

# Depuración con Puntos de Ruptura con el IDE NetBeans

En ocasiones no estamos interesados en ejecutar un programa instrucción por instrucción sino que sólo deseamos que el programa se detenga después de ejecutar ciertas instrucciones y nos permita inspeccionar los valores de algunas variables. En esos casos podemos colocar en esas instrucciones unas marcas llamadas puntos de ruptura que hacen que el programa se detenga antes de la ejecución de dichas instrucciones.

## Colocación de Puntos de Ruptura

Para colocar un punto de ruptura en un programa usando puntos de ruptura se sigue el siguiente procedimiento:

1. Utilizando la herramienta de desarrollo **NetBeans**, creé un proyecto llamado **capital**.
2. Haga que este proyecto sea el proyecto principal y establezca el nombre de la clase principal a **DemoCapital** dentro del paquete **pruebas**.
3. Cree una clase llamada **Capital** dentro del paquete **objetosNegocio** y edite el código de la clase **Capital** como se muestra en el siguiente listado:

### Capital.java

```
/*
 * Capital.java
 *
 * Creada el 11 de octubre de 2005, 12:36 PM
 */

package objetosNegocio;

/**
 * Esta clase permite tabular el capital acumulado de una inversión,
 * a una tasa de interés anual, con recapitalización mensual.
 * También determina el número de meses necesarios para que una
 * inversión
 * se duplique
 *
 * Ilustra el uso de la sentencia while
 *
 * @author mdomitsu
 */
public class Capital {
    private double capitaInicial;
```

```

private double tasa;
private int meses;

/**
 * Constructor que inicializa los atributos de la clase al valor
 * de sus parametros
 * @param capitaInicial Capital inicial
 * @param tasa Tasa de interes anual
 * @param meses Meses a tabular
 */
public Capital(double capitaInicial, double tasa, int meses) {
    this.capitaInicial = capitaInicial;
    this.tasa = tasa;
    this.meses = meses;
}

/**
 * Este metodo tabula el capital acumulado de una inversión,
 * a una tasa de interés anual, con recapitalización mensual
 */
public void tabla() {
    int mes = 1;
    double capital = capitaInicial;

    System.out.println("Mes      Capital");
    while(mes <= meses) {
        capital *= (1 + tasa / 12);
        System.out.println(mes + "      " + capital);
        mes++;
    }
}

/**
 * Este metodo determina el número de meses necesarios para
 * que una inversión se duplique
 */
public void duplica() {
    int mes = 0;
    double capital = capitaInicial;
    double dobleCapital = 2* capital;

    while(capital <= dobleCapital) {
        capital *= (1 + tasa / 12);
        mes++;
    }

    System.out.println("Meses: " + mes + "      Capital: " + capital);
}
}

```

4. Edite la clase principal **DemoCapital** como se muestra en el siguiente listado:

### DemoCapital.java

```

/*
 * DemoCapital.java.
 */

```

```
package pruebas;

import objetosNegocio.Capital;

/**
 * Esta clase permite probar la clase Capital
 *
 * @author mdomitsu
 */
public class DemoCapital {

    /**
     * Esta funcion prueba los metodos de la clase Capital
     * @param args Argumentos en la linea de comando
     */
    public static void main(String[] args) {
        DemoCapital demoCapital = new DemoCapital();

        Capital capital = new Capital(100.0, 0.60, 12);
        System.out.println("Capital Acumulado");
        capital.tabla();
        System.out.println("\nMeses para que se duplique el capital");
        capital.duplica();
    }
}
```

5. En el Panel de Edición haga clic en la pestaña del panel con el código de la clase Capital para que su código sea visible.
6. Coloque el cursor sobre la línea en la que se encuentre la instrucción donde quiere un punto de ruptura, en este caso la instrucción es:

```
mes++;
```

dentro del método `tabla()`.

