

Consigna para el modelado de la Tierra Media. Competencia por puntos, 2018a TPA

 consigna.md

Competencia de programación

Modelando combates en la Tierra Media

Modelaremos el comportamiento de los personajes de la Tierra Media, en su modo más combativo. Todas las unidades comienzan con una salud de 100 puntos.

- El enano ataca (10pts) cada vez con más fuerza (2 x ataques dados). Recibe ataques normalmente. Si descansa, recupera hasta el 25% de su salud inicial.
- El humano ataca (10 pts) cada vez con menos fuerza (-1 x ataques dados), a menos que descanse (recupera la fuerza inicial). Al recibir ataques pierde un poco de fuerza (-1 por ataque recibido).
- El elfo ataca (5pts) con más fuerza cada vez que recibe ataques ($+2^{\text{ataques recibidos}}$). Si descansa, se calma y se cura completamente, pero sus ataques se debilitan nuevamente.
- El goblin ataca (3pts) débilmente, pero es muy difícil de golpear (1 de cada 10 veces lo golpean). Si descansa, se cura completamente.
- El orco ataca (15pts) con mucha fuerza. Los golpes no le hacen mucho efecto a su salud (1/4 del daño). No le sirve de nada descansar.
- El troll ataca (30pts) con muchísima fuerza, pero 2 de cada 3 veces pifia el golpe. Si descansa, se vuelve de piedra y no puede atacar más. Los golpes no le hacen daño.

Material proporcionado

Se proveen a modo de ejemplo, las pruebas básicas sobre el Enano. Pueden probarse más aspectos si se considera necesario.

```
import org.junit.Assert;
import org.junit.Test;

public class EnanoTests {

    @Test
    public void enanoAtaca() {
        Personaje enano = new Enano();
        Personaje orco = new Orco();

        Assert.assertEquals(100, orco.salud());
        enano.atacar(orco);
        Assert.assertEquals(95, orco.salud());
        enano.atacar(orco);
        Assert.assertEquals(88, orco.salud());
        enano.atacar(orco);
        Assert.assertEquals(79, orco.salud());
    }

    @Test
    public void enanoDescansa() {
        Personaje enano = new Enano();
        Personaje orco = new Orco();

        Assert.assertEquals(100, enano.salud());
        for(int i=0; i<6; i++) {
            orco.atacar(enano);
        }
    }
}
```

```
    }  
    Assert.assertEquals(10, enano.salud());  
  
    enano.descansar();  
    Assert.assertEquals(35, enano.salud());  
    enano.descansar();  
    Assert.assertEquals(60, enano.salud());  
    enano.descansar();  
    Assert.assertEquals(85, enano.salud());  
    enano.descansar();  
    Assert.assertEquals(100, enano.salud());  
}  
  
}
```

Cómo se calificará

El conjunto de pruebas (funcionando) que cubran completamente los ataques de una unidad, valdrá 1XP por unidad cubierta. Asimismo, el conjunto que cubra (funcionando) completamente las defensas de la misma recibirá 1XP. Esto suma 12XP por la resolución de todas las combinaciones.

Por el simple hecho de participar, ya tienen garantizados 2XP por las pruebas sobre el Enano.

Quien complete el ejercicio con un buen diseño orientado a objetos, pruebas adecuadas, y prolijidad del código recibirá 10XP adicionales.

Se trabajará en grupos de 3 o 4 personas *presentes*. No se admitirán entregas de grupos que no cumplan con esa condición. Se prestará especial atención para detectar copias en las resoluciones. En caso de hacerlo, se considerarán no válidas ambas entregas (aún repitiendo código entre diferentes turnos).

Se recolectarán los ejercicios 30 minutos antes de terminar la clase, por la carpeta compartida especificada por los docentes. *Pasados 5 minutos, no se aceptarán más entregas.* La calificación estará disponible para la clase siguiente.