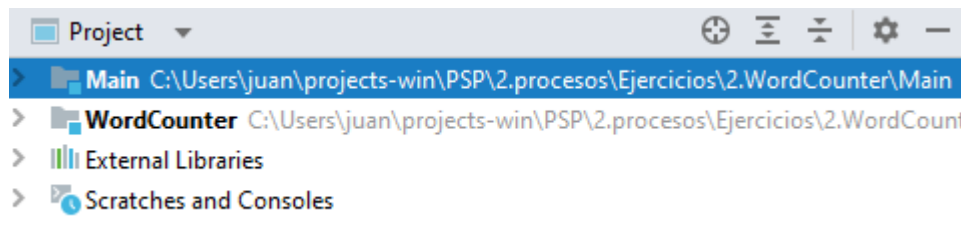


Práctica 2 - Contar palabras

El objetivo de esta práctica es aplicar los conocimientos adquiridos sobre las clases `ProcessBuilder`, `Process`, y `Streams` y sus derivados, escribiendo un programa capaz de contar palabras en fichero

En la práctica deberás entregar un proyecto de IntelliJ con **dos módulos**, uno padre, llamado `Main` y otro hijo, llamado `WordCounter`



Módulo hijo

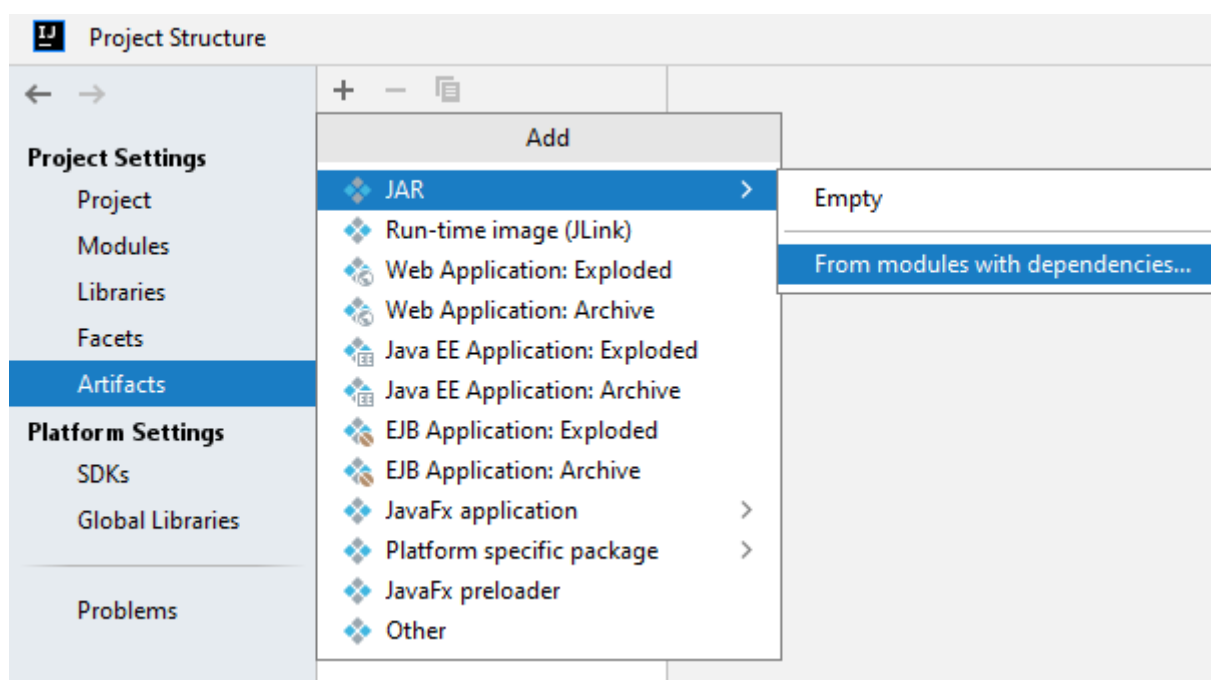
El módulo hijo implementará la funcionalidad de contar palabras. La implementación recibirá la ruta al fichero **a través de la entrada estándar**, por lo que el proceso debe implementar un **bucle infinito** en el que:

1. lee una ruta
2. abre el fichero
3. cuenta sus palabras
4. escribe por la salida estándar la cuenta.

