Laboratorio de programación orientada a objetos
Práctica de laboratorio 1

Importante: Las prácticas de laboratorio deben ser realizadas de forma individual, se debe trabajar solamente en la práctica ya que el navegar, chatear, mensajear o realizar actividades que no tengan nada que ver con la práctica implica la NO aprobación de la misma.
Es necesario anotar la hora de inicio y finalización de cada una de las prácticas.
Al terminar la práctica dar aviso al instructor para que sea revisada.

Práctica 11. Implementación de una clase abstracta e interfaces.

En esta práctica se modificaran las clases anteriores y se implementará una clase abstracta haciendo uso de características avanzadas del diseño de clases.

Realizar commit y subir a GitHub cada 15 minutos

El trabajo en esta práctica consiste en:

Paso 1. Crear una copia de la práctica 8 en la carpeta llamada practica11 (no crear sub carpetas con prácticas anteriores).

Paso 2. Modificar la clase **Personaje** para que sea una clase **abstracta** y el atributo **vida** será protegido.

Todos los métodos **decVida** y **addVida** serán abstractos.

Modificar los métodos getDetalle por toString (en todas las clases que la contengan)

Paso 3. En el paquete **utils** crear la clase ***enumerada*** avanzada **Escudo** con los valores SUPER, MEDIO, BAJO, NULO con los valores enteros 0, 1, 2, 3 respectivamente; el valor numérico se llamará **nivel**.

Paso 4. En la clase **Planta**, modificar el atributo **escudo** para que sea de tipo **Escudo** del paquete **utils**.

Los constructores que no reciben escudo será por default NULO, el método toString regresa además de nombre y vida el nivel de escudo.

Los métodos **decVida** restar la vida menos el nivel de escudo; sí tiene argumento restar la vida menos el nivel de escudo por el argumento recibido.

Los métodos **addVida** sumar la vida más el nivel de escudo; sí tiene argumento sumar la vida más el nivel de escudo por el argumento recibido.

Paso 4. En el paquete **utils** crear la interface **Muerto** con el atributo MAX_ZOMBIES y valor de 10 y la definición del método comer; no regresa ningún valor.

Paso 5. En la clase **Zombie** implementar la interface **Muerto** modificar el método getDetalle por **toString**.

Compilar y llamar al instructor para instrucciones.

Paso 6. En la clase **PruebaHerencia** compilar y hacer las correcciones necesarias para quitar todos los errores.

Las instancias previas eran:

new Personaje("David",100)
new Personaje("Bianca")
new Planta("Fabian",10,'B')
new Planta("Almendra",50)
new Planta("Ricardo",'C')
new Planta("Silvia")
new Zombie("Armando",80,false)
new Zombie("Josseline",true)
new Zombie("Eduardo")

Para las instancias Fabian y Ricardo el escudo deberá ser Medio y Bajo respectivamente.

Paso 7. Después de corregir los errores ejecutar, la salida deberá ser parecida a la siguiente:

Nota: Para compilar y ejecutar utilizar los modificadores -d y -cp respectivamente.

```
David   0       3
Soy planta NULO
3
David   9       3
***** Objeto 1 *****
Bianca  3       false
Soy zombie false
4
Bianca  3       false
***** Objeto 2 *****
Fabian  10      1
Soy planta MEDIO
0
Fabian  10      1
***** Objeto 3 *****
Almendra        50      3
Soy planta NULO
4
Almendra        62      3
***** Objeto 4 *****
Ricardo3        2
Soy planta BAJO
7
Ricardo17       2
***** Objeto 5 *****
```

Silvia 3 3
Soy planta NULO
1
Silvia 6 3
***** Objeto 6 *****
Armando 80 false
Soy zombie false
6
Armando 62 false
***** Objeto 7 *****
Josseline 3 true
Soy zombie true
6
Josseline 3 true
***** Objeto 8 *****
Eduardo 3 false
Soy zombie false
6
Eduardo 3 false
***** Objeto 9 *****