7. Haga clic con el botón derecho y seleccione la opción **Debug/Toggle Breakpoint** o presione las teclas **Ctrl + F8**, Figura 1.
8. La línea con el punto de ruptura aparece resaltada con el color rojo, Figura 2.

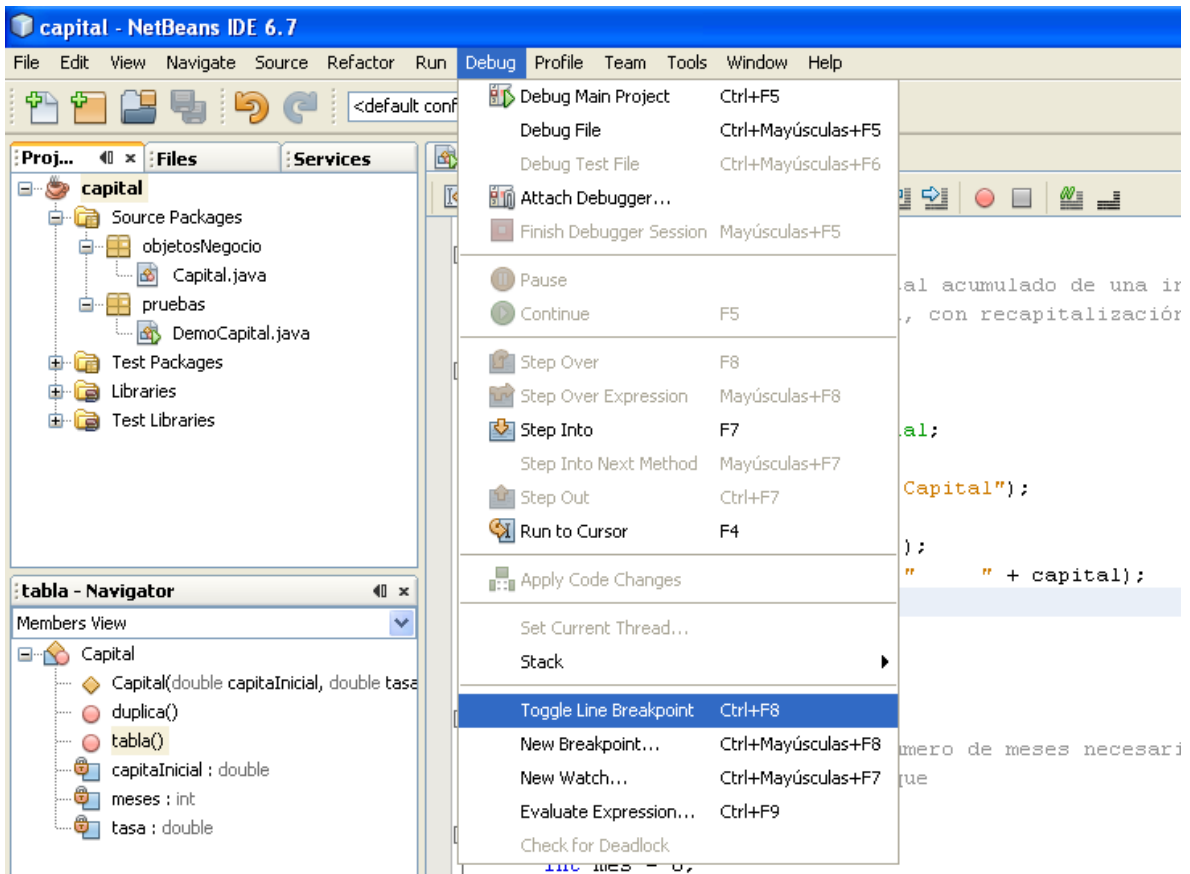


