EJERCICIOS UNIDAD 1

1. RA1 -CE e) Creación de procesos en Java

1.1 Paquete

El paquete de Java que contiene las librerías necesarias es: java.lang

1.2 Clases

Las clases que vamos a utilizar en estos ejercicios son:

- Process
- ProcessHandle
- ProcessBuilder
- Runtime

Runtime

- Encapsula un entorno de ejecución.
- No puedes instanciar un objeto de Runtime.
- Documentación de la clase: https://docs.oracle.com/en/java/javase/18/docs/api/java.base/java/lang/Runtime.html

Process

- Clase abstracta.
- Encapsula un proceso.
- Usada como superclase de los objetos creados con el método *exec()* en la clase *Runtime* o *start()* en la clase *ProcessBuilder*.
- Documentación de la clase: https://docs.oracle.com/en/java/javase/18/docs/api/java.base/java/lang/Runtime.html

ProcessHandle

- Interfaz que provee un control sobre los procesos nativos.
- Estas instancias se pueden obtener mediante el método *toHandle()* de la clase *Process*.
- Documentación de la clase: https://docs.oracle.com/en/java/javase/18/docs/api/java.base/java/lang/ProcessHandle.html

ProcessBuilder

- Proporciona otra manera de manejar procesos.
- Provee un mayor control sobre los procesos que *Runtime.*
- Permite manejar procesos con un mayor control
    - Variables de entorno.
    - Directorio de trabajo.
    - Entrada estándar y salida de error estándar.
    - Interacción con ficheros.
- Documentación de la clase: https://docs.oracle.com/en/java/javase/18/docs/api/java.base/java/lang/ProcessBuilder.html

Ejercicios con clase Runtime

1. Utilizando la clase **Runtime**, realiza un programa que lance un proceso hijo que ejecute el *notepad* de Windows pasándole como argumento un fichero de texto para que lo abra (crea un fichero de ejemplo en la raíz del proyecto para probarlo). Los argumentos se pasan a través de la línea de comandos (String[] args).

2. En este ejercicio el estudiante ha de crear un programa en java que:

- ***Dependiendo del sistema operativo:***
    - Abre la calculadora de Windows.
    - Abre el navegador Firefox en Ubuntu Linux.
- **Usa *ProcessHandle* para obtener:**

1. El nombre del comando
2. Los argumentos usados
3. Tiempo de comienzo.
4. Tiempo de CPU.
5. Propietario.
6. Número de hijos.

*Tutorial de ProcessHandle: https://www.demo2s.com/java/java- get-process-information-from-processhandle-and-processhandle-info.html.

* Tutorial para detector el Sistema operativo en Java: https://www.w3schools.blog/detect-operating-system-in-java

* Tutorial para abrir una aplicación desde el terminal en Windows: https://stackoverflow.com/questions/40493141/how-to-utilize-powershell-to-open-an-application-with-a-command

* Tutorial para abrir una aplicación desde el terminal de Linux: https://www.wikihow.com/Run-a-Program-from-the-Command-Line-on-Linux

*Tutorial de cómo abrir una aplicación desde el terminal de MacOS:
https://www.wikihow.com/Open-Applications-Using-Terminal-on-Mac

3. Modifica el programa del ejercicio 1 que lanzaba el Notepad de Windows. Ahora, el padre, mientras el hijo se está ejecutando, debe realizar una tarea sencilla que consiste en generar 10 números aleatorios entre el 0 y el 9. ¿qué ocurre y por qué?

Ejercicios con clase ProcessBuilder

4. Utilizando la clase *ProcessBuilder*, se debe crear un programa en Java que:

1. Pregunte una app con interfaz gráfica en el Sistema Operativo actual.
2. Abra la app.
3. Obtener la información del entorno del proceso.

5. Utilizando la clase *ProcessBuilder* ejecuta el comando ***"CMD /c ver"*** y redirige la salida estándar a un fichero llamado "salida.txt" y la salida de error a un fichero llamado "error.txt". Para ello, utiliza los métodos *redirectOutput* y *redirectError* de *ProcessBuilder.* Ejecuta después un comando que no exista como por ejemplo ***"CMD /c verr"***.