

Convocatoria ordinaria de Programación I

Examen 1 (22/11/2019)

DURACIÓN 3 horas

Nombre:

Apellidos:

Hora entrega: Código ordenador:

Notas:

- Rellene sus datos personales y el código del ordenador donde realiza el examen, y al finalizar el examen la hora de entrega. Entréguelo al profesor antes de salir.
- Para la realización del examen **SÓLO SE PUEDE ACCEDER** a la documentación disponible en la [página de la asignatura](#) y a la documentación en línea de [oracle.com](#) que incluye la especificación de las clases y los tutoriales.
- **No se permite llevar consigo ningún dispositivo electrónico durante el examen. Las pertenencias deben guardarse en el lugar indicado del aula y los móviles deben estar apagados.**

Enunciado

Se desea desarrollar un conjunto de clases que represente los elementos de un menú de una aplicación informática. Dichas clases representarán los distintos tipos de elementos (**Element**) según su finalidad. Hay elementos que se consideran opciones terminales (**Option**) que desencadenan la activación de alguna acción del programa. Existe un tipo de elemento con la finalidad estética de separar grupos de elementos relacionados (**Separator**). También se dispone de elementos menú (**Menu**) que representan a una secuencia de otros elementos. Los elementos que contiene un menú pueden ser de cualquier tipo: opciones, separadores y/o menús (submenús). No se establece límite en la profundidad de menús (menú dentro de menú).

Todos los elementos tienen un identificador único y disponen de un método para imprimirlo (**print**) que permite obtener una representación como String del elemento, convenientemente indentado (sangrado). Las opciones y los menús tienen también un texto a mostrar. A los menús se les puede añadir elementos y consultar si tienen un elemento especificado. También es posible obtener un array con los elementos de un menú. Los menús tienen también la característica de poderse plegar, lo que condiciona la manera de imprimirse.

Se identifican las clases **Element**, **Option**, **Menu** y **Separator**, a desarrollar en el paquete **menu**.

Clase Element

La clase abstracta **Element** suministra los siguientes métodos (abstractos o no):

Método	Descripción
Element(String)	Constructor que inicializa un elemento pasándole una String que representa su identificador.
String getId()	Devuelve el identificador del elemento.
String print(int)	Se le pasa por parámetro el nivel de indentación del elemento y devuelve una String que representa al elemento con un indentado de dos espacios y un guion por nivel. Por ejemplo: nivel 3 => " - - -".

Clase Option

Un objeto de la clase **Option** tiene un identificador y un texto descriptivo de la opción. La impresión de una opción devuelve una String formada por el texto de la opción precedida por su indentado y seguido de un salto de línea ("\n"). Además de los métodos de la clase base **Element**, dispone de los siguientes métodos:

Método	Descripción
Option(String, String)	Constructor que inicializa una opción pasándole por parámetro el identificador y el texto descriptivo de la opción.
String getText()	Devuelve el texto descriptivo de la opción.

Clase Separator

Un objeto de la clase **Separator** tiene solo el identificador. La impresión de un separador devuelve una String formada por una String literal compuesta de 9 caracteres guión ("-----") precedida por su indentado y seguido de un salto de línea ("\n"). Además de los métodos de la clase base **Element**, dispone de un constructor que inicializa el separador con el identificador pasado por parámetro.

Clase Menu

Un objeto de la clase **Menu** tiene un identificador y un texto descriptivo del menú. Cada objeto de la clase **Menu** contiene una secuencia de elementos (opciones, separadores y/o menús), pudiendo contener un máximo de 20 elementos. Los menús pueden estar en modo plegado (**folded**) y desplegado, que afecta a cómo se imprimen. La impresión de un menú devuelve una String formada por el nombre del menú precedida por su indentado y seguida de un salto de línea ("\n"). En caso de que el menú esté desplegado, la String con el resultado de la impresión va seguida de la concatenación de los resultados de imprimir **con un nivel más** cada uno de los elementos que contiene. Además de los métodos de la clase **Element**, la clase **Menu** dispone de los siguientes métodos:

Método	Descripción
Menu(String, String)	Constructor que inicializa un menú pasándole por parámetro el identificador y el texto descriptivo del menú. El menú está desplegado inicialmente.
String getText()	Devuelve el texto descriptivo del menú.
void setFolded(boolean)	Se le pasa por parámetro un valor booleano que establece si el menú está plegado con el valor true , o está desplegado, con el valor false .
boolean isFolded()	Devuelve el estado de la característica de plegado del menú.
boolean addElement(Element)	Añade el elemento pasado por parámetro al menú al final de la secuencia. Devuelve false si el elemento ya está contenido en la lista de elementos directamente, o si el número de elementos directos ya es de 20. En caso de añadirlo devuelve true . NO se permite el uso de contenedores de la librería de Java para almacenar los elementos.
boolean isContain(String)	Se le pasa un identificador de elemento y devuelve true si está contenido en el menú de elementos (directa o indirectamente) y false , si no está.
Element[] getElements()	Devuelve un array con los elementos que almacena el menú y en el mismo orden en que fueron añadidos.

Se dispone de una clase de apoyo ya implementada llamada **Tools** que suministra el método **static** String getIndent (level) que devuelve una String formada por el texto de indentación del nivel correspondiente.

Se suministra

Un programa principal en la clase MenuMain que crea varios objetos y utiliza las operaciones de las clases Element, Option, Separator y Menu. Para ejecutar las pruebas es necesario incluir en el método main() la sentencia: org.junit.runner.JUnitCore.main("menutest.MenuTest");

Ejercicio 1 (50%)

Desarrolle las clases anteriores según los requisitos descritos en el enunciado.

Ejercicio 2 (50%)

Desarrolle la clase llamada MenuTest que pruebe adecuadamente la funcionalidad de la clase Menu, empleando adecuadamente JUnit. Dicha clase debe pertenecer al paquete menutest.