



PRÁCTICA 1

TEMA 3

MARIO JIMÉNEZ MARSET

ÍNDICE

| | |
|--------------------------------------|---|
| 1. DESARROLLO – PROCEDIMIENTOS | 3 |
|--------------------------------------|---|

1. DESARROLLO – PROCEDIMIENTOS

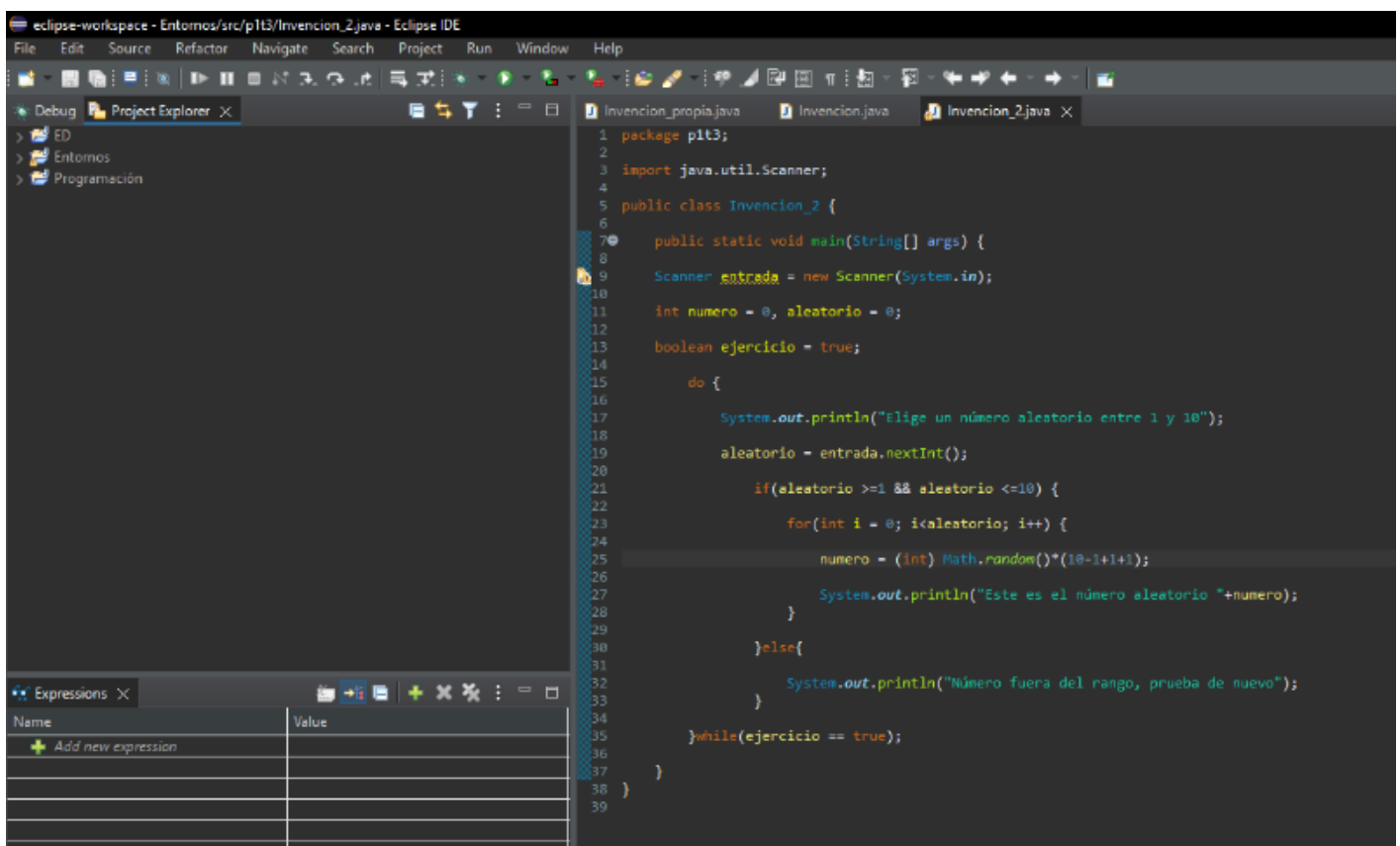
En primer lugar, se debía realizar un pequeño programa con el que poder realizar el test de depuración.

Este, consiste en la elección de un número de elementos entre 1 y 10, el cual se introduce por consola a través de la clase “Scanner”, anteriormente importada.

