

FAMILIA PROFESIONAL:

CICLOS FORMATIVOS:

MÓDULO:

Informática y Comunicaciones

Desarrollo de Aplicaciones Multiplataforma,

Programación de Servicios y Procesos

UNIDAD 1: PROGRAMACIÓN MULTIPROCESO

ACTIVIDADES

ACTIVIDAD 1x01

Escribe una clase **EjecutaComando** con un programa principal que reciba como argumentos un comando y sus opciones. Este programa principal deberá crear un proceso hijo con `Runtime` y ejecutar dicho comando con las opciones especificadas. El proceso padre deberá esperar a que el proceso hijo termine e informe de si se ha producido alguna anomalía en la ejecución del hijo. En caso de que no se pueda ejecutar correctamente el comando, escribirá en consola un mensaje de error.

Por ejemplo, si este programa intenta ejecutar el comando `cmd /c calc`, lo ejecutará correctamente abriendo la calculadora de Windows.

Si este programa intenta ejecutar el comando `cmd /c listar`, mostrará en consola el mensaje de error:

```
Error of running [cmd, /c, listar] is:
"listar" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.
```

ACTIVIDAD 1x02

Escribe una clase **Aleatorios** con un programa principal que cree un proceso hijo a partir de la clase **GeneraAleatorio**. Esta clase contendrá un programa principal que escribirá en su salida estándar un número aleatorio entre 0 y 10 cada vez que reciba una petición de ejecución desde el padre.

El proceso padre leerá líneas de texto de la entrada estándar y, por cada línea que lea, solicitará al hijo que le envíe un número aleatorio, lo leerá y lo imprimirá en pantalla. Cuando el proceso padre reciba la palabra `fin`, finalizará la ejecución del hijo y procederá a finalizar su ejecución.

Un ejemplo de ejecución de este programa podría ser:

```
Escribe una línea: uno
3
Escribe una línea: dos
1
Escribe una línea: tres
5
Escribe una línea: cuatro
3
Escribe una línea: cinco
6
Escribe una línea: fin
Finalizando
```

ACTIVIDAD 1x03

Escribe una clase **Mayusculas** con un programa principal que cree un proceso hijo a partir de la clase **PasaMayusculas**. Esta clase contendrá un programa principal que leerá una frase por su entrada estándar, la transformará todo a letras mayúsculas y escribirá esta frase convertida a mayúsculas en su salida estándar.

El proceso padre y el proceso hijo se comunicarán de forma bidireccional utilizando flujos (*streams*). El padre leerá frases de su entrada estándar y las enviará a la entrada estándar del hijo (utilizando el `OutputStream` del hijo). El padre imprimirá en pantalla lo que recibe del hijo a través del `InputStream` del mismo. Cuando el proceso padre reciba la palabra `fin`, finalizará la ejecución del hijo y procederá a finalizar su ejecución.

Un ejemplo de ejecución de este programa podría ser:

```
Escribe una frase: Hoy he ido a Teruel en autobús
HOY HE IDO A TERUEL EN AUTOBÚS
Escribe una frase: Y mañana iré a Huesca en tren
Y MAÑANA IRÉ A HUESCA EN TREN
Escribe una frase: fin
Finalizando
```

ACTIVIDAD 1x04

Escribe una clase **Primos** con un programa principal que cree un proceso hijo a partir de la clase **CalculaPrimos**. Esta clase contendrá un programa principal que leerá un número por su entrada estándar. Si el número es primo, escribirá por su salida estándar un mensaje El número N es primo, siendo N el número leído. Si el número no es primo, escribirá por su salida estándar todos los números primos menores que N , de mayor a menor, hasta llegar al número 1.

El proceso padre leerá un número entero por su entrada estándar (usando un `BufferedReader`). Si el dato leído no tiene formato de número entero, el padre escribirá en su salida estándar un mensaje de error específico y finalizará su ejecución. Si el dato leído es un número entero, el padre se lo pasará al hijo mediante un `PrintStream`. Cuando le llegue la respuesta del hijo, la mostrará por su salida estándar.

Un primer ejemplo de ejecución de este programa podría ser:

```
Escribe un número entero:
siete
Dato introducido no válido
```

Un segundo ejemplo de ejecución de este programa podría ser:

```
Escribe un número entero:
11
El numero 11 es primo.
```

Un tercer ejemplo de ejecución de este programa podría ser:

```
Escribe un número entero:
18
Los números primos menores que 18 son:
17
13
11
7
5
3
2
1
```

ACTIVIDAD 1x05

Escribe una clase **ProcesaValorRetorno** con un programa principal que cree un proceso hijo con `Runtime` a partir de la clase **ValorRetorno**.

El proceso padre esperará a que termine el hijo y recogerá su código de finalización. Si el código es menor que cero, escribirá por su salida estándar de errores el mensaje `Error en proceso de numeración` y lanzará una excepción de entrada/salida. Si el código es cero, escribirá por su salida estándar el mensaje `Proceso finalizado correctamente`. Si el código es mayor que cero, escribirá por su salida estándar un mensaje con el valor de retorno.

El proceso hijo será una clase **ValorRetorno** con un programa principal que genere de forma aleatoria un número entero entre -1 y 1 y finalice su ejecución con este valor como código de terminación.

Un primer ejemplo de ejecución de este programa podría ser:

```
Error en proceso de numeración
Excepción de Entrada/Salida
```

Un segundo ejemplo de ejecución de este programa podría ser:

```
Proceso finalizado correctamente
```

Un tercer ejemplo de ejecución de este programa podría ser:

```
Proceso finalizado con valor de retorno 1
```