PSP - U2 Actividades

Descargar estos apuntes

Índice

1. Clase psp.actividades.U2A8_LeerNombre

Realiza un programa en Java que reciba como argumentos, a través del main(), un nombre y lo visualice en pantalla.

> **Prueba de la clase de forma independiente**
>
> Puedes probar el funcionamiento de la clase de forma independiente, pulsado Shift + F6.
> Sin embargo, si quieres que la clase reciba argumentos a través del main, para probarla tendrás que ponerla como "Main Class" en la sección "Run" de las propiedades del proyecto.

Clase psp.actividades.U2A8_Lanzador

A continuación haz un programa que lance LeerNombre. Utiliza waitFor() para comprobar el valor de salida del proceso que se ejecuta.

> **Clase principal vs ejecutar clase**
>
> Si quieres que al pulsar el botón `Play` o `F6` en Netbeans se ejecute la clase LeerNombre, tendrás que ponerla como "Main Class" en la sección "Run" de las propiedades del proyecto.
>
> Si por el contrario, esta clase no necesita recibir parámetros en su ejecución, también puedes ejecutarla pulsando `Shift + F6` teniéndola seleccionada en el explorador de proyectos.

> **Valor de retorno**
>
> Prueba la ejecución del proceso LeerNombre dando valor a los argumentos del main() y sin darle valor.
> ¿Qué devuelve waitFor() en un caso y en otro?
>
> Si pruebas la clase de forma independiente, qué se muestra por consola si no le pasas argumentos y qué se muestra si le pasas argumentos.

2. Clase psp.actividades.U2A9_LanzadorComandos2

Realiza un programa en Java que reciba como parámetro un comando para ejecutar en una consola (shell).

Haz que tu programa ejecute el comando como un nuevo proceso y que muestre por pantalla la salida del comando (precedida por “> “) y los errores del comando (precedidos por “ERROR> “).

Ejecución de comandos shell

Recuerda que los comandos shell, para ejecutarse como un proceso y que devuelvan el control al padre cuando acaben de ejecutarse, deben ser lanzados como

```
cmd /c comando
```

Ejecuta comandos que existan y comandos que no existan, con parámetros correctos y con parámetros incorrectos.

Recogida de la salida del proceso hijo

La clase U2A9_LanzadorComandos2 debe recoger la salida del proceso hijo y mostrarla por pantalla. Para ello, usa los método `getInputStream()` y `getErrorStream` de la clase `Process` que te devolverán un `InputStream` que puedes leer con un `BufferedReader`.

Es importante mostrar la salida y los errores de forma diferenciada. En la actividad indica que la salida que se recoja del *InputStream* debe ir precedida por `>` y los errores que se recojan de `ErroStream` por `ERROR>`.

Además, si miras el **Anexo I** de la unidad, podrás ver como usar colores en la consola, esto te va a permitir diferenciar la salida y los errores de forma visual. También, usar `System.out.println()` para mostrar la salida y `System.err.println()` para mostrar los errores permite diferenciarlos en la consola.

3. Clase psp.actividades.U2A10_Sumador

Realiza un programa en Java que lea dos números desde la entrada estándar `(stdin)` y muestre por pantalla el resultado de la suma.

Controla que lo introducido por teclado sean dos números.

Clase psp.actividades.U2A10_Lanzador

A continuación haz un programa que lance Sumador, le proporcione las entradas necesarias y muestre el resultado por pantalla

```
La suma de 5 + 8 = 13
```

Opcional: Haz que se ejecute varias veces el programa Sumador. Para eso desde Lanzador deberás solicitar los números por la entrada estándar.

> **Pista**
>
> ¿Cómo puede un proceso pasarle información al proceso que lanza? ¿Cómo puede un proceso recibir información de otro proceso?

4. Clase psp.actividades.U2A11_Sumador2

Modifica el programa anterior para que la Clase Sumador2 lea los números, de 2 en 2, desde la entrada estándar y muestre por consola el resultado de cada una de las sumas.

```
La suma de 5 + 8 = 13
La suma de 9 + 34 = 43
...
```

Si en algún caso la cantidad de números que lee no es par, el que falte lo sustituirá por 0.

Hay que controlar los posibles errores que se produzcan en la lectura de los números y generar mensajes de error para avisar al usuario.

Clase psp.actividades.U2A11_Lanzador

Crea un lanzador que modifique el comportamiento de Sumador2 para que lea los números desde un fichero de texto (datos/numeros.txt).

La clase lanzadora será la encargada de mostrar por consola la información del proceso Sumador 2 (siguiendo el formato de la `actividad 2`: “>” y “ERROR>”).

