Netbeans: Depurando

Depurar nos permitirá examinar el código de las aplicaciones para buscar errores, ya que posibilita observar las líneas que se van ejecutando, así como los valores que van tomando las variables en cada paso

Para realizar la depuración de un programa, se debe establecer en primer lugar un **punto de interrupción** donde deberá pararse la ejecución de la aplicación y en ese punto examinaremos el valor de las variables. A partir de ahí podremos seguir depurando hasta el siguiente punto de interrupción o ejecutar el programa de forma "normal".

Vamos a ver como depurar utilizando el IDE NetBeans:

1. Abrimos el proyecto a depurar

Mediante NetBeans abrimos el proyecto que creamos en la práctica anterior.
Recordemos que era una cuenta de 1 a 10 utilizando la instrucción while.

```
package cuenta10;

/**
 * main class
 * @author David Escribano
 *  Programa que realiza una cuenta de 1  a 10 utilizando while
 */
public class Cuenta10 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        int cont=1;
        int fin=10;
        System.out.println("Inicio cuenta ");
        while (cont<=fin){
            System.out.println("Contador " + cont );
            cont++;
            }
    }

}
```

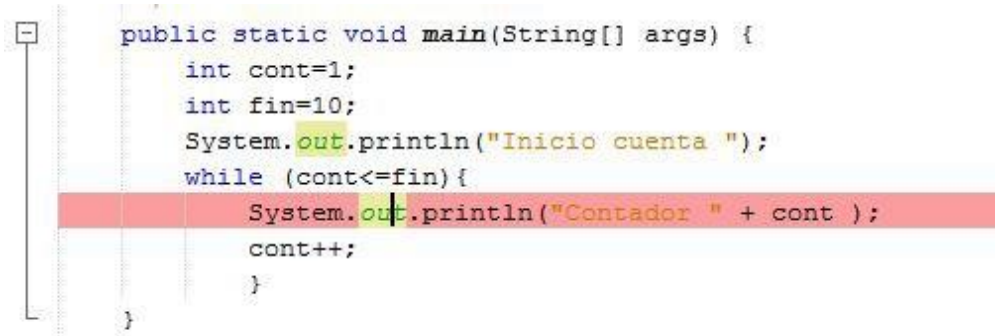
2. Establecemos puntos de interrupción

Para establecer un punto de interrupción realizaremos alguna de las siguientes acciones sobre la línea de código en la que se desee establecer el punto de interrupción:

- **Clic en el margen izquierdo**
- Menú contextual >Toggle Line Breakpoint
- Pulsando la combinación de teclas: Ctrl + F8
- Menú "Depurar >Toggle Line Breakpoint

Al realizar alguna de esas acciones, se marca en color rosado la línea que se ha convertido en un punto de interrupción, y se muestra un pequeño cuadrado en el margen izquierdo

Insertamos un punto de interrupción sobre la línea que muestra el valor del contador, en mi caso la línea 24



3. Ejecución de la aplicación en modo depuración

Una vez establecido al menos un punto de interrupción, se debe **ejecutar la aplicación en modo depuración**. Para esto se puede llevar a cabo sobre el proyecto o sólo sobre el archivo actual.

- **Para depurar archivo actual:**
  - Menú contextual > "Debug *File*"
  - Menú "Depurar > Debug *nombreArchivo*"
  - Pulsando la combinación de teclas: Ctrl + Mayúsculas + F5

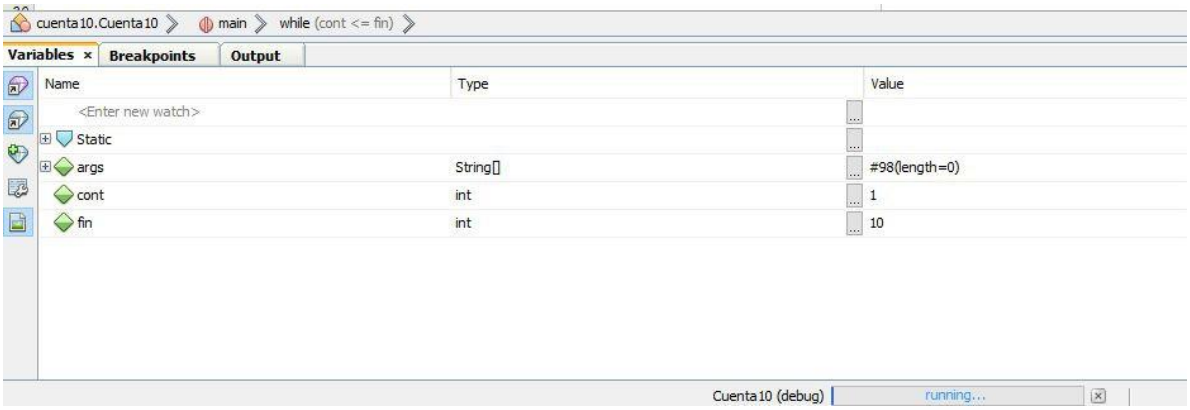
- **Para depurar proyecto:**
  - Menú "Depurar > Debug Main Project"

Iniciamos depuración del archivo actual.

