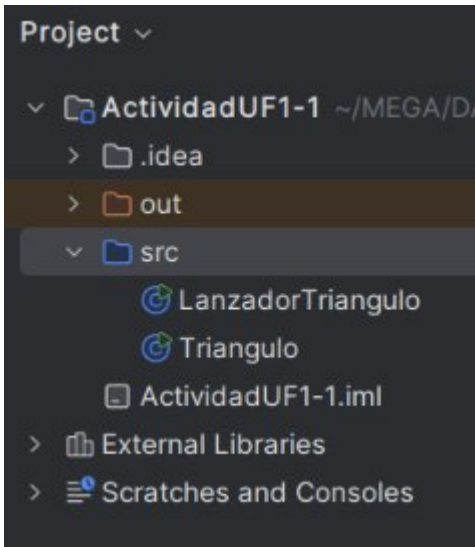


TRIÁNGULO NUMÉRICO

El programa consta de dos clases:



-Triangulo: Es la clase que ya venía en la tarea. Se le ha añadido la fecha de inicio del proceso y la de finalización. Se llama al método “.now()” de “LocalDateTime” dentro del “println” para que coja justo la hora a la que se imprime en pantalla y sea lo más preciso posible.

```
//AÑADE LA HORA DE INICIO DE PROCESO, FUERA DEL FOR PARA QUE NO SE REPITA EN CADA ITERACIÓN
System.out.println("Hora de inicio del proceso: " + LocalDateTime.now() + "\n");

int filas = Integer.parseInt(args[0]);
for (int i = filas; i ≥ 1; i--) {
    for (int n = 1; n ≤ i; n++) {
        System.out.print(n);
    }
    System.out.println();
}

//AÑADE LA HORA DE FINALIZACION DEL PROCESO, FUERA DEL FOR POR LO MISMO QUE LA DE INICIO
System.out.println("\n" + "Hora de finalización del proceso: " + LocalDateTime.now());
```

-LanzadorTriangulo: Desde esta clase se crea un proceso en base a la clase “Triangulo”

```
ProcessBuilder pb = new ProcessBuilder(...command: "java", "-cp", classpath, "Triangulo", String.valueOf(argumentos[i]));
```

Y se le añade toda la lógica para que realice lo que se pide en la tarea aunque también se han implementado varias mejoras que enumero en la siguiente página:

-Se ha creado un ArrayList de los procesos para manejarlos dentro de un for.

```
List<Process> procesos = new ArrayList<>();  
  
procesos.add(pb.start());
```

-Se le pasan los argumentos desde un Array de int dentro de un for. Esto hace no necesitar crear 3 procesos manualmente y se maneja todo de una sola vez.

```
int[] argumentos = {5, 7, 9};  
ProcessBuilder pb = new ProcessBuilder(...command: "java", "-cp", classpath, "Triangulo", String.valueOf(argumentos[i]));
```

-Se redirecciona la salida a un archivo ".txt" como se pide en el enunciado, pero se hace desde un Array de String. Antes del bloque for, si no existe la carpeta donde van los archivos, la crea.

```
String[] trianguloTxt = {"src/Triangulos/triangulo5.txt", "src/Triangulos/triangulo7.txt", "src/Triangulos/triangulo9.txt"};  
  
pb.redirectOutput(new File(trianguloTxt[i]));
```

-Se crea un segundo bloque for dentro del primero, en el cual añadimos la espera ".waitFor()" a cada proceso para que terminen todos a la vez y cuando lo hagan, salgan a su vez del primer for.

```
for (Process p : procesos) {  
    //SE LE AÑADE WAITFOR A CADA UNO DE LOS PROCESOS DENTRO DEL FOR PARA QUE TERMINEN TODOS JUNTOS  
    p.waitFor();  
}
```