Figura 1

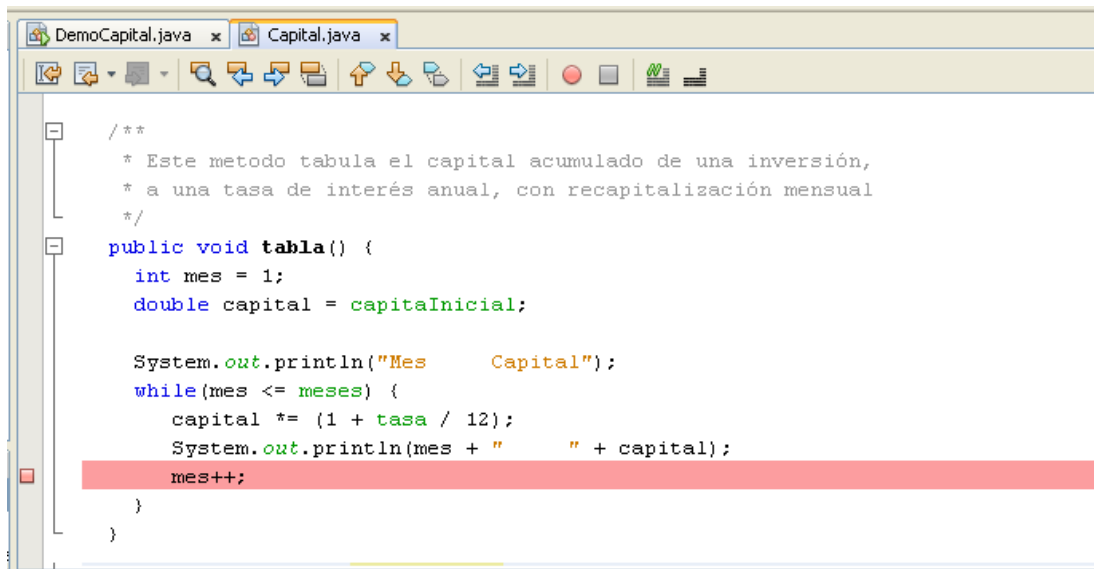


Figura 2

9. Otra forma de establecer un punto de ruptura es hace clic sobre el margen izquierdo del panel de edición a la altura de la instrucción en la que se desea colocar el punto de ruptura, Figura 3.

