

Programa con abstracción: Videojuego

Resumen

Este programa simula un videojuego en el que eres un aventurero buscando un tesoro, para encontrarlo debes enfrentarte a tres enemigos, el jugador y los enemigos derivan de la clase abstracta entidad que tiene puntos de vida, puntos de ataque y puntos de defensa, los cuales son usados para calcular las interacciones en el combate.

Clases

Entidad.java

Clase abstracta base para el jugador y el enemigo, contiene las estadísticas del combate, sus setters y getters, la función para calcular el daño de combate (85% a 100% del atk), la función para defender en combate con el doble de defensa y los métodos abstractos presentación y derrota que imprimen mensajes al inicio y fin del combate.

Jugador.java

Hereda los movimientos y estadísticas básicos de la entidad, la presentación y despedida del jugador que es la definición de los métodos abstractos de entidad y los movimientos exclusivos del jugador que aumentan sus características.

Enemigo.java

Segunda clase abstracta que contiene la lógica para decidir la acción del enemigo contiene el método para realizar el movimiento que afecta al jugador y un método abstracto para decidir y anunciar dicho próximo movimiento, también contiene un movimiento especial que es una interfaz que se le asigna a una clase enemigo y un entero que señala el próximo movimiento.

Slime, Fantasma y Dragon.java

Clases hijas de enemigo.java que contienen la lógica para el cálculo del movimiento enemigo y los mensajes que se despliegan para cada uno.

Interfaces

MovimientoEspecial.java

Interfaz que contiene el método ejecutaMovimientoEspecial que toma a un jugador y afecta sus estadísticas dependiendo de la clase que herede la interfaz.

La clase MovimientoSlime reduce la defensa del jugador por 2 puntos.

La clase MovimientoFantasma reduce la defensa y el ataque por 2 puntos.

La clase MovimientoDragon reduce la defensa y el ataque por 3 puntos.

Esta interfaz se le asigna dinámicamente a cada enemigo en la simulación

Simulación

La simulación comienza generando un Jugador y un Slime, se le asigna el movimiento Slime al Slime y comienza un combate.

Si al terminar el combate el jugador ganó se crea un Fantasma y se le asigna su movimiento.

Si al terminar ese combate el jugador ganó se crea un Dragón y se le asigna su movimiento.

Si se gana ese combate se despliega un mensaje de felicitaciones.

Combate

El método combate recibe una instancia de jugador y un objeto que heredó de la clase enemigo, los cuales llaman a su mensaje de presentación, las estadísticas del jugador se almacenan para aumentarlas en caso de que gane y empieza el ciclo de combate hasta que alguna de las entidades tenga hp menor a 0.

1. Se despliega las estadísticas del jugador y el hp del enemigo.
2. Se almacena la variable que indica el siguiente movimiento el enemigo junto con un mensaje en la consola que sirve como pista para cuál es.
3. Se despliegan las opciones de la acción del jugador y se espera a recibir la elección del usuario, hay cuatro opciones:
 - 3.1. El jugador ataca, si el movimiento del enemigo es defensa se usa el doble de su estadística de defensa para calcular el daño, si no, se usa el valor normal, después del cálculo se reduce el hp del enemigo en el número resultante.
 - 3.2. El jugador se defiende, si el enemigo ataca el cálculo del daño toma el doble de su defensa.
 - 3.3. El jugador usa su método de aumentar ataque en un valor aleatorio de 0 a 5.
 - 3.4. El jugador usa su método de aumentar defensa en un valor aleatorio de 0 a 5.
4. Si el enemigo no ha sido derrotado se activa su turno, y se usa el entero siguiente movimiento en su método realizar movimiento para decidir su acción
 - 4.1. Si el movimiento es 1, se calcula el daño (con el doble de defensa si el jugador escogió defensa) y se reduce el hp del jugador de acuerdo al resultado.
 - 4.2. Si el movimiento es 2 ya fue tomado en cuenta en el turno del jugador.
 - 4.3. Si el movimiento es 3 adicionalmente al efecto del ataque, se usa la interfaz de movimiento especial para afectar las estadísticas del jugador.

Al final del combate si el hp del jugador es menor a 0 se activa su mensaje de derrota y se regresa esa misma instancia y ya que es 0 no se activan los siguientes combates, pero si el hp del enemigo es 0 se reestablecen las estadísticas del jugador a las del inicio del combate y se aumentan para el siguiente, se regresa esa instancia.

Conclusión

Este programa usa la abstracción para realizar el combate del jugador con cualquier tipo de enemigo usando el mismo código para cada combate, además se usó una interfaz para

asignar a cada enemigo un movimiento personalizado que se active con la misma función del enemigo.