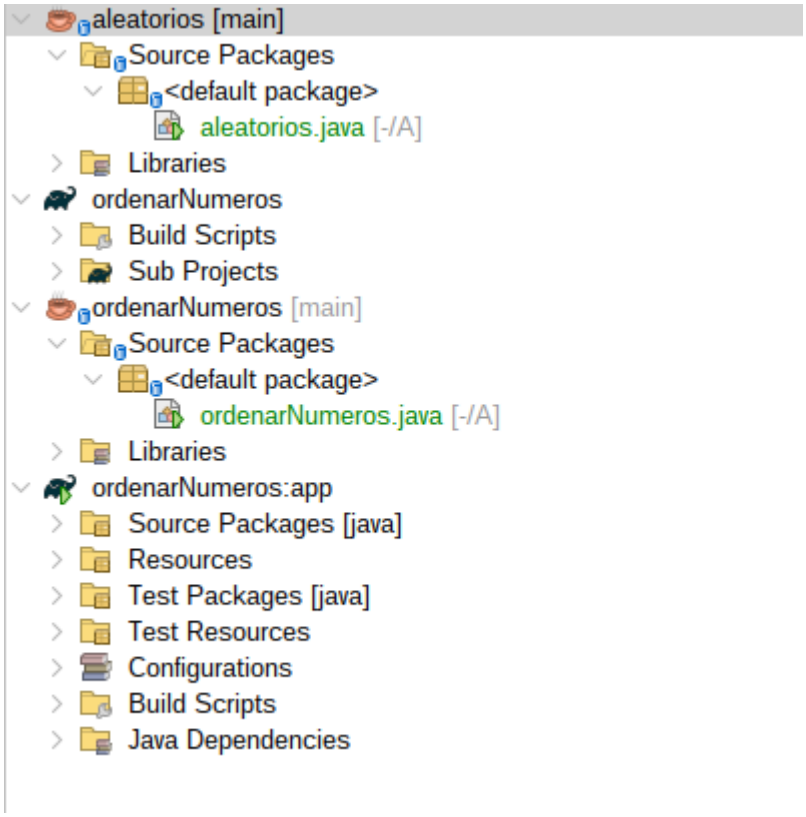
¿Cómo se hace?

La estructura de los proyectos quedaría así:



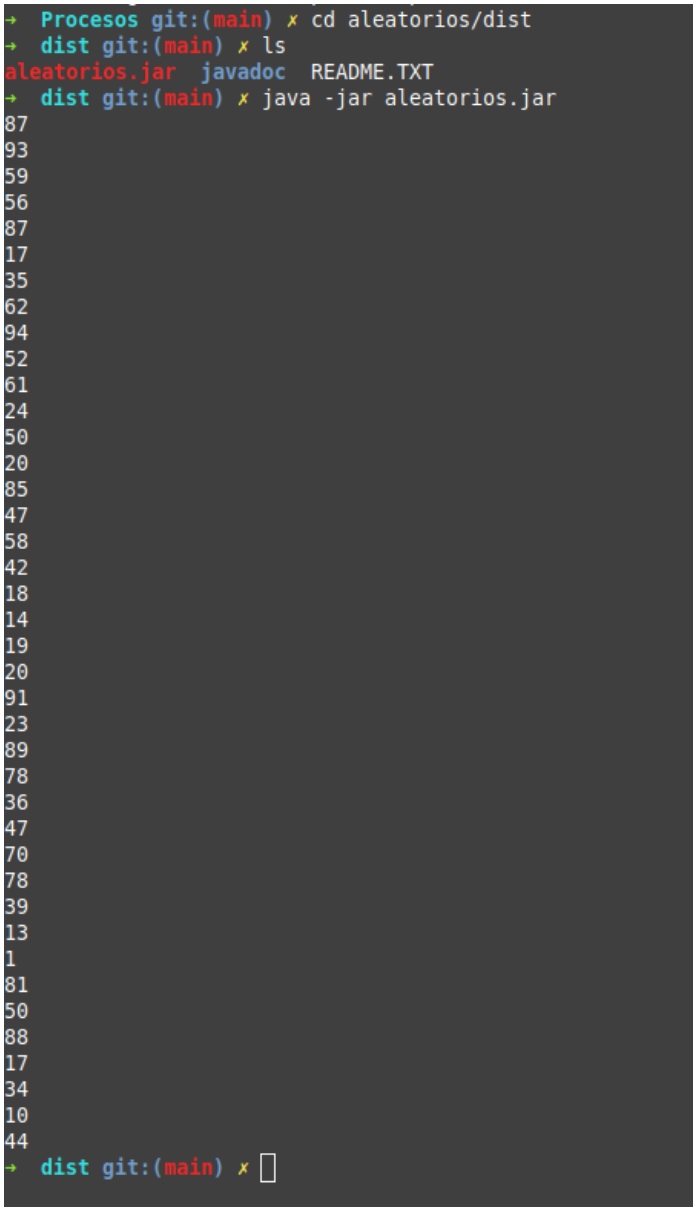
El código de aleatorios.java sería este:

