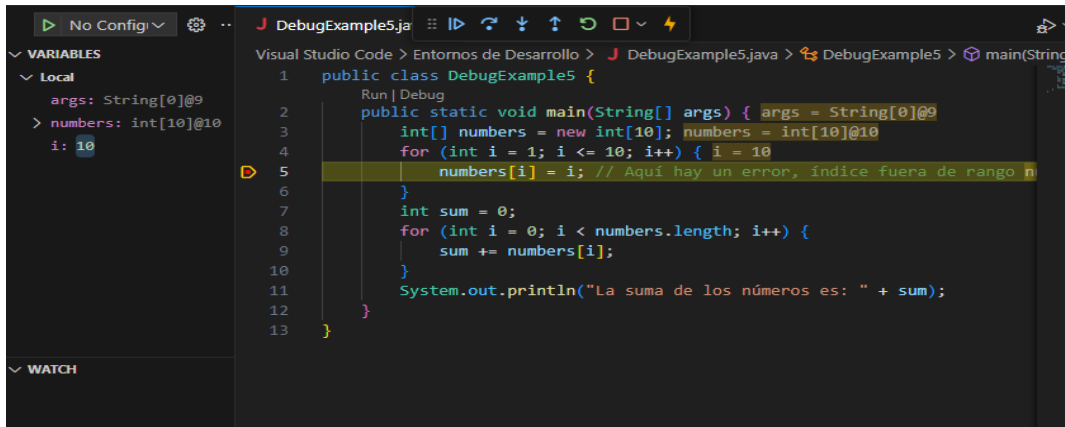


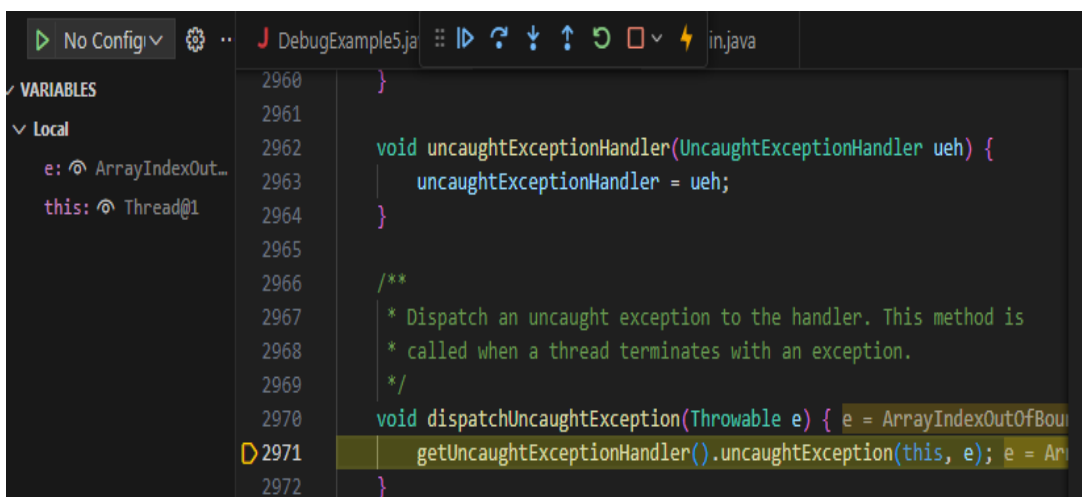
## Entornos de desarrollo – entrega 3 – UD 2

### 1. Ejercicio 1 VSC:



```
1 public class DebugExample5 {
2     public static void main(String[] args) { args = String[0]@9
3         int[] numbers = new int[10]; numbers = int[10]@10
4         for (int i = 1; i <= 10; i++) { i = 10
5             numbers[i] = i; // Aquí hay un error, índice fuera de rango n
6         }
7         int sum = 0;
8         for (int i = 0; i < numbers.length; i++) {
9             sum += numbers[i];
10        }
11        System.out.println("La suma de los números es: " + sum);
12    }
13 }
```