Después, se impone la condición de ser uno o varios números entre ese rango de cifras a través de una estructura básica de control “if”, dentro del cual se introduce un “for”, donde se le da el valor random a la variable “numero” (entre 1 y 10) y se saca por pantalla (si se dice que sean 5 los elementos a sacar, el programa deberá de sacar 5 números aleatorios dentro del rango de números pedido).

Si el numero introducido (aleatorio) está fuera del rango, se sacará por pantalla un mensaje pidiendo de nuevo que se saque un número dentro de ese rango.

Para que se vuelva a repetir el ciclo, se ha utilizado un “do while”, donde la condición para cumplirse es que una variable booleana declarada sea “true”, lo cual se va a cumplir siempre (en este caso).



```
1 package pit3;
2
3 import java.util.Scanner;
4
5 public class Inversion_2 {
6
7     public static void main(String[] args) {
8
9         Scanner entrada = new Scanner(System.in);
10
11         int numero = 0, aleatorio = 0;
12
13         boolean ejercicio = true;
14
15         do {
16
17             System.out.println("Elige un número aleatorio entre 1 y 10");
18
19             aleatorio = entrada.nextInt();
20
21             if(aleatorio >=1 && aleatorio <=10) {
22
23                 for(int i = 0; i<aleatorio; i++) {
24
25                     numero = (int) Math.random()*(10-1+1+1);
26
27                     System.out.println("Este es el número aleatorio "+numero);
28                 }
29             }else{
30
31                 System.out.println("Número fuera del rango, prueba de nuevo");
32             }
33
34         }while(ejercicio == true);
35
36     }
37
38 }
39
```

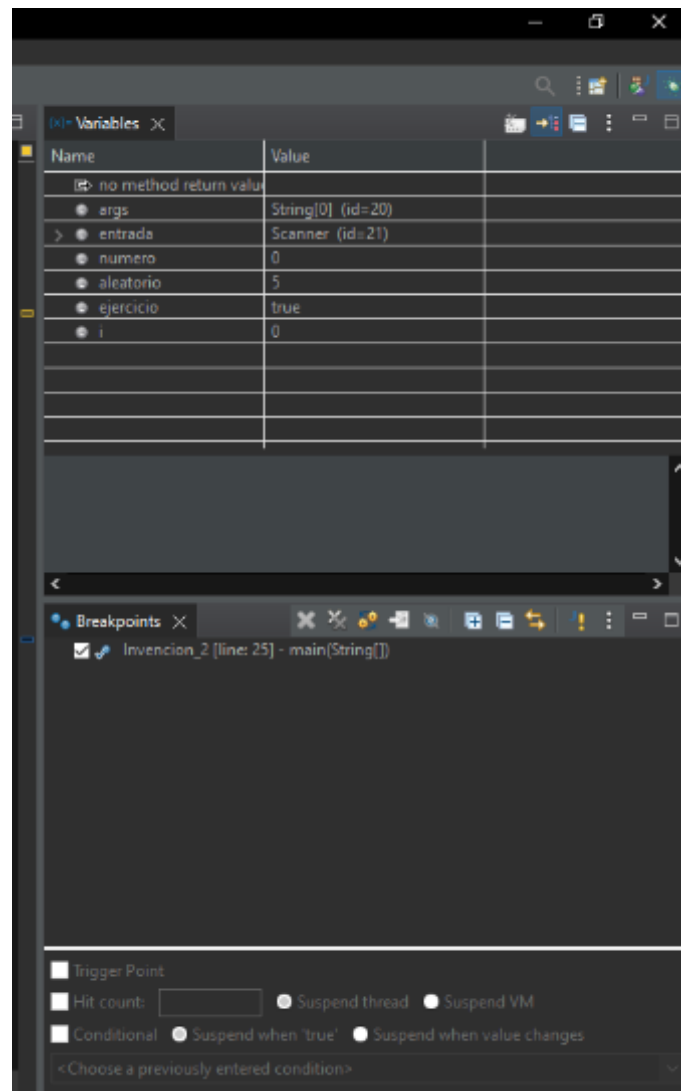
Entonces, con el pequeño programa en funcionamiento, se puede realizar el test de depuración para encontrar posibles fallos (hay sólo un fallo puesto intencionadamente para la resolución del “debugging”).

Para encontrar el error o errores, en primer lugar, se establecen los “breakpoints”. Los “breakpoints” son puntos que rompen el flujo del programa. Justo en la línea de código donde se establece, será hasta donde el programa pueda ejecutar.

En este caso, se establecerá un “breakpoint” en la línea 25: esto quiere decir que el programa se ejecutará hasta esa línea.

