Tema 5 - Práctica

1. *(1,5)* Realiza las siguientes modificaciones a las clases implementadas en las prácticas anteriores:

- Para la clase **Personaje**:
  - ★ Añade dos nuevos atributos: *res* para representar la resistencia mágica o defensa frente a ataques de tipo mágico, y *vel* para representar la velocidad del Personaje.
  - ★ Cambia el nombre del atributo *def* a *arm* para referir a la armadura o defensa física.
  - ★ Getters y setters para los nuevos atributos.
  - ★ Los constructores cambiarán su implementación: ahora, los constructores por parámetros que no dan valores a las estadísticas (*pv*, *atq*, *arm*, *res*, *vel*) les darán un valor por defecto inicial: 100 para *pv* y 10 para todos los demás atributos.
  - ★ El constructor que inicializa todos los atributos con valores específicos debe incluir los nuevos atributos.
  - ★ El método subirNivel() en su implementación por defecto tiene ahora una probabilidad del 50% de incrementar cada estadística por separado. Es decir, con cada subida de nivel, hay un 50% de probabilidad de que aumente *pv*, otro 50% de que aumente *atq*, y así sucesivamente. Cada probabilidad debe calcularse por separado.
  - ★ El método defender() deberá incluir un parámetro extra que indique el tipo de daño que se está recibiendo: físico para los ataques estándar o mágico para los ataques de tipo mágico.
  - ★ Implementa un método realizarTurno() que solicitará por teclado acciones al usuario. Las posibles acciones se describen más adelante.

- Para la clase **Combate**:
  - ★ Ahora la estadística que rige que Personaje golpea primero es la velocidad *vel*.
  - ★ Si un Personaje tiene más del doble de la velocidad que otro, podrá realizar dos veces.
  - ★ El primero de los parámetros del método combatir() se considerará el Personaje jugable, y como tal, se le ofrece la posibilidad de escribir por teclado que acción realizará de entre las presentes en el método realizarTurno() de la clase Personaje:
    - → **Atacar**. Se realiza el ataque estándar.
    - → **Acción Especial**. Esta opción solo aparecerá a los Personajes con las habilidades correspondientes, y variará según el tipo de Personaje.
    - → **Defender**. Durante un único turno, se incrementan los valores de def y res en un 20% de su valor actual.
    - → **Pasar turno**. No se realiza ninguna acción.

Ahora se comenzarán a escribir los diferentes tipos de Personaje como sus **subclases**. Ten en cuenta que todas las subclases tienen detalles que deberás sobrescribir e implementar correctamente. Además, la probabilidad de aumentar una estadística concreta cuando suben de nivel cambia según la descripción proporcionada.

2. *(1)* Escribe una subclase **Guerrero** que herede de Personaje. El Guerrero tiene una probabilidad del 80% de mejorar su ataque y éste incrementa el doble de lo habitual, y del 75% de vida y armadura; sin embargo, solo tiene un 20% de posibilidades de mejorar su resistencia mágica. Su velocidad no presenta ni ventajas ni penalizaciones. Además, todo Guerrero tiene un atributo adicional, "furia", que cuando está activo duplica temporalmente la estadística de ataque del guerrero, pero que provoca que al defenderse reciba también el doble de daño.
    - Añade el atributo furia a la clase.
    - Añade un constructor por defecto, un constructor por parámetros, y getters. Cuando sea pertinente, utiliza los valores de la superclase.
    - Escribe un método modificarFuria() que active o desactive la furia en función de su estado actual.
    - Sobrescribe el método subirNivel() de la clase Personaje para que los incrementos de estadísticas se correspondan con las ventajas y penalizaciones descritas.
    - Actualiza los métodos atacar() y defender() para tener en cuenta el modificador de furia.
    - Modifica el método toString para describir el estado del Guerrero.

