PRÁCTICA 3: EJERCICIOS DE PROGRAMACIÓN MULTIPROCESO

Escribe en Java los programas necesarios:

- Utiliza de manera adecuada la Programación Orientada a Objetos.
- Presta atención a las excepciones que pueda originar el código e informa al usuario.
- Imprime en pantalla los mensajes necesarios cuando haya que informar al usuario.
- No olvides comentar el código y documentar los métodos.

1. EJERCICIO 4

Cree un par de programas similares a los que hizo en el Ejercicio 2 y el Ejercicio 3 para hacer la traducción de algunas palabras del inglés al español:

- Las palabras a traducir pueden llegar al padre desde la línea de comando (pidiendo palabras al usuario) o desde un archivo de texto como argumento para el programa (donde cada palabra va en una línea diferente). El padre envía cada palabra, una por una, al niño (Traductor) y obtiene la traducción hasta que no hay más para traducir.
    - El proceso padre escribe cada palabra traducida recibida en un archivo: `Translations.txt`, en el formato: `<original_word> – >> <translation>`*
- El proceso hijo (Traductor) traduce cada palabra recibida en su entrada estándar y la escribe en su salida estándar. Si no puede traducir una palabra, escribe la palabra "desconocido". El traductor puede recibir como argumento el nombre de un archivo de diccionario para usar, si no, usa una lista fija interna de palabras como diccionario. (Use un HashMap para crear el diccionario desde un archivo o desde una lista fija)

> *Nota:* Cuando traduciendo desde la línea de comando, decida un método para detener ambos procesos, padre / hijo, cuando el usuario no quiera traducir más.

1.1. Ejemplos de uso [Ex4.jar(padre) Translator.jar (hijo)]

Caso	Comando	Comentarios
Sin argumentos para el padre	`java –jar Ex4.jar`	*El padre le pide palabras al usuario y el hijo usa un diccionario interno*
El padre recibe el diccionario como argumento	`java –jar Ex4.jar –d dict.txt`	*El padre le pide palabras al usuario y comienza el hijo con dict.txt como argumento*
El padre recibe un archivo para traducir como argumento	`java –jar Ex4.jar –f archivo.txt`	*El padre obtiene palabras de file.txt y se las envía al niño que usa el diccionario interno*
El padre recibe como argumentos: un archivo para traducir y un diccionario puede venir en cualquier orden	`java –jar Ex4.jar –d dict.txt –f archivo.txt`	*El padre obtiene palabras de file.txt y comienza el hijo con dict.txt como argumento*

2. EJERCICIO 5

Hagamos uso de la multiprogramación en el sistema operativo para realizar una tarea intensiva: sumar números.

1. Escribe un programa llamado Adder que toma como argumento el nombre de un archivo de texto que contiene números enteros, cada número viene en una línea diferente. El programa imprime en la salida estándar la suma total de todos los números encontrados en el archivo y termina. En caso de que se encuentre algún error, imprime el mensaje de error al error estándar y finaliza. (Genere también el archivo Adder.jar)

2. Escriba un programa llamado Ex5_Accounter que calcule la suma total de los números encontrados en una lista de archivos. El programa realiza las siguientes tareas:

- Toma una lista de nombres de archivos como argumento.
- Inicia un proceso hijo (Adder.jar) para cada nombre de archivo.
- Cuando un hijo termina, obtiene un resultado de él y acumula una cantidad hasta la suma total.
- Guarda cada resultado secundario y la suma total en un archivo de resultados: Totals.txt
- Imprime la suma total e informa al usuario sobre el archivo de resultados generado.
- Además, imprime cualquier mensaje de error proveniente de procesos secundarios.