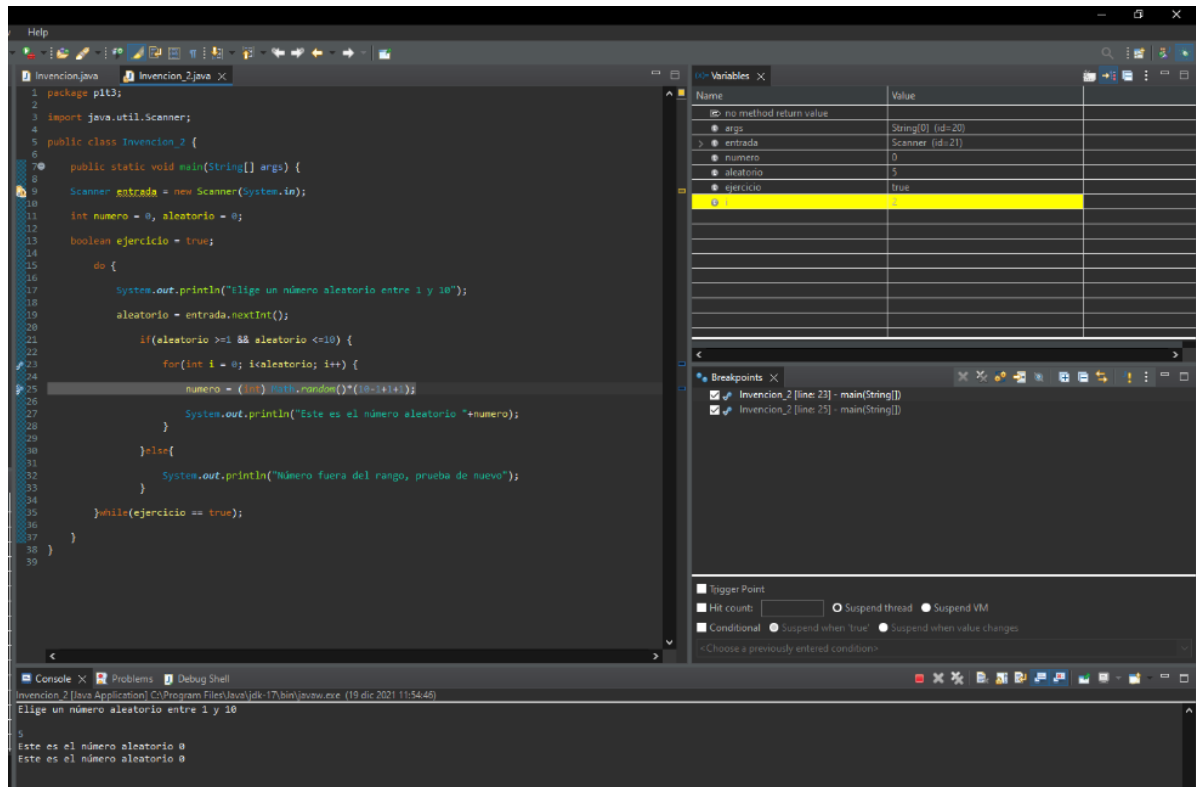
Al clicar en el botón “debugging”, empieza el programa. Por ejemplo, en la entrada del scanner se ha introducido el número “5” (el cual está dentro del rango que pide el programa). Entonces, Eclipse muestra en la ventana “Variables” las propias variables del programa.

Dentro de esas variables, aparece el “5” que se ha introducido manualmente, además de la variable “i” introducida en el “for”, con valor igual a 0. Además, en el apartado de “breakpoints” aparece el único breakpoint que se ha puesto en el programa hasta el momento.



Eclipse muestra la línea de código pausada resaltándola.

Si se pulsa el botón “Step Over” (o F6) Eclipse salta a la siguiente línea de código después de la que se ha pausado. Esto sirve para poder ejecutar el programa mientras se tiene el panel de “Variables” delante, donde las variables que van cambiando se subrayan de color amarillo, y obviamente van variando su valor.