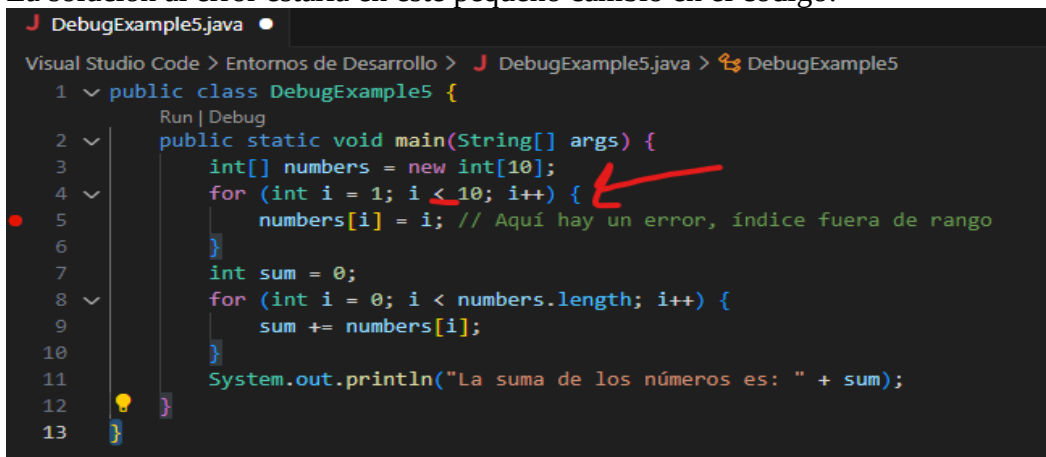
Primero llega a  $i = 10$  y luego, al exceder el tamaño del array, salta la excepción `ArrayIndexOutOfBoundsException`. Esto sucede porque en los arrays la primera posición es el 0 y no el 1, por lo que un array de tamaño  $N$  (10 en este caso) comenzará en 0 y terminará en  $N-1$  (9 en este caso).



```
2960 }
2961
2962 void uncaughtExceptionHandler(UncaughtExceptionHandler ueh) {
2963     uncaughtExceptionHandler = ueh;
2964 }
2965
2966 /**
2967  * Dispatch an uncaught exception to the handler. This method is
2968  * called when a thread terminates with an exception.
2969  */
2970 void dispatchUncaughtException(Throwable e) { e = ArrayIndexOutOfBou
2971     getUncaughtExceptionHandler().uncaughtException(this, e); e = Ar
2972 }
```

Aquí podemos ver la excepción sin tratar (try/catch).

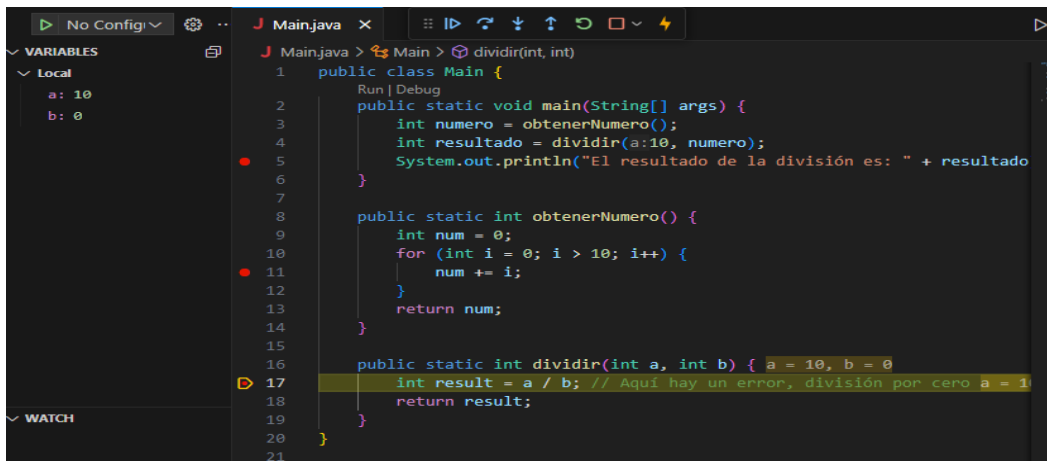
La solución al error estaría en este pequeño cambio en el código:



```
1 public class DebugExample5 {
2     public static void main(String[] args) {
3         int[] numbers = new int[10];
4         for (int i = 1; i < 10; i++) { ←
5             numbers[i] = i; // Aquí hay un error, índice fuera de rango
6         }
7         int sum = 0;
8         for (int i = 0; i < numbers.length; i++) {
9             sum += numbers[i];
10        }
11        System.out.println("La suma de los números es: " + sum);
12    }
13 }
```

Bastaría con cambiar la condición del bucle for para que la última iteración fuese la que toca ( $i = 9$ , ya que si  $i \leq 10$ , llegará a ser 10, pero  $i < 10$  nunca llegará a ser 10).