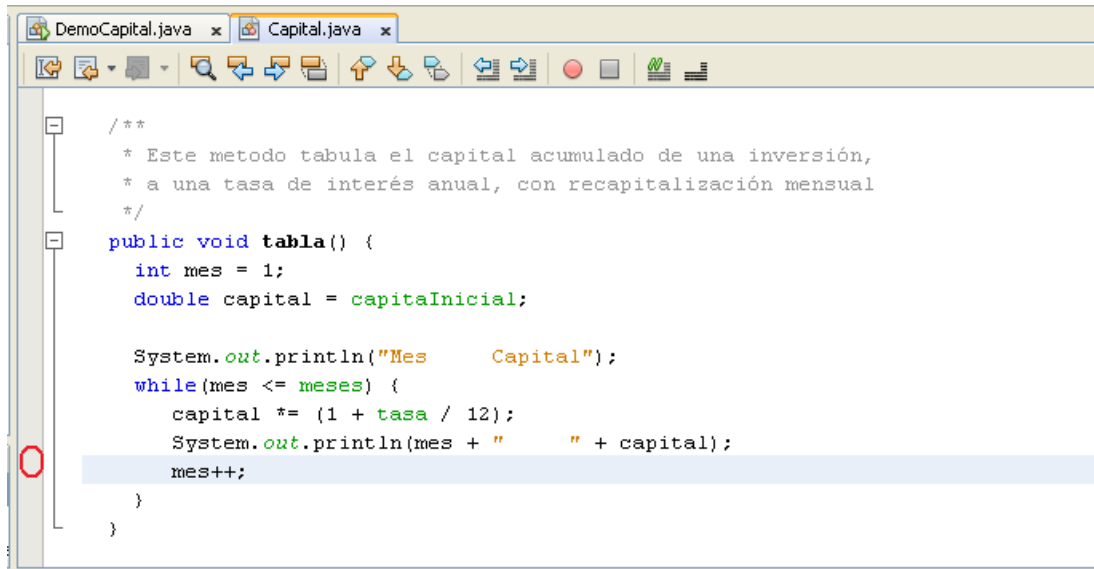


Figura 3

10. La línea con el punto de ruptura aparece resaltada con el color rojo, Figura 2.
11. Para eliminar un punto de ruptura de una instrucción se repite el procedimiento empleado para establecerlo: pasos 6 y 7 o el paso 9.

## Ejecución de un Programa Desde el inicio del Programa Hasta el Primer Punto de Ruptura

Para iniciar la ejecución de un programa desde el principio hasta el primer punto de ruptura se sigue el siguiente procedimiento:

1. Seleccione del menú principal la opción **Debug/Debug Main Project** o presione la tecla **Ctrl + F5** como se ve en la Figura 4. Otra forma de realizar esta tarea es hacer clic en el icono **Debug Main Project** de la barra de tareas, Figura 5.