3. *(1,25)* Escribe una subclase **Mago** que herede de Personaje. Un Mago tiene un atributo adicional, *mag*, que representa una cantidad de puntos de magia que determinan su destreza utilizando hechizos, que parte de un valor de 10 y con cada nivel tiene un 85% de probabilidades de aumentar. Los Magos solo tienen una probabilidad de mejora del 35% de vida y armadura, y del 15% de ataque; sin embargo, sus probabilidades de aumento son del 80% para la resistencia mágica y del 65% para la velocidad. Además, tienen la posibilidad de lanzar conjuros además de atacar como el resto de Personajes.
    - Añade el atributo magia a la clase.
    - Sobrescribe el método subirNivel() de la clase Personaje para que los incrementos de estadísticas se correspondan con las ventajas y penalizaciones descritas.
    - Añade un constructor por defecto, un constructor por parámetros, y getters. Cuando sea pertinente, utiliza los valores de la superclase.
    - Añade un método lanzarConjuro() que, dependiendo del valor recibido por teclado, permita al mago lanzar los siguientes hechizos al objetivo:
        - ★ Bola de fuego (1): lanza una bola de fuego incandescente que causa un 70% de sus puntos de magia como daño a un enemigo.
        - ★ Escudo arcano (2): se protege a sí mismo o a un aliado, aumentando su estadística de armadura y resistencia mágica el 50% de los puntos de magia del mago.
        - ★ Céfiro (3): un fuerte viento causa un 30% de sus puntos de magia como daño a todos los enemigos presentes.
        - ★ Presteza mental (4): agiliza las reacciones propias o de un aliado, aumentando su velocidad un 100% de los puntos de magia del mago.
    - Modifica el método toString para describir el estado del Mago.

4. *(1)* Escribe una subclase **Ladrón** que herede de la clase Persona. Un Ladrón tiene una probabilidad de incremento al subir de nivel del 85% de velocidad el doble de lo habitual, y del 60% de ataque; pero se reduce al 40% en todos los demás atributos.
    - Sobrescribe el método subirNivel() de la clase Personaje para que los incrementos de estadísticas se correspondan con las ventajas y penalizaciones descritas.
    - Añade un constructor por defecto, un constructor por parámetros, y getters. Cuando sea pertinente, utiliza los valores de la superclase.
    - El Ladrón tiene una acción especial, Robar. Esta funcionalidad se implementará al completo en prácticas posteriores. Por ahora, crea el método que la contendrá, e implementa como funcionalidad que el Ladrón golpee con su velocidad en lugar de con su ataque.
    - Modifica el método toString para describir el estado del Ladrón.

5. *(1,5)* Escribe una subclase **Cazador** que herede de Personaje. Un Cazador tiene únicamente una ventaja en la posibilidad de incrementar su velocidad del 70%, pero trae consigo un Compañero Animal que comparte parcialmente sus atributos y ataca con él.

- Implementa al **Compañero Animal** como una clase anidada dentro de Cazador. Un Compañero Animal sigue siendo un Personaje. El atributo raza del compañero animal contendrá su especie, que puede ser de tres tipos con sus respectivas variaciones en atributos respecto al cazador:
  - ★ Cánido: 20% de todos los atributos.
  - ★ Felino: 30% de ataque y velocidad, 15% de todos los demás.
  - ★ Rapaz: 35% de velocidad, 25% de resistencia mágica, 15% de ataque, 5% de vida y armadura.

    Esto quiere decir que un cánido siempre tendrá un 20% de todas las estadísticas del Cazador al que acompaña, y su subida de nivel **no** incrementará sus estadísticas de forma independiente.
- Sobrescribe el método subirNivel() de la clase Personaje para que los incrementos de estadísticas se correspondan con las ventajas y penalizaciones descritas. En el caso del Compañero Animal, modifícalo para ajustarse a su Cazador.
- Añade un constructor por defecto, un constructor por parámetros, y getters. Cuando sea pertinente, utiliza los valores de la superclase.
- Modifica el método atacar() para que los ataques del Compañero Animal se sumen a los del Cazador. El método defender() no se ve afectado.
- Modifica el método toString para describir el estado del Cazador. Recuerda que debe incluir el estado del Compañero Animal, por lo que es recomendable implementar también su toString().