## 2. Ejercicio 2 VSC:

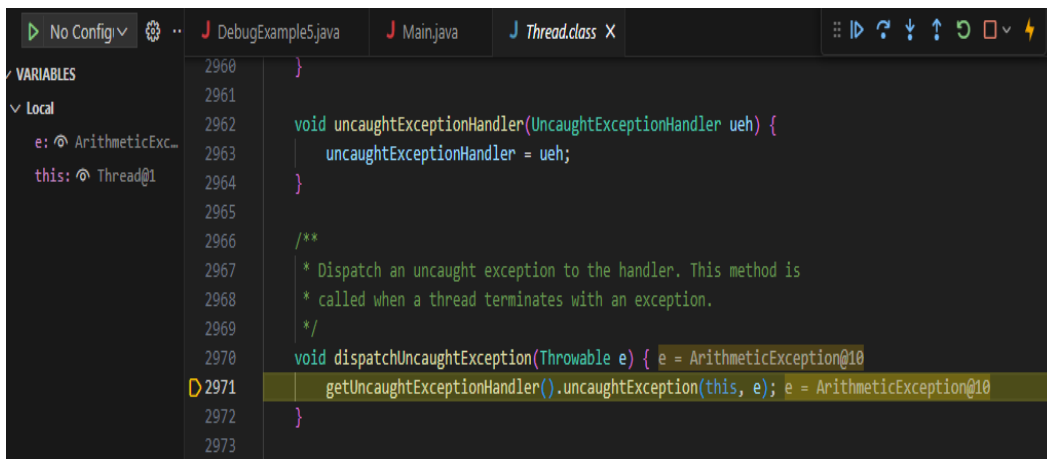


The screenshot shows the VS Code editor with a Java file named `Main.java`. The code defines a `Main` class with three methods: `main`, `obtenerNumero`, and `dividir`. The `main` method calls `obtenerNumero` and then `dividir` with arguments `a=10` and `numero=0`. The `dividir` method attempts to calculate `a / b`, which results in a runtime exception because `b` is zero. The error message in the console says: `Exception in thread "Main" java.lang.ArithmeticException: / by zero`. The variables `a` and `b` are shown in the `VARIABLES` pane as `a: 10` and `b: 0`.

```
1 public class Main {
2     public static void main(String[] args) {
3         int numero = obtenerNumero();
4         int resultado = dividir(a:10, numero);
5         System.out.println("El resultado de la división es: " + resultado);
6     }
7
8     public static int obtenerNumero() {
9         int num = 0;
10        for (int i = 0; i > 10; i++) {
11            num += i;
12        }
13        return num;
14    }
15
16    public static int dividir(int a, int b) {
17        int result = a / b; // Aquí hay un error, división por cero a = 1
18        return result;
19    }
20 }
21
```

Al producirse el error en la línea 17, la ejecución del método `dividir()` termina abruptamente y no llega a retornar resultado alguno (línea 18). El fragmento de código que no se llega a alcanzar es el printeo por pantalla del resultado (línea 5), ya que no se alcanza el “return” del método (línea 18). Esto sucede porque el método `main()` se ejecuta de manera lineal, pero al llegar a una llamada a otro método (`dividir`, por ejemplo) se ejecuta de manera lineal DENTRO de dicho método (líneas 17 y 18 en este caso) para luego seguir con la línea 5. Además, en la línea 11 hay un fragmento de código que tampoco es alcanzable, ya que la condición del bucle `for` nunca se va a cumplir porque 0 nunca puede ser mayor que 10.

