



Universidad de Oviedo

Editor de texto con macros

Introducción

En el código inicial descargado encontraréis el prototipo de un editor de texto básico.

Nuestro editor permite **abrir ficheros**, **añadir texto** al final del documento, **borrar la última palabra** y **sustituir** una palabra por otra. A continuación puede verse un ejemplo de funcionamiento.

```
> open quijote.txt
```

En un lugar de la mancha de cuyo nombre

```
> insert no quiero acordarme
```

En un lugar de la mancha de cuyo nombre no quiero
acordarme

```
> delete
```

En un lugar de la mancha de cuyo nombre no quiero

```
> replace a x
```

En un lugxr de lx mxnchx de cuyo nombre no quiero

Tareas a realizar

Se desea añadir la posibilidad de **grabar macros**, de tal forma que se simplifique la tarea de tener que escribir las mismas acciones varias veces.

```
> record m1  
> delete  
> insert final  
> replace i y  
> stop
```

```
> open quijote.txt
```

En un lugar de la mancha de cuyo nombre

```
> execute m1
```

En un lugar de la mancha de cuyo fynal

```
> open regenta.txt
```

La heroica ciudad dormía la siesta. Vetusta,

```
> execute m1
```

La heroyca cyudad dormía la syesta. fynal

Deberéis modificar el código entregado para que realice la tarea descrita.

Detalles de implementación

- Si durante la grabación de una macro se vuelve a grabar, la grabación anterior simplemente se descarta y se empieza una nueva.
- Mientras se está grabando las acciones se continúan ejecutando normalmente en el documento abierto.
- Abrir un documento modifica el estado del editor, no del documento, y por tanto no es una acción que deba grabarse.
- Durante la grabación de una macro puede ejecutarse una macro previamente grabada.

Se puede utilizar la prueba incluida al final del documento para comprobar su correcto funcionamiento.

Tareas adicionales

Se debe indicar en un documento de texto en la carpeta raíz del proyecto el patrón o **patrones de diseño** utilizados en su resolución, estableciendo la **correspondencia entre los participantes de la estructura genérica del patrón** y las clases e

interfaces de vuestra solución, así como los métodos relevantes de cada uno y sus atributos importantes, si es el caso.

En el proyecto inicial puede verse un fichero de prueba de cómo sería dicha correspondencia para un ejemplo tomado de las diapositivas de clase. Puede usarse dicho documento como una plantilla para llevar a cabo dicha tarea.

Criterios generales de evaluación

- Se pide, de entre todos los diseños posibles, aquella solución que, ante futuras versiones, requiera el mínimo número de modificaciones en la aplicación (y que el lugar donde hacer estos cambios, de ser necesarios, sea evidente). Aquellos patrones que se introduzcan de manera innecesaria penalizarán en la evaluación.
- Igualmente, un diseño o una implementación que sean innecesariamente complejos y dificulten la comprensión del código también serán penalizados.
- Los nombres elegidos en vuestra solución (clases e interfaces, métodos, variables...) son importantes y, como tal, serán evaluados.
- Penalizará gravemente el que en la solución haya elementos que muestren falta de dominio en programación o en el paradigma orientado a objetos.

- Es condición necesaria que la aplicación funcione correctamente (no se admitirán soluciones que fallen en ejecución o no realicen total o parcialmente las tareas solicitadas).
- Todo código que entregue el alumno será evaluado. Por tanto, todo aquello que consideréis que sea temporal, redundante, innecesario o incorrecto deberá ser borrado antes de la entrega final (en caso contrario, se supondrá que el alumno presenta dicho código proponiéndolo como correcto y se penalizará como tal). Esto incluye también el código comentado.

Prueba de funcionamiento

```
> record m1
> replace a x
> insert aa
> stop

> record m2
> insert bbb
> execute m1
> stop

> open quijote.txt
En un lugar de la mancha de cuyo nombre
> execute m1
En un lugxr de lx mxnchx de cuyo nombre bbb aa
```