ordenarNumeros.java [-/A] × aleatorios.java [-/A] ×

Source History

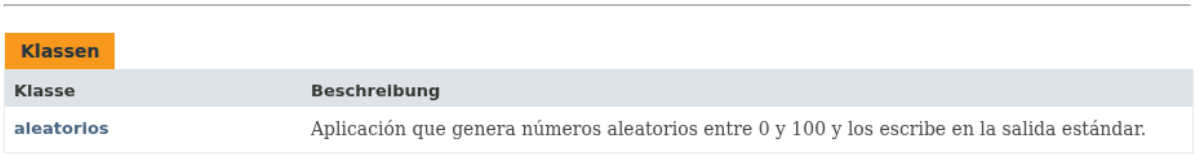
```java
import java.util.Random;

/**
 * Aplicación que genera números aleatorios entre 0 y 100
 * y los escribe en la salida estándar.
 *
 * @author Rodrigo
 * @version 1.0
 */
public class aleatorios {

    //Número de valores aleatorios a generar(40).
    private static final int CANTIDAD_NUMEROS = 40;

    /**
     * Valor mínimo del rango de números aleatorios (inclusivo).
     */
    private static final int VALOR_MINIMO = 0;

    /**
     * Valor máximo del rango de números aleatorios (inclusivo).
     */
    private static final int VALOR_MAXIMO = 100;

    /**
     * Método principal que genera números aleatorios y los
     * escribe en la salida estándar.
     *
     */
    public static void main(String[] args) {
        // Generador de números aleatorios
        Random random = new Random();

        // Generar y mostrar los números aleatorios
        for (int i = 0; i < CANTIDAD_NUMEROS; i++) {
            // Generar un número aleatorio en el rango [VALOR_MINIMO, VALOR_MAXIMO]
            int numeroAleatorio = random.nextInt(VALOR_MAXIMO - VALOR_MINIMO + 1) + VALOR_MINIMO;

            // Escribir el número en la salida estándar
            System.out.println(numeroAleatorio);
        }
    }
}
```

Después de escribir el código, hemos generado un jar y generado javadocs (botón derecho en el proyecto>clean and build o botón derecho>generar javadoc, y ejecutado en terminal de la siguiente manera:



Podemos ver que se generan los números aleatorios correctamente.

Con sus correspondientes javadoc(siento que esté en alemán, es que tengo el lenguaje del sistema en este idioma porque me voy a presentar a un examen de certificación. Klassen=Clase, Beschreibung= Descripción)



Klassen

Klasse	Beschreibung
aleatorios	Aplicación que genera números aleatorios entre 0 y 100 y los escribe en la salida estándar.

El código de ordenarNumeros sería el siguiente:

ordenarNumeros.java [-/A] × aleatorios.java [-/A] ×

Source History

```java
import java.util.ArrayList;
import java.util.Collections;
import java.util.Scanner;

/**
 * Aplicación que ordena un conjunto indeterminado de números
 * recibidos a través de la entrada estándar y muestra el resultado
 * ordenado en la salida estándar.
 *
 * @author Rodrigo
 * @version 1.0
 */
public class ordenarNumeros {

    /**
     * Método principal que lee números de la entrada estándar,
     * los ordena y los muestra por la salida estándar.
     *
     */
    public static void main(String[] args) {
        // Lista para almacenar los números leídos
        ArrayList<Integer> numeros = new ArrayList<>();

        System.out.println("Introduce números para que sean ordenados: (cualquier otro carácter para finalizar)");

        // Scanner para leer de la entrada estándar
        Scanner scanner = new Scanner(System.in);

        // Leer todos los números disponibles en la entrada estándar
        while (scanner.hasNextInt()) {
            int numero = scanner.nextInt();
            numeros.add(numero);
        }

        // Cerrar el scanner
        scanner.close();

        // Ordenar los números de menor a mayor
        Collections.sort(numeros);

        // Mostrar los números ordenados en la salida estándar
        System.out.println("Números ordenados:");
        for (Integer numero : numeros) {
            System.out.println(numero);
        }
    }
```

De la misma manera, comprobamos que se ejecuta correctamente ordenarNumeros.

```
→  dist git:(main) ✗ java -jar ordenarNumeros.jar
Introduce números para que sean ordenados: (cualquier otro carácter para finalizar)
4
8
75
2
46
q
Números ordenados:
2
4
8
46
75
→  dist git:(main) ✗
```

Y su javadoc correspondiente muestra en el navegador:

Klassen

Klasse	Beschreibung
ordenarNumeros	Aplicación que ordena un conjunto indeterminado de números recibidos a través de la entrada estándar y muestra el resultado ordenado en la salida estándar.

Por último, vamos a ejecutar con la tubería.

```
→  Procesos git:(main) ✗ ls
 aleatorios   martinez_tabernero_luis_rodrigo_PSP01_Tarea   ordenarNumeros  'PSP01 Programación multiproceso.pdf'
→  Procesos git:(main) ✗ java -jar aleatorios/dist/aleatorios.jar | java -jar ordenarNumeros/dist/ordenarNumeros.jar
Introduce números para que sean ordenados: (cualquier otro carácter para finalizar)
Números ordenados:
4
6
11
12
13
21
24
30
30
30
34
36
36
36
37
37
40
40
41
42
44
45
48
51
54
62
65
66
67
67
69
80
81
82
85
88
90
91
93
100
→  Procesos git:(main) ✗
```

Y vemos que ha ordenado los números aleatorios