Nos aparece en la parte inferior un panel con tres pestañas.

- **Variables**: donde observaremos el valor de las variables locales. *En nuestro caso cont y fin.*

- **Breakpoints**: Puntos de ruptura introducidos.
- **Output:** Terminal de salida



Cuando iniciamos la depuración las variables muestran cont 1 y fin 10, el programa se detiene en el punto de interrupción.
Observa que la instrucción donde ha llegado aparece en color verde.
Una vez se detiene podemos observar el valor de las variables

```
 */
public class Cuenta10 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        int cont=1;
        int fin=10;
        System.out.println("Inicio cuenta ");
        while (cont<=fin){
            System.out.println("Contador " + cont );
            cont++;
            }
    }

}
```

A partir del momento en que se para la ejecución del programa se puede continuar con la ejecución línea a línea utilizando la opción **Debug>Step Over (F8)**

Ejecutamos **Debug>Step Over** y *se deberá detener en la siguiente instrucción cont++, observa que aparece en verde.*

```
public static void main(String[] args) {
    int cont=1;
    int fin=10;
    System.out.println("Inicio cuenta ");
    while (cont<=fin){
        System.out.println("Contador " + cont );
        cont++;
        }
}
```

A partir de ahí mediante **Debug>continue** o pulsando **F5** continuará la ejecución de la depuración hasta el siguiente punto de ruptura.

Así se va mostrando en verde la línea que se va ejecutando en cada momento, y se van **actualizando** en la ventana inferior los valores que van tomando las variables

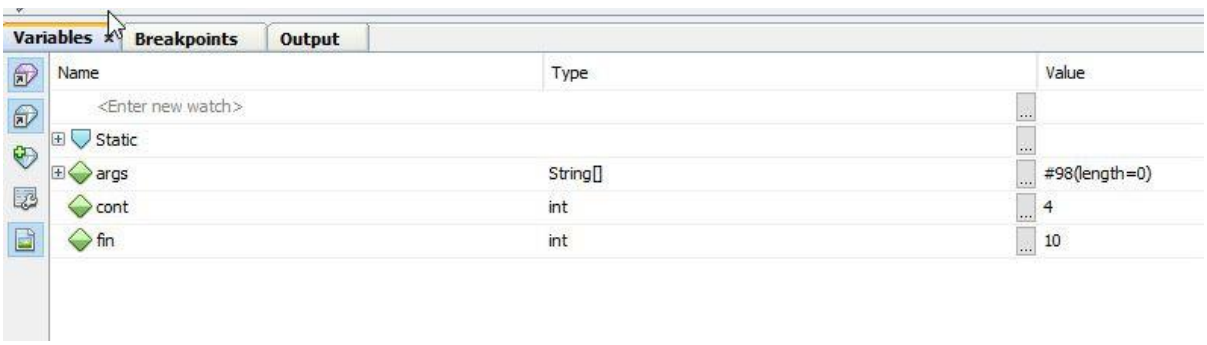
Voy ejecutando paso a paso

```
public class Cuenta10 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        int cont=1;
        int fin=10;
        System.out.println("Inicio cuenta ");
        while (cont<=fin){
            System.out.println("Contador " + cont );
            cont++;
            }
    }

}
```

Observa cómo se va modificando el valor de las variable cont



Practicando

Realiza capturas de pantalla que demuestren realización

Realiza la depuración del programa Adivinan: Comprueba paso a paso si el programa funciona correctamente.

```
package Adivina;
import java.util.Scanner;

public class Adivinan {
    public static void main(String[] args) {
        int num_adiv=0;
        int num_usu=0;
        int veces=0;
        Scanner sc=new Scanner(System.in);
        num_adiv=(int)(Math.random()*100);
        do {
           System.out.println("Dame número");
           num_usu=sc.nextInt();
           sc.nextLine();
           veces++;

           if (num_usu<num_adiv) {
                 System.out.println ("El num buscado es mayor");
           }else {
                 if (num_usu>num_adiv) {
                       System.out.println ("El num buscado es menor");
                 }
                 else {
                       System.out.println ("Acertaste en "+ veces);
                 }
           }
         }while (num_adiv!=num_usu);
    }
}
```