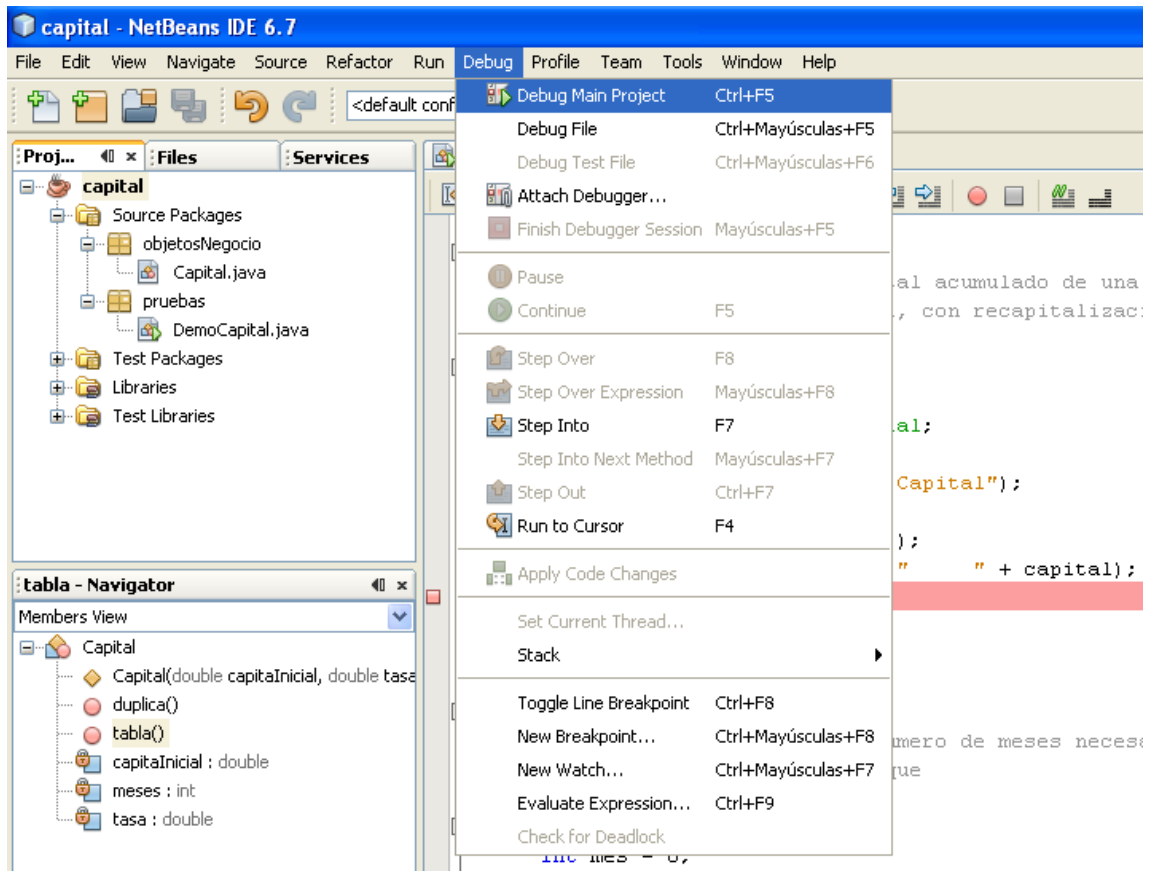


Figura 4

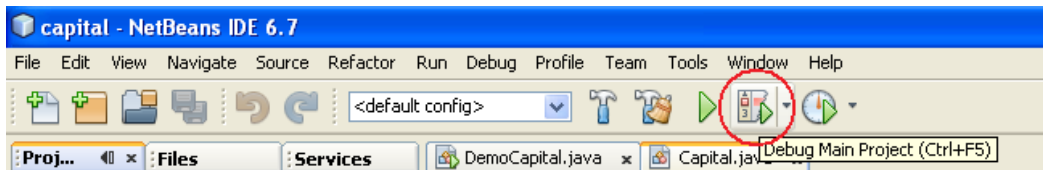


Figura 5

2. Al hacerlo, el programa ejecutará las instrucciones hasta antes de la instrucción donde se encuentra el punto de ruptura, Figura 6.
3. Como ya se ejecutó una iteración del ciclo, en el panel de salida se puede ver que ya se escribió una línea de la tabla, Figura 7.
4. Podemos inspeccionar los valores de los atributos de la clase y de las variables del método en el panel Variables, Figura 8. Para ver los atributos hay que expandir el nodo **this**, haciendo clic en el icono + que se encuentra a la izquierda del nodo.

