

Programación 2 - Certamen N° 3

Eduardo Godoy.

14 de agosto de 2018

Nombres:

Ruts:

Puntaje

Nota

Resultados de aprendizaje a evaluar:

1. Conceptos generales de Programación Orientada a Objetos en Java.
2. Utilización del Lenguaje Java y sus características principales para resolver problemas.
3. Diseño e implementación de interfaces de usuario.
4. Implementación y explotación de fuentes de persistencia de información.
5. Uso de programación con multi-hilos o multi-threads en Java.

Instrucciones:

- El puntaje máximo del certamen es 100 puntos, siendo el 60 el mínimo requerido para aprobar.
- El certamen es resuelto en grupos de 3 personas.

Contenido:.

Tema	Puntaje total	Puntaje obtenido
Java aplicación de características del lenguaje para solucionar un problema	30 puntos	
Java Swing y Manejo de Archivos	30 puntos	
Evaluación de conceptos de Orientación a Objetos y aplicación de estos a la resolución del problema	40 puntos	

Puntaje total: 100 puntos **Exigencia:** 60 puntos.

1. Java Threads (30 pts)

Un videojuego tiene **Personajes** los cuales se enfrentan en una batalla definida por turnos. Cada personaje tiene un *nombre* (String) y un *nivel propio de energía* (int). Además poseen la capacidad de *alimentarse* (método), que recibe por parámetro una cantidad de energía (int) con el que incrementa el *nivel propio de energía*. Los personajes pueden ser:

■ Guerreros:

- Tienen además un *arma* (String). La cual posee 3 tipos de ataques:
 - Golpe directo: Daña al oponente restandole 20 puntos de energía.
 - Giro letal: Daña al oponente de forma reiterada restandole 30 puntos de energía.
 - Super arma: El arma triplica su tamaño dañando al oponente y restandole 70 puntos de energía. A su vez el guerrero se resta puntos de vida equivalentes al 30 % del daño realizado.
 - Pierde turno: El oponente logra eludir el ataque del guerrero.
- Cada uno de los ataques anteriores se ejecutan de forma aleatoria y uno por turno, siendo el 0 Golpe directo, 1 Giro letal, 2 Super arma, 3 Pierde turno.
- Estos ataques se ejecutan mediante la invocación del método *combatir()*.
- Al momento de la instanciación del Guerrero, este recibe su *nombre* y *arma*.
- Los guerreros son siempre creados con un *nivel propio de energía* igual a 150.

■ Magos:

- Tienen además un *hechizo* (String). El cual posee 3 niveles de daño definidos a continuación:
 - Llamada: El mago envía una flama dañando al oponente y restandole 20 puntos de energía.
 - Incendio: Incinera al oponente y su entorno dañándolo y restandole 35 puntos de energía. Además se emite una jugada adicional aleatoria entre 0 y 1 siendo 0 no afecta con mas daño y 1 afecta con 5 puntos más al daño final.
 - Explosión: El mago emite una explosión dañando al oponente restandole 60 puntos de energía. Al ejecutar esta habilidad el mago puede recuperar energía ejecutando el método *alimentarse()* por el 50 % del daño realizado al oponente.
 - Pierde turno: El mago falla la ejecución del ataque.
- Cada uno de los ataques anteriores se ejecutan de forma aleatoria y uno por turno, siendo el 0 Llamada, 1 Incendio, 2 Explosión, 3 Pierde turno.
- Estos ataques se ejecutan mediante la invocación del método *encantar()*.
- Al momento de la instanciación del Mago, este recibe su *nombre* y *hechizo*.
- Los magos son siempre creados con un *nivel propio de energía* igual a 100.
- Poseen el método *encantar* que disminuye en 2 unidades el *nivel propio de energía* cada vez que se ejecuta un ataque exitoso.

