: 4  Potencia de Fuego: 10  Blindaje: No  Precio: 400 € |
|  | Sanitario  Velocidad: 7  Blindaje: 5  Potencia de Fuego: No.  Precio: 500€ |

 ARTILLERIA

|  |  |
| --- | --- |
|  | Cañon Antiaereo  Velocidad: 1  Blindaje: No  Potencia de Fuego. 22  Precio: 1.100 € |
|  | Torpedero móvil.  Velocidad: 3  Blindaje:2  Ptencia de Fuego: 19  Precio: 1.350 € |
|  | Cañon  Velocidad: No  Blindaje: No  PotenciaDeFuego: 14  Precio: 1.100 € |

El General quiere:

* Que hagamos una jerarquía de unidades militares, basándonos en una interface que tiene que devolver el blindaje, el movimiento, el precio y la potencia de fuego.
* Que creemos un factory method como ha hecho el cadete Jacinto para crear las unidades, nos devolverá la unidad correcta según el número si nos dan el numero 1 el TransporteMX, el primero de la lista y así sucesivamente.

Que probemos con un main que funciona.