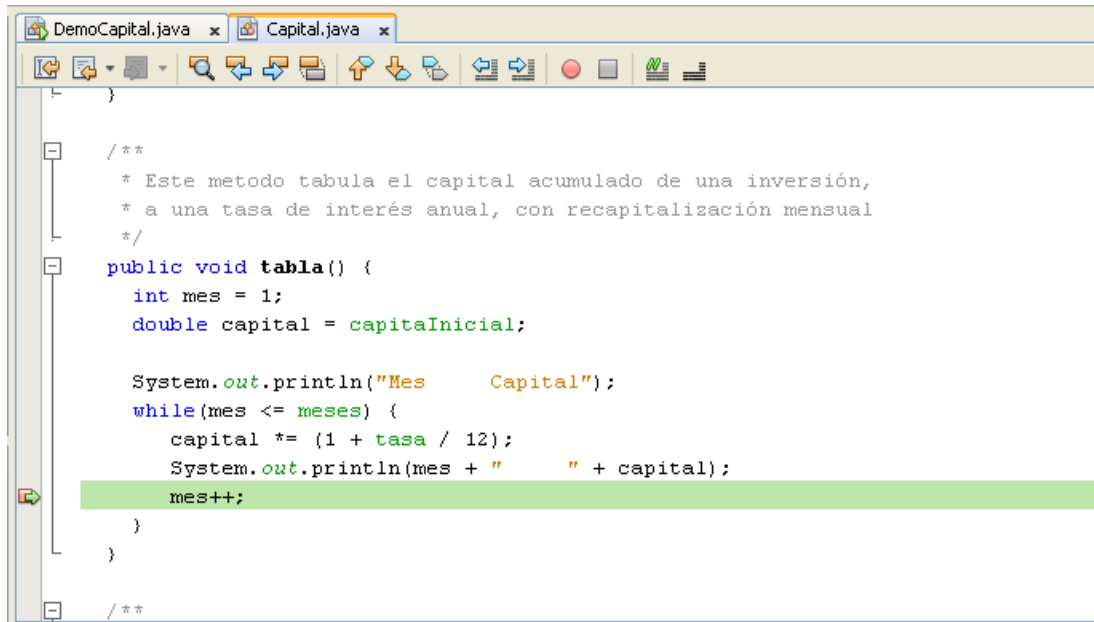


Figura 6

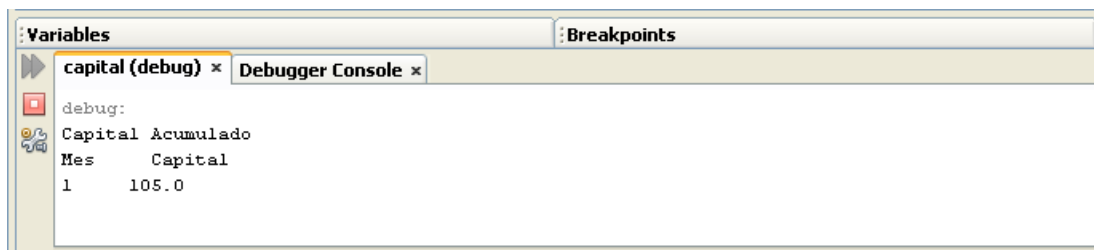


Figura 7

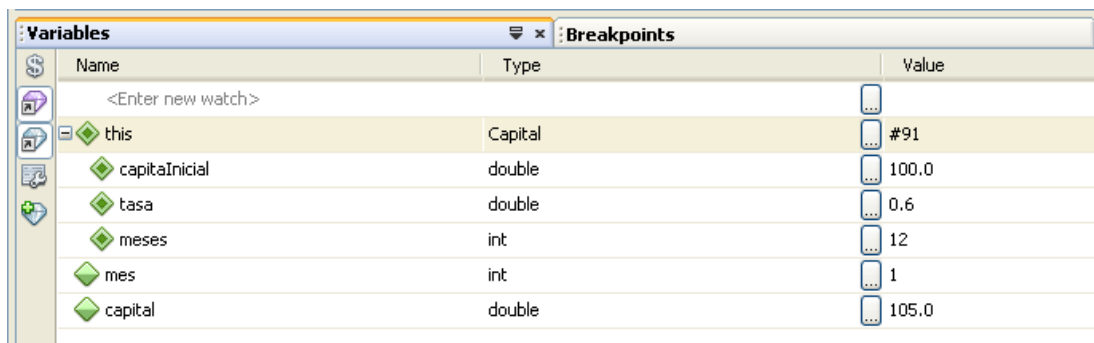


Figura 8

## Ejecución de un Programa Hasta el Siguiete Punto de Ruptura

Para ejecutar el programa hasta el siguiente punto de ruptura se sigue el siguiente procedimiento:

1. Seleccione del menú principal la opción **Debug/Continue** o presione la tecla **F5** como se ve en la Figura 9. Otra forma de realizar esta tarea es hacer clic en el icono **Continue** de la barra de tareas, Figura 10.

