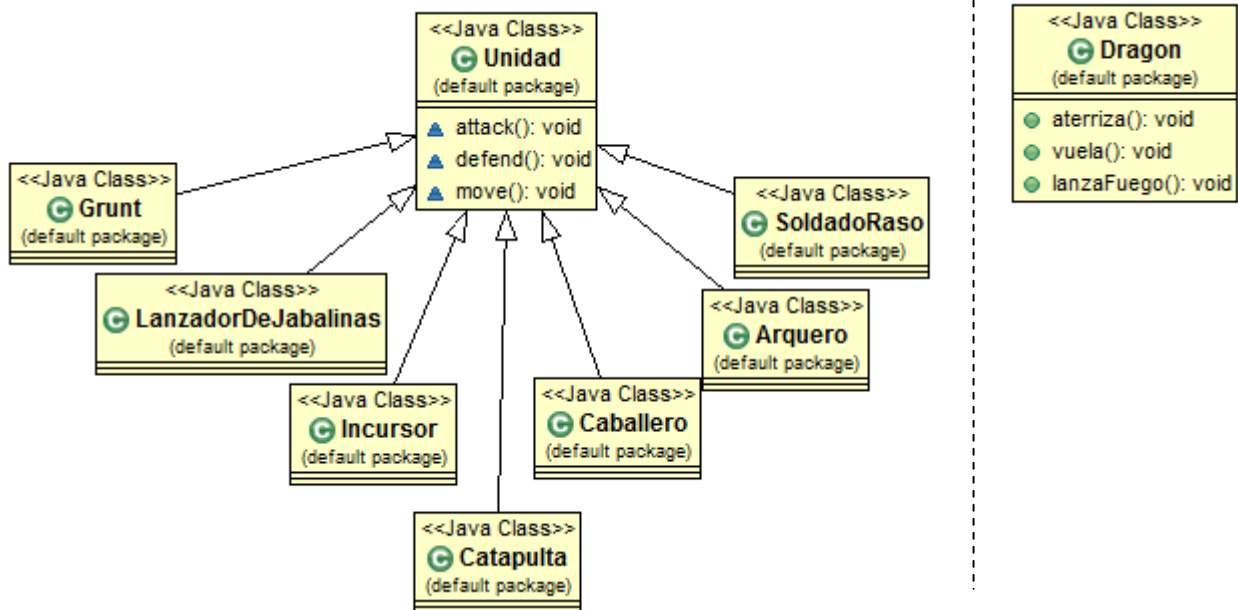


El juego Warcraft: Humanos vs Orcos® es un juego de estrategia donde dos especies combaten entre sí: los humanos y los orcos. Cada Especie dispone de los mismos tipos de unidades militares, aunque particularizada para cada especie. Para representar en memoria estos personajes se ha decidido utilizar la siguiente jerarquía:



Un programador colega nuestro nos proporciona una clase Dragon que nos gustaría incluir al conjunto de unidades de nuestro Juego. Sin embargo esta clase nueva no respeta la interfaz de Unidad así que no podemos hacer que herede directamente de ella. Para ello decidimos que un dragón actuando como Unidad debe:



- Atacar: echando a volar y a continuación lanzando fuego.
- Defenderse: atacando (para eso es un dragón)
- Moverse: volando y luego aterrizando.

Se pide:

1º/ Indica cual sería el patrón adecuado para poder crear y manipular dragones como un tipo más de unidades. **Justifica tu respuesta.**

2º/ ¿Qué clases/interfaces habría que crear y como se combinan con Unidad? **Justifica tu respuesta. (2 puntos)**

3º/ Implementa este nuevo tipo de Unidad dragón con el comportamiento indicado en el enunciado, e indicando las clases de las que hereda y las interfaces que implementa: **(1,5 puntos)**

Una vez implementada esta clase de unidad, tras varias partidas de prueba los jugadores beta-testers concluyen que estas unidades Dragon son excesivamente poderosas. Así que los diseñadores del juego deciden que a lo largo de una partida sólo puede haber un dragón en juego, y aunque aparezca en distintas partidas será siempre el mismo dragón (así por ejemplo si hubiese resultado herido, al volver a aparecer éste seguirá con el mismo daño). Los diseñadores nos piden una solución que cumpla los tres requisitos siguientes:

- Que el dragón sea siempre el mismo durante toda la ejecución del juego.
- Evitar que el resto de programadores puedan (por error) crear distintos dragones
- Como además el código del juego está dividido en múltiples clases, que se pueda acceder a ese objeto dragón desde cualquier método/clase del código del juego.

Se pide:

4º/ Indica cual sería el patrón adecuado para manejar de esta manera el dragón y justifica tu elección. **(1,5 puntos)**

5º/ Define la jerarquía de objetos necesaria para implementar esta funcionalidad. Describe e implementa los métodos que tendrá cada nueva Interfaz/Clase. Indicando además quién y cómo se cubren los tres requisitos expuestos. Implementa las operaciones. **(3 puntos)**

6º/ Completa el siguiente ejemplo de código con dos clases distintas C1 y C2, cada una con un método donde se requiere al mismo dragón y se invoca a uno de sus métodos como unidad. **(1 punto)**

```
class C1{
    void m1 () {
        // definimos una variable "d" que contendrá el dragón

        // obtenemos el dragón

        // lo utilizamos para atacar
        d.attack();
    }
}

class C2 {
    void m2 () {
        // definimos una variable "d" que contendrá el dragón

        // obtenemos el dragón

        // lo utilizamos para moverlo
        d.move();
    }
}
```