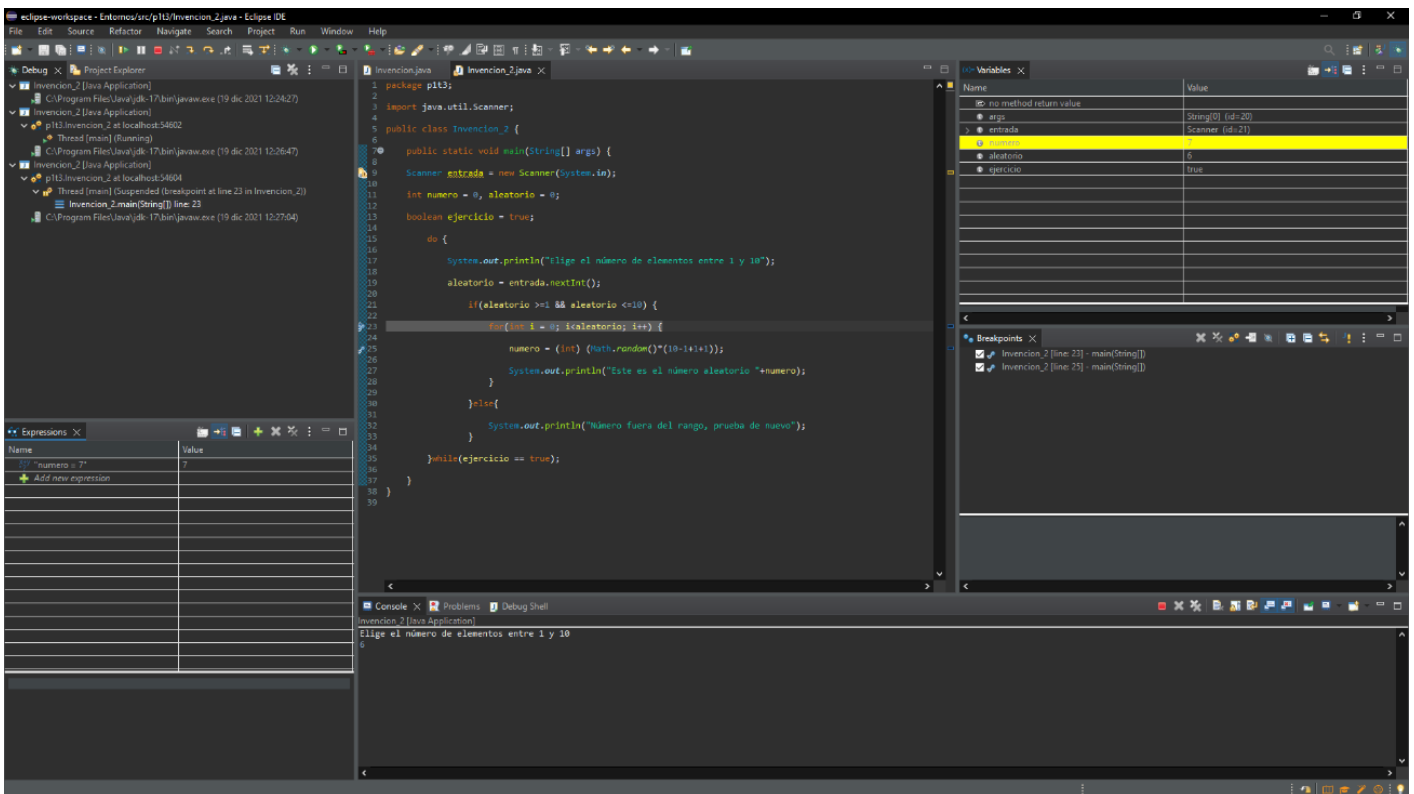
Al pulsar este botón, se ha ejecutado el “System.out.println”, el cual muestra el número que se ha introducido manualmente. Sin embargo, al clicar varias veces el F6, muestra por pantalla el número “0”, en vez del “5”.

Esto demuestra que hay un fallo entre esas líneas.

Para ver dónde está ese fallo, se ha introducido otro “breakpoint” en el “for”.

Al hacer varios F6, se ve que el valor de la variable “i” va cambiando cada vez que se ejecuta el propio “for”. Esto demuestra, que el fallo del programa que hace que el número impreso en pantalla sea 0 está dentro del valor dado a “numero”.

