

3ª Práctica – IO

La finalidad de la presente práctica es ejercitar el uso de entradas y salidas. Lo haremos para dos aspectos distintos: por un lado simplemente usar clases disponibles, y por otro crear una clase propia de IO. Para ello nos apoyaremos en la práctica anterior extendiéndola.

1) Uso de clases disponibles.

Se trata de que la clase concreta que implementa `Typography`, sea capaz de construirse a partir de un fichero de texto conteniendo los patrones de los caracteres. El formato del fichero puede ser simplemente una secuencia de patrones donde cada uno de ellos se define en una línea previa mediante el carácter que representa, como se ve en la figura (a diferencia de la tipografía de la práctica anterior, en la que se especificaban símbolos de fondo y de frente, en este caso hay total libertad para usar los símbolos que se deseen en cada punto).

```
O
.*****.
*.....*
*.....*
*.....*
*.....*
*.....*
.*****.
I
*****
...*...
...*...
...*...
...*...
...*...
...*...
*****
```

En constructor llevará por tanto un parámetro con el nombre del fichero que contiene los grafemas (que razonablemente incluirá info de su tamaño, p.ej. "Simple7.typ").

2) Creación de una clase de IO.

En esta segunda parte crearemos una clase de IO de tipo "procesamiento", que nos ofrecerá la posibilidad de escribir texto usando una de nuestras tipografías con grafemas. Se llamará por tanto razonablemente "GraphemeWriter" y extenderá "FilterWriter", teniendo que reescribir los métodos de esta, y comportándose como un Wrapper.

De este modo podremos usar nuestros grafemas en cualquier escritura. P.ej. para escribir por la salida estándar sería:

```
Typography t =new MyTypography("Simple7.typ");
GraphemeWriter gw=new GraphemeWriter(new OutputStreamWriter(System.out), t );
gw.write("HolaMundo",0,2);
gw.flush();
```

O para hacerlo a un fichero:

```
Typography t =new MyTypography("Simple7.typ");
GraphemeWriter gw=new GraphemeWriter(new FileWriter("HolaMundoBanner.txt"), t );
gw.write("HolaMundo",0,2);
gw.flush();
```

La naturaleza de la acción que queremos realizar supone lo siguiente:

Cuando se escribe un carácter o una secuencia, no se puede construir una representación multilínea de la misma, sino que hay que mantener en memoria la secuencia de grafemas a escribir hasta que aparezca un carácter '\n'. Es en ese momento cuando se escribe la representación de una línea de texto en grafemas que equivale a varias líneas de texto reales.

Por otro lado, no todos los caracteres son representables, pero esto podemos obviarlo por el simple hecho de que representaremos únicamente aquellos caracteres que estén incluidos en nuestra tipografía.

Ya como un "problema" más complejo nos encontraríamos frente a qué hacer cuando se llama al método flush(). Queda planteado como "problema abierto"...

Nota.- No debe dejarse de leer la documentación de FilterWriter para entender qué y cómo debe extenderse.