



Trabajo Práctico State y Strategy

Objetivos

Comprender los patrones de diseño State y Strategy. Discutir alternativas de diseño e implementación.

Encriptación

Debe implementar la clase EncriptadorNaive, que se encarga de transformar cadenas de texto para que puedan ser enviadas por una red de computadoras en forma segura. La misma posee una única forma de encriptar que consiste en cambiar el orden de las palabras. Luego, debe agregar nuevas formas de encriptar y desencriptar, por ejemplo:

1. Modificando cada aparición de una vocal en el texto por la vocal siguiente en el orden tradicional, esto es: si aparece una a, la cambia por una e, si aparece una e la cambia por una i y así siguiendo. Para las consonantes, números y cualquier otro carácter no realiza ningún cambio.
2. Escribiendo cada letra por un numero que la representa. Las letras del abecedario están numeradas la 'a' con el 1, la 'b' con el 2 y así siguiendo. El espacio esta representado por el numero 0 y los números están separados por comas. Por ejemplo la palabra "Diego" se representa por la cadena "4,9,5,7,15". No se distinguen mayúsculas, ni minúsculas. Asuma que no hay acentos ni otros caracteres diferentes a las letras y los espacios.

Por su parte, el método desencriptar realiza la inversa de cada uno. Recibe un String como parámetro y retorna otro String donde realizo la desencriptacion.

- public String encriptar(String texto)
- public String desencriptar(String texto)

Sobre este enunciado:

1. Realice un diagrama de clases UML de la solución propuesta
2. Realice los test de unidad necesarios para la clase EncriptadorNaive y sus método encriptar y desencriptar, para que pueda ser utilizada para encriptar y desencriptar cadenas de texto utilizando las diferentes formas de encriptación.
3. Implemente en Java.

Video Juego

Una máquina de video juegos funciona ingresando fichas. Además de los controles para jugar, la maquina posee un único botón de inicio y una ranura para ingresar fichas. Cuando la maquina se enciende presionar el botón de inicio genera un cartel en la pantalla que indica que ingresen fichas, si se introduce una ficha y se presiona el botón de inicio se comienza el juego para un único jugador; si en cambio antes de presionar el inicio se ingresaron dos fichas, pueden jugar dos jugadores. Cuando termina el juego, vuelve al momento inicial.

1. Realizar un diagrama de clases UML.

2. Realizar test que permitan chequear que los cambios de estados funcionan correctamente ante diferentes combinaciones de operaciones.
3. Implemente en Java.



Reproductor MP3

Se desea desarrollar un sistema que permita controlar la reproducción de música en un reproductor MP3. El sistema permite realizar las siguientes acciones:

- Play: Reproduce la canción seleccionada, si es que el reproductor esta en modo de selección de canciones. Caso contrario produce un error.
- Pause: Pausa la canción que se está reproduciendo. Si la canción está en pausada retoma la reproducción, caso contrario genera un error.
- Stop: Para la reproducción o la pausa, y pasa a selección de canciones. Caso contrario no hace nada.

Se asume que existe una clase "Song" que entiende los mensajes `play()`; `pause()`; `stop()`; con la cual usted debe interactuar.

1. Realizar un diagrama de clases UML.
2. Realizar test que permitan chequear que los cambios de estados funcionan correctamente ante diferentes combinaciones de operaciones.
3. Implemente en Java.