■ Otras Reglas Generales:

- El combate termina cuando la energía de uno de los dos personajes sea menor o igual a 0.
- El estado del combate se debe visualizar por consola mediante el siguiente formato: *Personaje;tipo de ataque;daño realizado*.
- La generalización de las clases se muestra en la figura 1.
- Desarrolle las clases e interface mostradas en el modelo.
- Implemente los Java Threads asociados al Mago y Guerrero, los cuales simularan el combate permitan ejecutar los ataques de cada uno.

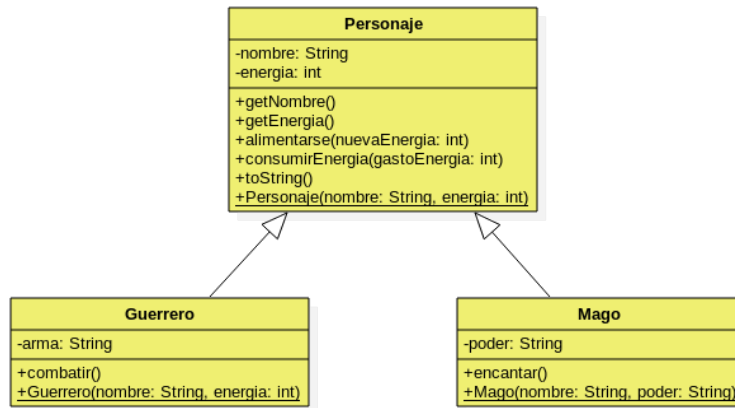


Figura 1: Diagrama de clases

- Implemente la clase que maneje los estados de energía de los personajes y permita ejecutar sus ataques de aleatorios según lo indicado anteriormente.
- implemente una clase que escriba en dos archivos lo siguiente:
 - El primero llamado ataque.csv debe agregar por cada turno un registro con el daño realizado según el siguiente formato: *Personaje;tipo de ataque;daño realizado*.
 - El segundo llamado danio.csv debe agregar por cada turno un registro con el daño recibido según el siguiente formato: *Personaje;daño realizado*.
- Implementar la clase que debe utilizar los métodos para *combatir* y *alimentarse*.
- La clase principal que permita instanciar personajes de tipo guerreros y magos. Luego iniciar la batalla.

¿Cómo será evaluado en la pregunta 1?			
Tópico	Logrado	Medianamente logrado	No logrado
Tópico 1 - Java Interface .	5pts Crea las Interface Persona con sus métodos requeridos.	2pts Crea Interface con algunos métodos requeridos en el. Crea métodos o Atributos en otras clases no indicadas en el problema.	No crea clases requeridas.
Tópico 1-a - Clase Guerrero	3pts Define e implementa la Clase Guerrero con sus métodos relacionados	2pts Define e implementa método acercandose parcialmente a la salida esperada.	No define método o definido pero no cumple con lo mínimo esperado.
Tópico 1-b - Clase Mago	5pts Define e implementa la Clase Mago con sus métodos relacionados.	2pts Define e implementa método acercandose parcialmente a la salida esperada.	No define método o definido pero no cumple con lo mínimo esperado.
Tópico 2 - ThreadMago	5pts Define e implementa la clase Thread para simular el ataque de Mago con sus métodos y atributos requeridos.	2pts Define e implementa la clase acercandose parcialmente a la salida esperada.	No define clase o definida pero no cumple con lo mínimo esperado.
Tópico 3 - ThreadGuerrero	5pts Define e implementa la clase Thread para simular el ataque de Guerrero con sus métodos y atributos requeridos.	2pts Define e implementa la clase acercandose parcialmente a la salida esperada.	No define clase o definida pero no cumple con lo mínimo esperado.
Tópico 4 - Clase Combate	3pts Define e implementa la clase de que permite coordinar el combate de los Personajes simulación del combate con sus método main incluidos.	2pts Define e implementa clase y método acercandose parcialmente a la salida esperada.	No define método o definido pero no cumple con lo mínimo esperado.
Tópico 5 - Clase Inicializadora	2pts Define e implementa la clase de control que permite iniciar el proceso de simulación del combate con sus método main incluido.	2pts Define e implementa clase y método acercandose parcialmente a la salida esperada.	No define método o definido pero no cumple con lo mínimo esperado.
Paradigma Orientación a Objetos	2pts Resuelve el problema utilizando OO y patrones de diseños presentados en clase.	1pts Utiliza parte del POO y patrones de diseño para resolver el problema.	No utiliza el POO para dar solución al problema.
Total máximo puntaje	30pts	15pts	hasta 10 pts