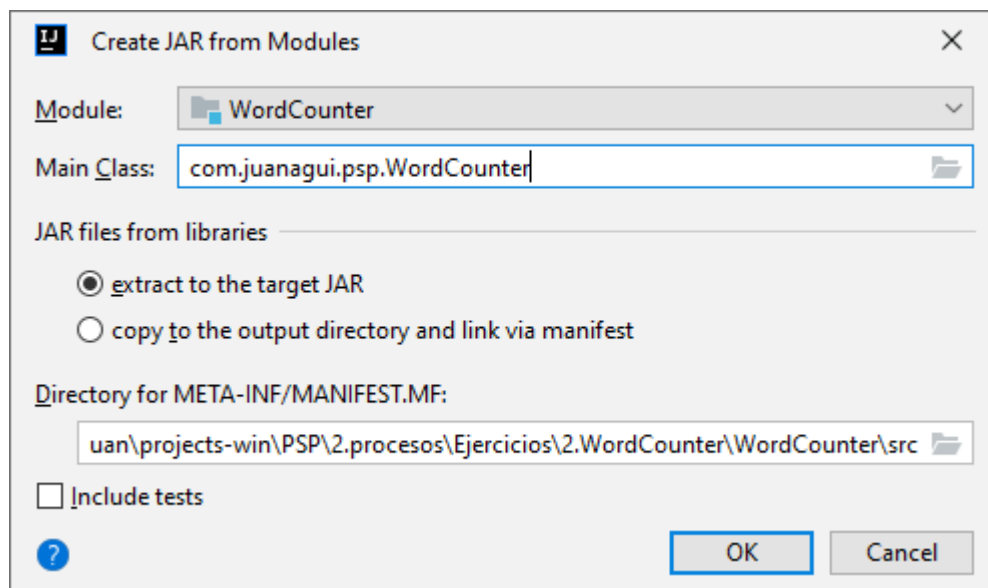
Deberás **generar un artefacto** para este módulo.

Generar un JAR

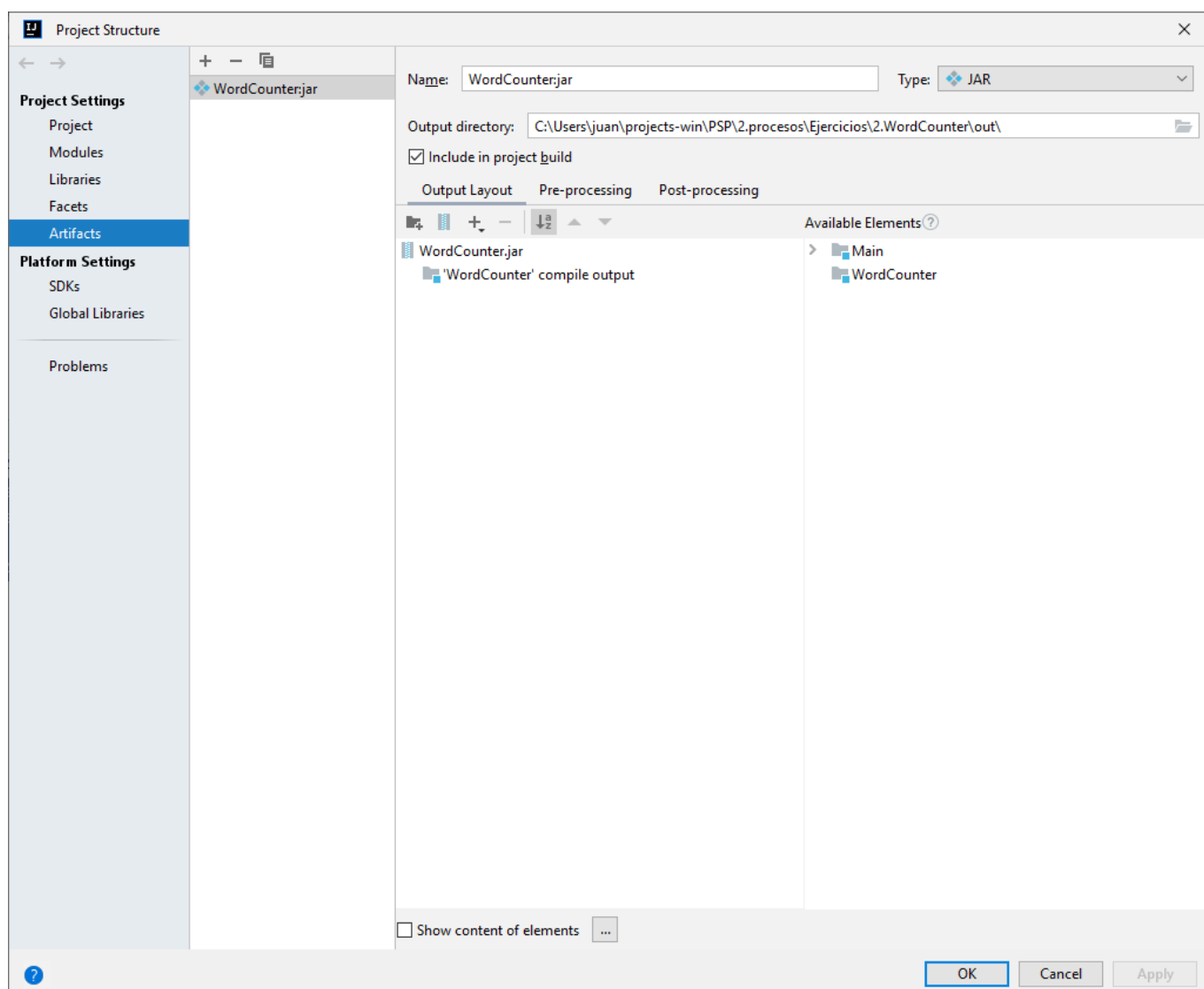
Para generar un JAR a partir de un módulo, vamos al diálogo **estructura de proyecto** y en artefactos, añadimos un JAR a partir de un módulo con dependencias, como se observa en la figura:



