EJERCICIO 1

APARTADO A

Crea la clase **Flecha**. De una flecha necesitamos saber su longitud en centímetros y el material de su punta. Ambos privados.

Una punta puede estar hecha de hierro, piedra o hueso. Para hacer más sencillo el ejercicio asignaremos números a cada material: 1 (hierro), 2 (piedra) y 3 (hueso).

Al crear una flecha es necesario indicar su longitud y el número del material del que está hecha su punta *(ver nota 1 a pie de página).* [1]

También se podrá crear una flecha sin necesidad de indicarle nada al constructor. En ese caso siempre se creará una flecha de 56 ctms con punta de piedra.

Implementa además los métodos:

- `cambiarPunta(int material)`: cambia el material de la punta de la flecha por el indicado en el parámetro *(ver nota 1 a pie de página).*
- `toString()`: obtiene la cadena de la siguiente forma: *"Flecha de X ctms con punta de M"* Siendo X los centímetros de la flecha y M el material de la flecha (escrito con palabras, nada de número).

APARTADO B

Crea la clase **Arco**. De un arco necesitamos saber su longitud en centímetros, su peso en gramos y el material del que está hecho: madera, metal o fibra de vidrio. Para hacer más sencillo el ejercicio asignaremos números a cada material: 1 (madera), 2 (metal) y 3 (fibra). *(ver nota 1 a pie de página).*

Además la clase va a tener un carcaj. Un carcaj es una Bolsa o caja en forma de tubo, que se empleaba para llevar flechas.

El constructor de esta clase sólo necesitará el material del que está hecho el arco y la capacidad del carcaj (el número de flechas que puede guardar). Además tendrá en cuenta lo siguiente:

- Si un arco es de madera, pesará 300 gramos y tendrá una longitud de 900 centímetros.
- Para un arco de metal, su peso será de 530 gramos y su longitud será de 750 centímetros.

- Un arco de fibra de vidrio pesará apenas 100 gramos y su longitud será de 1100 centímetros.

Cuando se crea un arco, el carcaj de flechas se llena automáticamente con flechas de longitud aleatoria entre 40 y 60 centímetros y con punta de material aleatorio.

Implementa los siguientes métodos:

- `disparar()`: si hay flechas en el carcaj, se mostrará por pantalla *"Se ha disparado una flecha* descripción del objeto flecha" *y* 'eliminará' una flecha de la lista. Si no hay flechas, indicará que no se ha podido disparar por falta de flechas.
- `recargar()`: rellena completamente el carcaj de flechas con flechas de longitud y material aleatorios.

EJERCICIO 2

APARTADO A

Las señales son hechizos simples que los brujos aprenden durante su formación en *Kaer Morhen*. De una **señal** se necesita saber su nombre, su daño, su energía y su nivel.

- ✔ Además, cada señal puede ser de un tipo: quinética, fuego, viento, hipnótica. Una señal siempre será de uno de los tipos antes indicados. *Nota: la implementación de los tipos se debe realizar con las cadenas indicadas. NO está permitido el uso de números.*
- ✔ El nivel de la señal es la maestría que tiene el brujo con ella. Será un número decimal e influirá en otros parámetros.
- ✔ La energía hace referencia a la cantidad de poder mágico que consume esa señal. Será un número con decimales.
- ✔ El daño es la cantidad de daño que hace la señal al ser lanzada. Depende de su tipo. Será un número con decimales.
- ✔ El nombre de la señal no tiene ningún misterio, no?? just a name!

Menos la propiedad daño que tendrá el modificador de visibilidad por defecto , todas las demás propiedades serán privadas.

El constructor de la señal sólo tendrá como parámetros el nombre y el tipo de señal *(no hace falta que se controle si el tipo introducido es correcto o no. Eso es algo que se pide en el apartado B)* y rellenará los demás atributos de la siguiente forma:

- Todas las señales empiezan en el nivel 1.
- La energía será el resultado de multiplicar un numero aleatorio entre 5 y15 por la división del nivel de la señal entre 5.
- El daño depende del tipo de señal:
  - Si es de fuego: un numero aleatorio entre 50 y 100 por la división del nivel de la señal entre 10.
  - Si es de viento: un numero aleatorio entre 25 y 50 por la división del nivel de la señal entre 10.
  - Si es hipnótica o quinética, el daño es 0.

Aparte del constructor, debes crear el método `toString`. El cual debe devolver la cadena con el formato siguiente:

Señal: *nombre de la señal.*
Tipo: *tipo de la señal.*
Nivel: *nivel de la señal.*
Daño: *daño de la señal* puntos.
Consume *energía de la señal* puntos.

Ejemplo:

```
Señal: Igni.
Tipo: Fuego.
Nivel: 1.
Daño: 45 puntos.
Consume 12 puntos.
```

APARTADO B

Mejora el constructor de la clase Señal controlando que se introduce un tipo de señal correcto. En caso de que no sea correcto el tipo introducido, asigna el tipo 'hipnótica' por defecto.

APARTADO C

Ahora crea la clase **Brujo**. Un brujo tendrá un nombre, una edad y lugar de nacimiento. Además todo brujo tendrá una señal que podrá usar y una cantidad de magia. La magia es un valor numérico con decimales que va gastándose a medida que el brujo lanza señales. Todas las propiedades serán privadas.

Para crear un brujo es necesario saber su nombre, su lugar de nacimiento y la señal que va a usar. La edad será un número aleatorio entre 18 y 55. La cantidad de magia de un brujo será un valor aleatorio entre 90 y 180.

APARTADO D

Un brujo puede conjurar la señal contra otro brujo. Este método recibirá un brujo y devolverá una cadena. Su funcionamiento es el siguiente:

- si el brujo tiene magia suficiente, se le resta la energía de la señal a su magia y se lanza la señal (crea un mensaje indicando que se ha lanzado la señal con nombre XXXX)

    - A continuación, se le resta el daño de la señal a la magia del otro brujo (crea un mensaje indicando el daño que ha hecho la señal y añádelo al mensaje anterior).
    - Finalmente devuelve ese mensaje.
- Si no hay magia suficiente, devuelve un mensaje indicándolo.

Nota: No hay que controlar si los brujos están vivos o muertos, ni si uno mata al otro, ni nada más. Sólo lo que se indica en los puntos anteriores.

Nota: añade los métodos que necesites en las clases Señal y Brujo.

Ejemplo de ejecución del método conjurar:

Si hay magia suficiente

```
Gerald ha lanzado la señal Ignis…
… Besemir ha recibido 45 puntos de daño.
```

Si NO hay magia suficiente

```
Gerald no puede lanzar su señal!!!
```

APARTADO E

Crea le método `toString`, el cual devolverá toda la información del brujo (señal incluida) con el siguiente formato:

Este brujo es *nombre* **de** *lugar_nacimiento*. **Tiene** *edad* **años y** *magia* **puntos de magia. Utiliza la señal:** *información de la señal*

Ejemplo:

```
Este brujo es Gerald  de Rivia. Tiene 38 años y 160 puntos
de magia. Utiliza la señal:
Señal: Igni.
Tipo: Fuego.
Nivel: 1.
Daño: 45 puntos.
Consume 12 puntos.
```