2. Java Swing y Archivos (30 pts)

Una vez finalizado el combate y utilizando los archivos generados a partir del resultado de la pregunta 1, realizar lo siguiente:

- Generar una interfaz con Java Swing que permita visualizar:
 1. En su parte superior: Dos botones con los nombre respectivos de cada personaje.
 2. En su parte inferior: Dos Tabla de Datos.
 3. Al presionar un boton, éste recuperará los registros desde el archivos asociados al personaje que referencia el boton.
 4. En la tabla de datos izquierda se debe desplegar el daño realizado y en la derecha el daño recibido.
 5. En el desarrollo de esta interfaz se requiere implementar los patrones de diseño vistos en clases.
 6. Incluir buenas practicas de programación.

¿Cómo será evaluado en la pregunta 2?			
Tópico	Logrado	Medianamente logrado	No logrado
Manipulación de archivo.	5pts Lee correctamente los archivos, los mapea a entidad y crea ArrayList.	3pts Realiza dos de las tres acciones del punto anterior.	No realiza la acciones del punto anterior.
Diseño e implementación de interfaz.	10pts Crea la interfaz de forma correcta y con los componentes requeridos.	5pts Crea la interfaz de forma correcta e implementa algunos componentes requeridos.	No crea interfaz.
Integración Interfaz con contenido de archivos	10pts Asocia acciones a botones y despliega registros en tablas según lo requerido en el problema.	5pts Crea la clase entidad para Movie o la Rating (no ambas).	No crea las clases entidad.
Paradigma Orientación a Objetos	5pts Implementa la solución según patrones de dise o y programas vistos en clases.	2pts Implementa la solución sin patrones de diseño.	No se implementa según lo indicado.
Total máximo puntaje pregunta 2	30pts	15pts	hasta 10 pts

3. Interrogación (40 pts)

Interrogación individual sobre codificación y aplicación de conceptos dentro de la solución entregada.

1. Pregunta de manejo de conceptos aplicados a la solución implementada.
2. Pregunta de codificación 1.
3. Pregunta de codificación 2.

¿Cómo será evaluado en la pregunta 3?			
Tópico	Logrado	Medianamente logrado	No logrado
Pregunta de manejo de Conceptos aplicados a la solución.	15pts Responde de forma correcta según lo esperado.	8pts Responde de forma incompleta o con apoyo del profesor.	No responde según lo esperado.
Pregunta de codificación 1.	15pts Responde de forma correcta según lo esperado.	7pts Responde de forma incompleta o con apoyo del profesor.	No responde según lo esperado.
Pregunta de codificación 1.	10pts Responde de forma correcta según lo esperado.	5pts Responde de forma incompleta o con apoyo del profesor.	No responde según lo esperado.
Total máximo puntaje pregunta 2	40pts	20pts	hasta 10 pts

Condiciones de entrega:

- Debe compilar, en caso contrario se tendrá nota mínima 1.0 y no tendrá derecho a interrogación.
- Subir proyecto a Aula Virtual como archivo comprimido, eliminado de ante mano los archivos .class.
- Se debe incluir dentro del comprimido un archivo de tipo .txt que contenga nombre apellido y rut de ambos integrantes del grupo.
- Se debe incluir dentro del comprimido un archivo README.txt con instrucciones de instalación/compilación.
- Entrega el Lunes 27 de Agosto, hasta las 17:55hrs por Aula Virtual.
- El no cumplimiento con la hora de entrega, será penalizado con 2 punto de descuento por cada minuto de retraso.
- Si bien el trabajo colaborativo esta permitido, la presunción o evidencia de copia entre dos o más grupos será penalizada con 30 puntos de descuento de la nota final para cada uno.