

Java 8: ¿Qué es un descriptor funcional? Pon un ejemplo. **(1 punto)**

Patrones: Como sabéis, la interfaz `Iterator<E>` tiene los siguientes métodos:

- `E next()`
 - Retorna el siguiente elemento de la iteración
 - Lanza `NoSuchElementException` si éste no existe
- `boolean hasNext()`
 - Indica si existe un elemento siguiente en la iteración (asegurando que cuando retorna `true`, `next()` no lanzará ninguna excepción)
- `remove()`
 - Para el caso que nos ocupará la única cosa que necesitáis saber es que lanzará `UnsupportedOperationException`

Lo primero que se pide **(2 puntos)** es implementar una clase, llamada `LinesIterator`, que implemente la interfaz `Iterator<String>` usando la referencia a `BufferedReader` que se le pasa como parámetro en el constructor y que podéis suponer que no es nula.

El método relevante de `BufferedReader` (para simplificar no consideraremos `close`) es:

- `public String readLine() throws IOException`
 - Retorna la siguiente línea del fichero de entrada o `null` si ya se ha llegado al fin del fichero.
 - Puede lanzar `IOException` si ha habido algún problema al leer del fichero. Recordad que `IOException` es una excepción comprobada (*checked*).

Tenemos también la siguiente clase:

```
1 public abstract class SequenceProcessor<E> {
2     public void processSequence(Iterator<E> iterator) {
3         while (iterator.hasNext()) {
4             E element = iterator.next();
5             processElement(element);
6         }
7     }
8     protected abstract void processElement(E element);
9 }
```

¿Qué patrón sigue este código? Justificad la respuesta en función de la estructura del código de la clase i la del patrón **(1 punto)**.

Implementad dos subclases de `SequenceProcessor<String>` **(1 punto cada una)** tales que:

- **Counter:** calcula el número de elementos que hay en la secuencia (un `int`)
- **MeanLength:** calcula la media de las longitudes de las cadenas de la secuencia (un `double`)

Si queremos calcular varias cosas usando sendos `SequenceProcessor<E>` sobre una misma secuencia, lo podemos hacer, pero realizamos varios recorridos de la misma, lo que puede ser costoso. ¿Qué patrón (y porqué) usaríais para solucionar este problema? Implementad vuestra solución y mostrad el diagrama de clases general **(3 puntos)**. Aplicad vuestra solución al caso de, dado un `BufferedReader`, calcular y después escribir, tanto el número de líneas como la media de sus longitudes **(1 punto)**.