Seguidamente seleccionamos el módulo y la clase que contiene el `main`



El resultado final, en el que marcamos la casilla para que se genere el JAR al construir el proyecto, se observa en la siguiente figura:



Módulo padre

El módulo padre utilizará la clase `Files.newDirectoryStream` para leer los ficheros contenidos dentro del directorio `files` en el que se encuentran los ficheros de texto cuyas palabras se desean contar.

Debes definir un **número de procesos que se van a crear de antemano**. Esos procesos irán recibiendo rutas de ficheros para contar las palabras **sin bloquear la ejecución**. En otras palabras, supongamos que el número de procesos es 3; el módulo padre creará entonces 3 procesos hijos y se conectará a sus entradas y salidas. Recuerda que para lanzar un `jar`, la línea de comandos es:

```
java -jar ruta-absoluta-al-fichero.jar
```

Como tengo 3 procesos disponibles, ahora debo recorrer el directorio y asegurarme de enviar un fichero a cada proceso. Después espero a que los procesos me contesten por el stream de entrada con el número de palabras que han contado y presento el dato y el nombre del fichero por mi salida estándar. Ahora bien, en el ejemplo os he puesto 4 ficheros, por tanto, ahora tendremos que enviar el último fichero a uno de los procesos y esperar solo a que ese proceso me devuelva la cuenta.

⚠ asegúrate de que los procesos de ejecutan concurrentemente. Bloquear el proceso padre hasta que el hijo termine de procesar un fichero es un grave error de concepto

La salida esperada para los ficheros de ejemplo proporcionados es

```
File The Dunwhich horror by H.P Lovecraft has 21230 words
File The empty house and other horror stories by Algernon Blackwood has 71561
words
File The Works of Edgar Allan Poe has 146599 words
File This Crowded Earth by Robert Bloch has 42457 words
```

Deberás finalizar el programa de forma elegante, asegurándote de

- Cerrar todos los streams de entrada y salida establecidos entre el padre y los hijos
- Destruir los procesos hijos
- Esperar a que finalicen

Ayuda: Cómo depurar procesos hijos

Nota: Este documentos está añadido en el [aula virtual](#)

Caso de uso

Cuando utilizamos la clase `ProcessBuilder` para crear subprocessos escritos en Java, el depurador de IntelliJ no puede conectarse a ellos de forma automática porque solo está conectado al proceso principal.

Una solución

1. Instalar el plugin [AttachMe](#)
2. **Seguir sus instrucciones.** Para Linux no tienes que hacer nada más.

3. Para Windows, tienes que exportar una variable de entorno que le diga a Java que se los subprocesos se conecten automáticamente a IntelliJ. Usa como patrón esta variable:

```
set JAVA_TOOL_OPTIONS=-javaagent:"c:\users\juan\.attachme\attachme-agent-1.2.1.jar"=port:7857 -agentlib:jdwp=transport=dt_socket,server=y,suspend=y,address=*:0
```

y sustituye los valores para que el path al jar de **attachme-agent** coincida con el que tienes instalado en tu equipo, tanto la ruta como el número de versión. El número de puerto debe coincidir con el que has configurado con el plugin.

Una vez exportado, lanza tu proceso principal desde la línea de comandos con la orden **java** y los subprocesos automáticamente se conectarán al IDE.