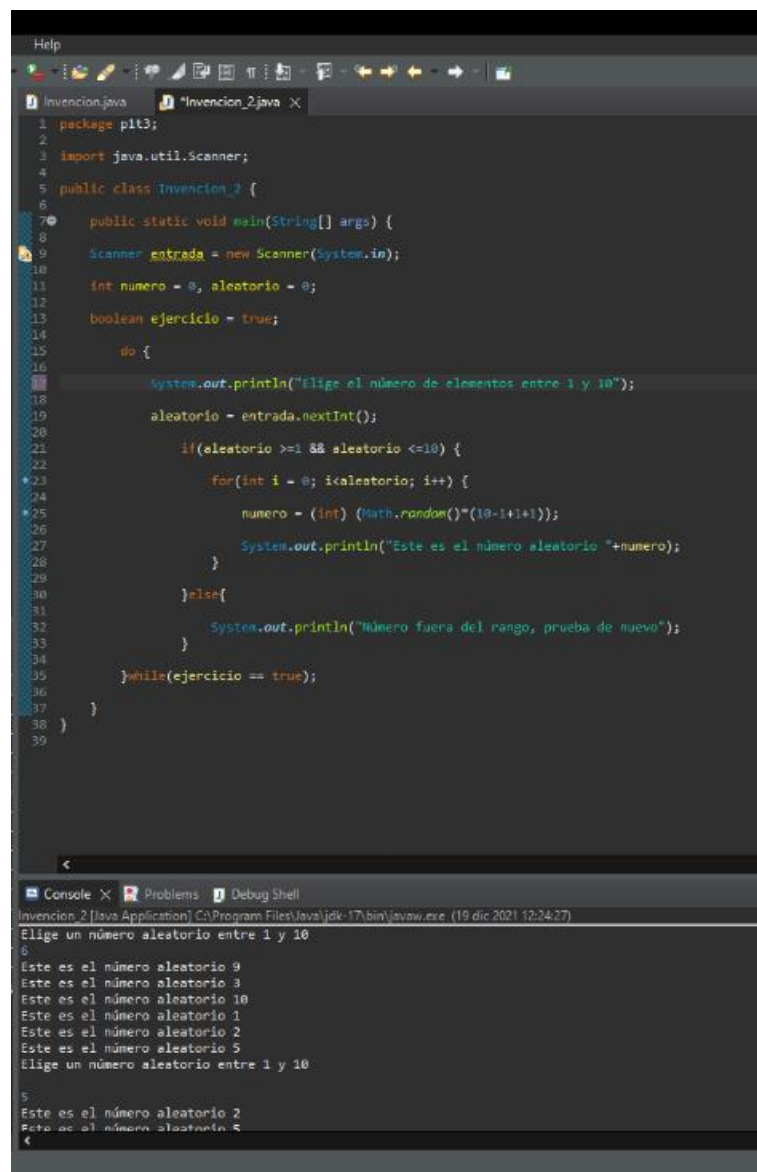
Existe otra ventana llamada “Expressions” donde, si se declara una variable y su valor (en este caso, “numero” igual a 7, dará igual a ese valor de 7).



{En esta imagen se aprecia el nuevo valor dado a “numero” dentro de “Expressions”, el cual se recoge en “Variables”}

Además, también existe la posibilidad de que, si hay un programa que se “comunica” con otros programas (por la programación orientada a objetos), y hay un error ocasionado por uno de ellos, en la línea de código del otro programa que se comunica con ese programa, si se pulsa “Step into” o F5, se hace un salto hacia la línea del otro programa para ir en busca del fallo.

En conclusión, esta es una forma muy interesante de buscar y encontrar los fallos en el código. Finalmente, el error en este caso estaba en dos paréntesis dentro del “Math.random” de la variable “numero”, sin los cuáles se hacía mal la operación que se pretendía.



```
Help
Inversion.java  Inversion_2.java X
1 package pit3;
2
3 import java.util.Scanner;
4
5 public class Inversion_2 {
6
7     public static void main(String[] args) {
8
9         Scanner entrada = new Scanner(System.in);
10
11         int numero = 0, aleatorio = 0;
12         boolean ejercicio = true;
13
14         do {
15
16             System.out.println("Elige el número de elementos entre 1 y 10");
17
18             aleatorio = entrada.nextInt();
19
20             if(aleatorio >= 1 && aleatorio <= 10) {
21
22                 for(int i = 0; i < aleatorio; i++) {
23
24                     numero = (int) (Math.random() * (10 - 1 + 1 + 1));
25
26                     System.out.println("Este es el número aleatorio " + numero);
27                 }
28             } else {
29                 System.out.println("Número fuera del rango, prueba de nuevo");
30             }
31         } while(ejercicio == true);
32     }
33 }
34
35
36
37
38
39
```

```
Console X Problems Debug Shell
Inversion_2 [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (19 dic 2021 12:24:27)
Elige un número aleatorio entre 1 y 10
6
Este es el número aleatorio 9
Este es el número aleatorio 3
Este es el número aleatorio 10
Este es el número aleatorio 1
Este es el número aleatorio 2
Este es el número aleatorio 5
Elige un número aleatorio entre 1 y 10
5
Este es el número aleatorio 2
Este es el número aleatorio 5
€
```

El programa finalmente funciona como se deseaba.