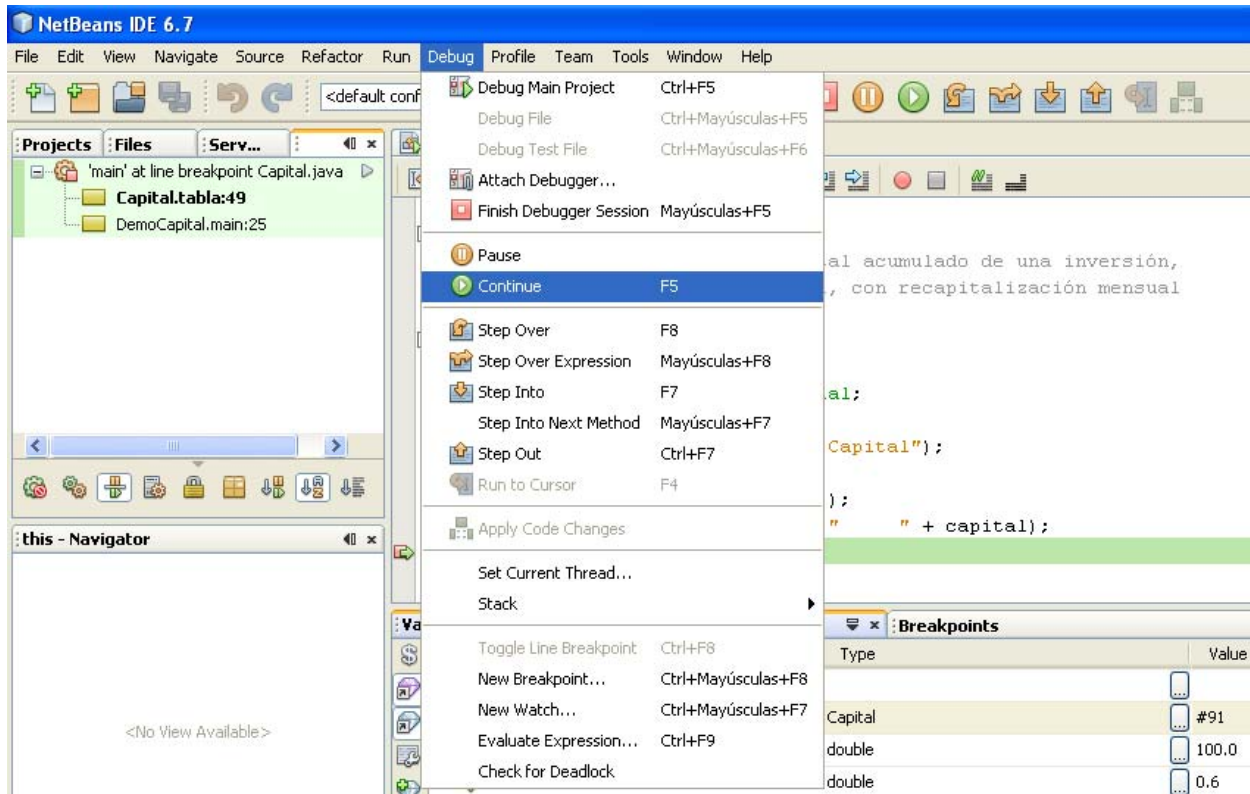


Figura 9



Figura 10

2. Al hacerlo, NetBeans ejecutará las instrucciones hasta antes de la instrucción con el siguiente punto de ruptura. En este caso como el único punto de ruptura se encuentra dentro de un ciclo, el programa ejecutará otra iteración deteniéndose en la misma instrucción. Podemos verificar que se ejecutó la iteración observando la salida del programa en la que se puede ver que se ha desplegado la segunda línea de la tabla, Figura 11 o inspeccionando la variable mes en el panel Variables, Figura 12.
3. Repita los pasos 1 y 2 para ejecutar cada una de las iteraciones del ciclo, observando en cada iteración la salida del programa y los valores de las variables del método. Deténgase cuando la variable mes tome el valor de 12.