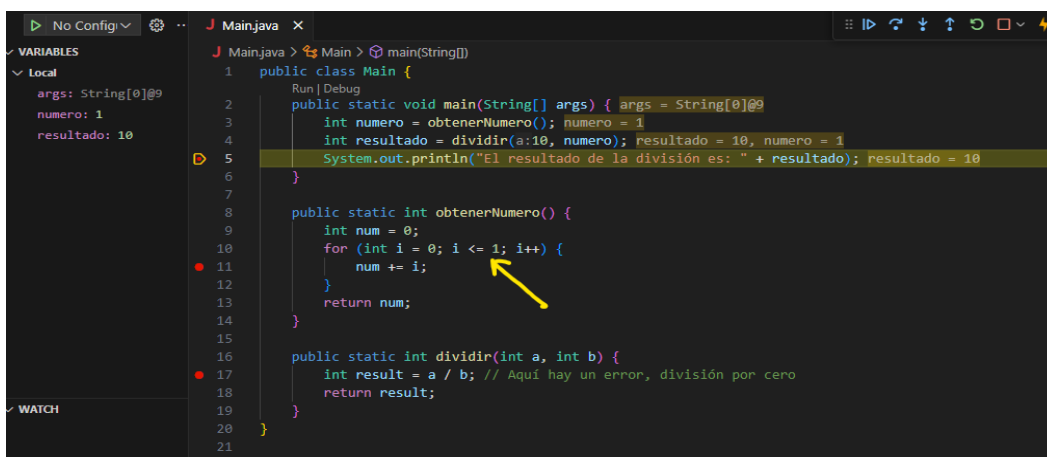
**Nota:** De hecho, existen 3 líneas que no son alcanzables: 18, 5 y 11.



The screenshot shows the VS Code editor with a Java file named `DebugExample5.java`. The code defines a `Thread` class with two methods: `uncaughtExceptionHandler` and `dispatchUncaughtException`. The `dispatchUncaughtException` method calls `getUncaughtExceptionHandler().uncaughtException(this, e)`, which results in an `ArithmeticException` being thrown. The error message in the console says: `Exception in thread "Thread@1" java.lang.ArithmeticException: / by zero`. The variables `e` and `this` are shown in the `VARIABLES` pane as `e: @ ArithmeticExc...` and `this: @ Thread@1`.

```
2960 }
2961
2962 void uncaughtExceptionHandler(UncaughtExceptionHandler ueh) {
2963     uncaughtExceptionHandler = ueh;
2964 }
2965
2966 /**
2967  * Dispatch an uncaught exception to the handler. This method is
2968  * called when a thread terminates with an exception.
2969  */
2970 void dispatchUncaughtException(Throwable e) {
2971     getUncaughtExceptionHandler().uncaughtException(this, e); e = ArithmeticException@10
2972 }
2973
```

Se puede apreciar la excepción sin tratar (`ArithmeticException: / by zero`).



The screenshot shows the VS Code editor with a Java file named `Main.java`. The code defines a `Main` class with three methods: `main`, `obtenerNumero`, and `dividir`. The `main` method calls `obtenerNumero` and then `dividir` with arguments `a=10` and `numero=1`. The `dividir` method attempts to calculate `a / b`, which results in a runtime exception because `b` is zero. The error message in the console says: `Exception in thread "Main" java.lang.ArithmeticException: / by zero`. The variables `args`, `numero`, and `resultado` are shown in the `VARIABLES` pane as `args: String[0]@9`, `numero: 1`, and `resultado: 10`. A yellow arrow points to the `for` loop condition in the `obtenerNumero` method, which is `for (int i = 0; i <= 1; i++)`.

```
1 public class Main {
2     public static void main(String[] args) {
3         int numero = obtenerNumero();
4         int resultado = dividir(a:10, numero);
5         System.out.println("El resultado de la división es: " + resultado);
6     }
7
8     public static int obtenerNumero() {
9         int num = 0;
10        for (int i = 0; i <= 1; i++) {
11            num += i;
12        }
13        return num;
14    }
15
16    public static int dividir(int a, int b) {
17        int result = a / b; // Aquí hay un error, división por cero
18        return result;
19    }
20 }
21
```

Para solventar el error basta con cambiar la condición del `for` a `(int i = 0; i <= 1; i++)`.