> ⚠ **Lectura del fichero de texto**
>
> Fíjate que se pide que el lanzador modifique el comportamiento de Sumador2 para que lea del fichero.
>
> En ningún caso hay que modificar Sumador 2 para que acceda al fichero datos/numeros.txt.
>
> Tampoco hay que hacer que Lanzador lea el fichero y pase los datos a Sumador2. Sumador2 debe leer el fichero por sí mismo.
>
> El fichero de texto debe contener los números a sumar, con un número por línea. Por ejemplo, para el siguiente fichero de texto:
>
> ```
> 6
> 8
> 4
> 1
> 5
> ```
>
> La salida sería la siguiente:
>
> ```
> > La suma de 6 + 8 = 14
> > La suma de 4 + 1 = 5
> > La suma de 5 + 0 = 5
> ```

5. Clase psp.actividades.U2A12_Argumentos

Realiza un programa en Java que reciba argumentos a través del main y devuelva con `System.exit()` los siguientes valores:

- Si el número de argumentos es < 1 debe devolver -1
- Si el argumento es una cadena debe devolver 2
- Si el argumento es un número entero menor que 0 debe devolver 3.
- Si no se cumple ninguna de las situaciones anteriores, debe devolver 0

Clase psp.actividades.U2A12_Lanzador

Realiza un programa que ejecute el anterior como un proceso. Este segundo programa deberá mostrar en pantalla un mensaje indicando lo que pasa en función del valor devuelto.

Es decir, tendrá que mostrar un mensaje diferente indicando qué tipo de argumento ha recibido Argumentos, en función del valor devuelto por el proceso lanzado.

6. Clase psp.actividades.U2A13_Shell

Haz un programa que emule a la shell de un sistema operativo.

El programa pedirá, por la entrada estándar, la introducción de un comando.
El comando se parseará y dividirá en trozos y se pasará a `Interprete`.

Se esperará a que el proceso `Interprete` termine y se recuperará la información de salida y de error, mostrándola por pantalla (igual que en la `actividad 2)`.

Se repetirá el proceso hasta que se introduzca el comando "exit", momento en el que el proceso padre mostrará un mensaje indicando que la ejecución ha acabado y finalizará su ejecución.

Clase psp.actividades.U2A13_Interprete

Crea un programa que reciba la información del proceso Shell, y ejecute el comando de shell recibido. Se creará un proceso hijo al que se le pasará toda la información del comando para que la ejecute.

> **Abuelo, Padre, Hijo**
>
> En esta actividad, tenemos un proceso abuelo (Shell) que lanza un proceso padre (Interprete) que lanza un proceso hijo (la consola con el comando que se quiere ejecutar).
>
> Por lo tanto, para que la información llegue al abuelo, el padre debe recoger la información de salida y error del hijo antes de pasársela al abuelo. Y el abuelo debe esperar a que el padre termine para poder recoger su información de salida y error.

7. Clase psp.actividades.U2A14_Controlador

Haz un programa que calcule

$$\sum_{i=1}^{n} \sqrt{i}$$

El programa controlador recibe como parámetro el valor de n y, de forma opcional, también se le puede pasar la cantidad de operaciones a realizar por el proceso hijo (por defecto, si no se indica nada, 100).

Va a dividir equitativamente el reparto entre (n/100)+1 procesos hijos.

- Al primer proceso le asignará el cálculo desde 0 hasta 99,
- al 2º desde 100 hasta 199,
- al 3º desde 200 hasta 299,
- ...
- al último le asignará los que queden hasta n.

El programa recogerá el resultado que muestre cada proceso hijo, los irá sumando y cuando todos hayan acabado, mostrará el resultado final.

Todos los procesos se lanzarán antes de bloquear al Controlador.

> La suma total de las raíces cuadradas desde 0 hasta 10875 es: 756105.62767

Clase psp.actividades.U2A14_Calculador

Esta clase recibirá como parámetro el número de proceso hijo que es, el valor inferior del cálculo que debe realizar y el valor superior.

La clase, utilizando la salida de error, añadirá a un archivo (ResultadosParciales.txt) el resultado del sumatorio parcial que realiza:

> Hijo 1: La suma de las raíces cuadradas desde 0 hasta 99 es: 661.46294
> Hijo 2: La suma de las raíces cuadradas desde 100 hasta 199 es: 1202.77239
> …
> Hijo 109: La suma de las raíces cuadradas desde 10800 hasta 10875 es: ….

A través de la salida estándar enviará el resultado numérico al proceso padre.

Tip

Para que Calculador guarde la información en un archivo, el controlador deberá configurar el proceso hijo de forma adecuada.

Nunca Calculador debe escribir directamente en el archivo, siempre debe hacerlo escribiendo en la salida de error.

8. Clase psp.actividades.U2A15_Conversor

Investiga cómo se puede convertir un documento (.odt o .docx) a PDF usando LibreOffice desde línea de comandos.

La clase Conversor recibe el nombre (puede incluir una ruta) de un documento como parámetro y crea un proceso hijo `psp.actividades.U2A15_Doc2Pdf` que se encarga de realizar la conversión a PDF de ese documento.

La clase padre mostrará el resultado de la conversión (satisfactorio o fallido).

Los archivos generados se guardarán en una carpeta `PDF` en la raíz del proyecto.