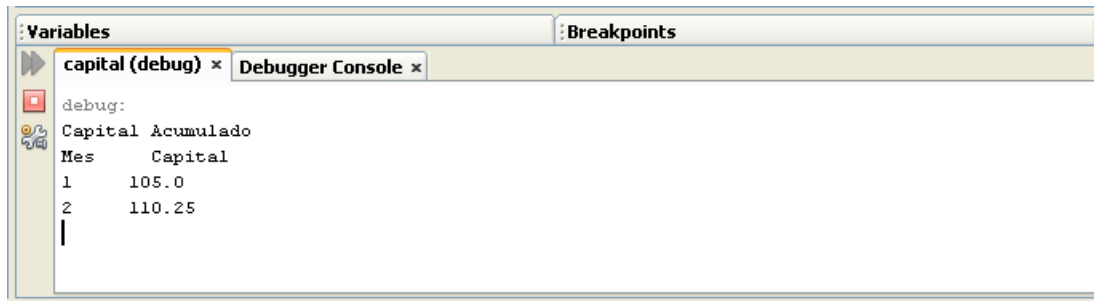


Figura 11

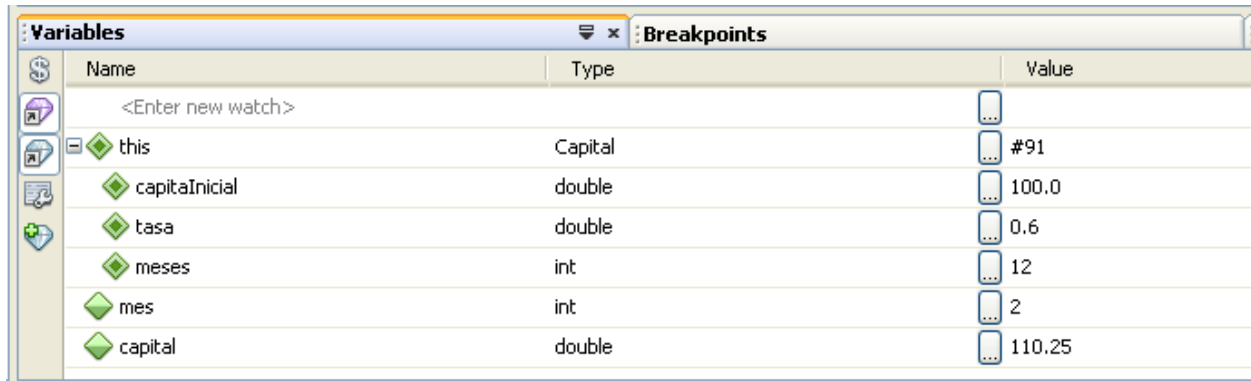


Figura 12

- Si se vuelve a presionar la tecla **F5** el programa continuará su ejecución. En este caso como el ciclo termina, el depurador ya no encontrará otro punto de ruptura y el programa ejecutará hasta el final, mostrando el resto de los resultados, Figura 13 y terminando la sesión de depuración.

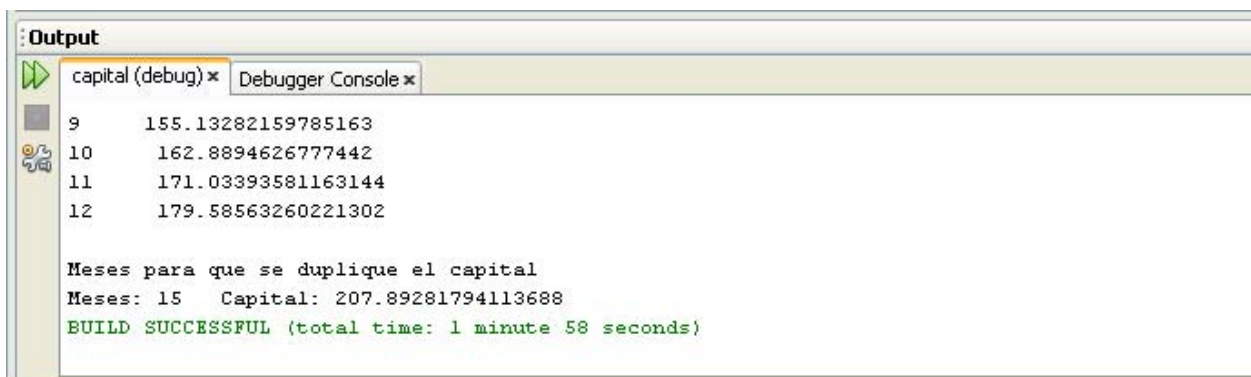


Figura 13