

Tema 1. Práctica 2: Implementar un Shell.

En esta práctica vamos a implementar un shell de linux. ¿Qué hace un shell?

- Permite escribir un comando.
- Muestra por pantalla la salida de ejecución del comando.
- Permite escribir un nuevo comando, y así sucesivamente.
- En caso de escribir el comando “exit”, finaliza la ejecución.

Por lo tanto vamos a realizar una aplicación que permite al usuario introducir comandos, mostrará por pantalla el resultado. Cuando el comando introducido sea “exit” saldrá de la aplicación. Cuando el comando sea “last-command” mostrará información del último comando ejecutado.

1. Creamos una clase Command.
 - a. Tendrá dos constructores:
 - i. Uno de ellos recibe dos parámetros: un array de argumentos del tipo cadena y una cadena con el nombre de un fichero donde se redirige la salida estándar (cadena vacía si no se redirige la salida estándar, o lo que es lo mismo: si se muestra la salida del comando por consola).
Por ejemplo:
[“ls”, “-la”]
“/home/usuario/fichero.txt”
 - ii. El otro constructor recibe un único parámetro: una cadena con todos los argumentos concatenados y un “>” si se redirige la salida.
““ls -la > /home/usuario/fichero.txt””
““ls -la””
 - b. Esta clase sobreescribe el método `toString()` para devolver información del proceso: el comando a ejecutar, el número de parámetros, los parámetros concretos. Si el comando ha sido ejecutado mostrará el pid (identificador único) y el texto de la salida (si no ha sido redireccionado). Si el comando ha finalizado también se indicará junto con el código de salida (`exitValue`).
 - c. Añadimos una función ejecutar que ejecuta el comando. La función devuelve una cadena con el texto de la salida estándar del proceso. Si la salida estándar ha sido direcccionada devolverá una cadena vacía.
 - d. Añade gestión de excepciones a la hora de crear el proceso y a la hora de ejecutarlo.
2. Creamos el programa principal (cuya clase se llamará `Shell`). Escribirá por pantalla el carácter “>”. A continuación leerá un comando por consola y mostrará el resultado de su ejecución. Si está redireccionada la salida estándar no mostrará nada.
3. Habrá dos comandos especiales:
 - a. “last-command” mostrará toda la información del último comando.
 - b. “exit” saldrá del shell.
4. Comenta el código con Javadoc y exporta la documentación en una carpeta doc del proyecto.