6. *(0,75)* Escribe una subclase abstracta **Creyente** que herede de Personaje. Los Creyentes tienen un atributo adicional, *fe*, una cantidad de puntos de fe que determinan su destreza haciendo milagros. Sus bonificaciones y penalizaciones varían en función del tipo de Creyente.
    - Añade el atributo fe a la clase.
    - Añade un constructor por defecto, un constructor por parámetros, y getters. Cuando sea pertinente, utiliza los valores de la superclase.
    - Añade un método abstracto plegaria() que permitirá al creyente efectuar milagros en su turno según lo introducido por el usuario a través del teclado. Los tipos de plegaria cambiarán en función del tipo de Creyente.
    - Modifica el método toString para describir el estado del Creyente.

7. *(1)* Escribe una subclase **Paladín** que herede de Creyente. Un Paladín es un tipo de Creyente que combate cuerpo a cuerpo, por lo que tienen una posibilidad de incrementar su armadura del 70% el doble de lo habitual, y aumentar su vida en un 5% adicional según una probabilidad del 50%; asimismo tienen un 60% de probabilidades de aumentar su ataque. Sin embargo, la velocidad solo tiene un 15% de probabilidades de incrementar, y la fe un 30%.
    - Añade un constructor por defecto, un constructor por parámetros, y getters. Cuando sea pertinente, utiliza los valores de las superclases.
    - Sobrescribe el método subirNivel() de la clase Personaje para que los incrementos de estadísticas se correspondan con las ventajas y penalizaciones descritas.
    - Sobrescribe el método plegaria() con los posibles milagros de los que dispone un paladín:
        - ★ Imbuir arma (1): el paladín añade el 80% de sus puntos de fe a sus puntos de ataque.
        - ★ Baluarte de fe (2): el paladín aumenta su propia armadura un 30% de sus puntos de su fe.
        - ★ Fogonazo sagrado (3): el paladín ciega a su oponente, reduciendo su velocidad y resistencia mágica un 40% de sus puntos de fe.
    - Modifica el método toString para describir el estado del Paladín.

8. *(1)* Escribe una subclase **Clérigo** que herede de Creyente. Un Clérigo es un tipo de Creyente que combate a distancia y sana a sus aliados, por lo que tienen una posibilidad de incremento de sus estadísticas de fe y resistencia mágica del 80%, y estas aumentan el doble de lo habitual. Sin embargo, su vida y armadura solo tienen un 20% de probabilidades de incrementar, y su ataque un 10%.

- Añade un constructor por defecto, un constructor por parámetros, y getters. Cuando sea pertinente, utiliza los valores de las superclases.
- Sobrescribe el método subirNivel() de la clase Personaje para que los incrementos de estadísticas se correspondan con las ventajas y penalizaciones descritas.
- Modifica el método plegaria() con los posibles milagros de los que dispone un paladín:
  - ★ Sanación (1): sana el 70% de sus puntos de fe como vida a un aliado.
  - ★ Rezo sagrado (2): sana el 35% de sus puntos de fe como vida a todo el grupo.
  - ★ Cólera divina (3): causa el 55% de sus puntos de fe como daño sagrado a un objetivo.
- Modifica el método toString para describir el estado del Clérigo.

9. *(1)* Crea una subclase **Monstruo** que herede de Personaje. Un monstruo será un Personaje enemigo cuyas estadísticas varían en función del tipo (raza) de monstruo.

- Añade un constructor por defecto, un constructor por parámetros, y getters. Cuando sea pertinente, utiliza los valores de las superclases.
- Las ventajas y desventajas de los diferentes tipos de Monstruo afectan a sus estadísticas de la siguiente manera:
  - ★ Bestia: 80% de aumentar el doble de lo habitual en ataque y velocidad, sin modificadores para los puntos de vida, y el 15% en armadura y resistencia mágica.
  - ★ No-muerto: un 70% de aumentar cuatro veces lo habitual en resistencia mágica, sin modificadores para los puntos de ataque, un 30% en vida y armadura, un 5% en velocidad.
  - ★ Gigante: sus puntos de vida aumentan el 100% de las veces pero con la misma probabilidad de aumentar el doble o el triple de lo habitual, sin modificadores en armadura, y un 10% en velocidad y resistencia mágica.
- Modifica el método toString para describir el estado del Monstruo.