Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela Ciencias y Sistemas
Melissa Beatriz Liz Florecita Rivas Lucas
Carnet: 201504476

Convenciones de Java

Las convenciones ayudan a que los programas sean más entendibles, es decir haciéndolos más fácil de leer. Proporcionan información sobre las funciones del identificador.

Nomenclatura en Java

Clases e interfases

- La primer letra de cada palabra en mayúscula y todas las palabras siguientes minúsculas. Esta nomenclatura también es llamada lomo de camello o CamelCase
- Para las clases, los nombres deben de ser sustantivos después de la palabra reservada “class”.
- Para la interfase, los nombres son adjetivos y después de la palabra reservada “interface”
- Intentar mantener los nombres de las clases simples y descri-ptivos.
- Use palabras completas, evite acrónimos y abreviaturas

```
01.  Interface  Bicycle
02.  Class MountainBike implements Bicyle
03.
04.  Interface Sport
05.  Class Football implements Sport
```

Paquetes

- Todo se debe escribir en minúscula
- El prefijo del nombre de un paquete se escribe siempre con letras ASCII
- Si se utilizan paquetes dentro de otros paquetes, se unen mediante un punto (.)

```
01.  com.sun.eng
02.  com.apple.quicktime.v2
03.
04.  // java.lang packet in JDK
05.  java.lang
```

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela Ciencias y Sistemas
Melissa Beatriz Liz Florecita Rivas Lucas
Carnet: 201504476

Variables

- Los nombres de la variables, excepto las constantes, empezaran con minúscula.
- La primera letra de cada palabra interna en mayúsculas.
- Los nombres de variables deberían ser cortos y llenos de significado.
- Se deben evitar los nombres de variable de un sólo carácter, excepto para variables temporales.
- No debería comenzar con un guion bajo ('_') o caracteres.

```
01.   // variables para la clase MountainBike
02.     int speed = 0;
03.     int gear = 1;
```

Métodos

- Los métodos deben de ser verbos.
- Cuando la palabra es compuesta la primera letra es en minúscula, y la primera letra de la segunda palabra es en mayúscula.

```
01.   void changeGear(int newValue);
02.   void speedUp(int increment);
03.   void applyBrakes(int decrement);
```

Constantes

- Los nombres de variables constantes de clases y las constantes ANSI deberían escribirse todo en mayúsculas.
- Las palabras deben de separarse por "-".

```
01.   static final int MIN_WIDTH = 4;
02.
03.   // Algunas variables constantes utilizadas en la clase Float predefinida
04.   public static final float POSITIVE_INFINITY = 1.0f / 0.0f;
05.   public static final float NEGATIVE_INFINITY = -1.0f / 0.0f;
06.   public static final float NaN = 0.0f / 0.0f;
```

Espacios

- Los espacios son usados para entender mejor el código y distinguir el principio y el fin.

- Las separaciones por sangría necesarias, por lo regular no son más de 4 espacios.
- Todos los operadores binarios excepto «.» se deben separar de sus operandos con espacios en blanco.

Comentarios

Los programas Java pueden tener dos tipos de comentarios:

- Los comentarios de implementación son para comentar nuestro código o para comentarios acerca de una implementación en concreto
- Los comentarios de documentación describen clases Java, interfaces, constructores, métodos y atributos. Cada comentario de documentación se encierra con los delimitadores de comentario /**...*/

Indentacion

- Evitar las líneas de más de 80 caracteres
- Cuando una expresión es demasiado larga, se recomienda romper
    - Romper después de una coma
    - Romper antes de un Operador
- Alinear la nueva línea con el comienzo de la expresión al mismo nivel de la línea anterior.

```
nombreLargo1 = nombreLargo2 * (nombreLargo3 + nombreLargo4 – nombreLargo5)
               + 4 * nombreLargo6; // PREFERIDA

nombreLargo1 = nombreLargo2 * (nombreLargo3 + nombreLargo4
                               - nombreLargo) + 4 * nombreLargo6; // EVITAR
```

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela Ciencias y Sistemas
Melissa Beatriz Liz Florecita Rivas Lucas
Carnet: 201504476

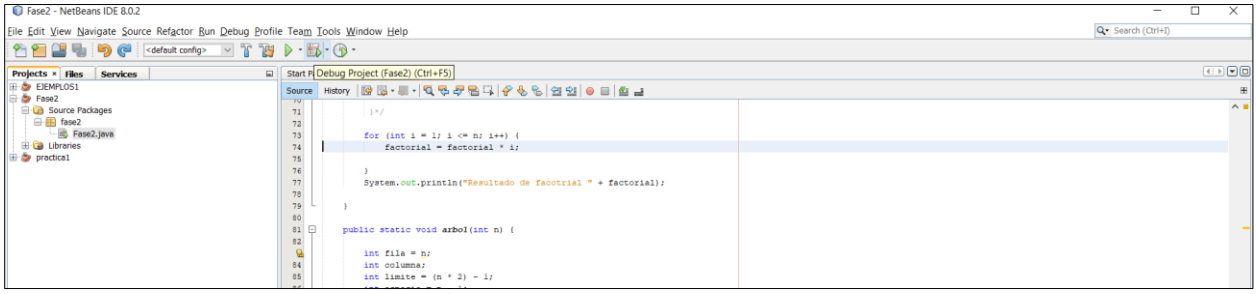
MANUAL DEBUGGER

Debugger también conocido como depurador, se encarga de depurar un programa, es decir lo recorre paso a pasos, viendo el valor de cada una de las variables. Esto permite observar exactamente qué es lo que está pasando en el programa cuando se está ejecutando una línea de código especifico.

- Finish debugger: se detiene la depuracion
- Continue (F5): La ejecución del programa continúa hasta el siguiente breakpoint.
- Step Over (F8): Ejecuta una línea de código
- Step Into (F7) Ejecuta una línea de código. Si la instrucción es una llamada a un método, salta al método y continúa la ejecución por la primera línea del método.
- Step Out : Si la línea de código actual se encuentra dentro de un método, se ejecutarán todas las instrucciones que queden del método y se vuelve a la instrucción desde la que se llamó al método

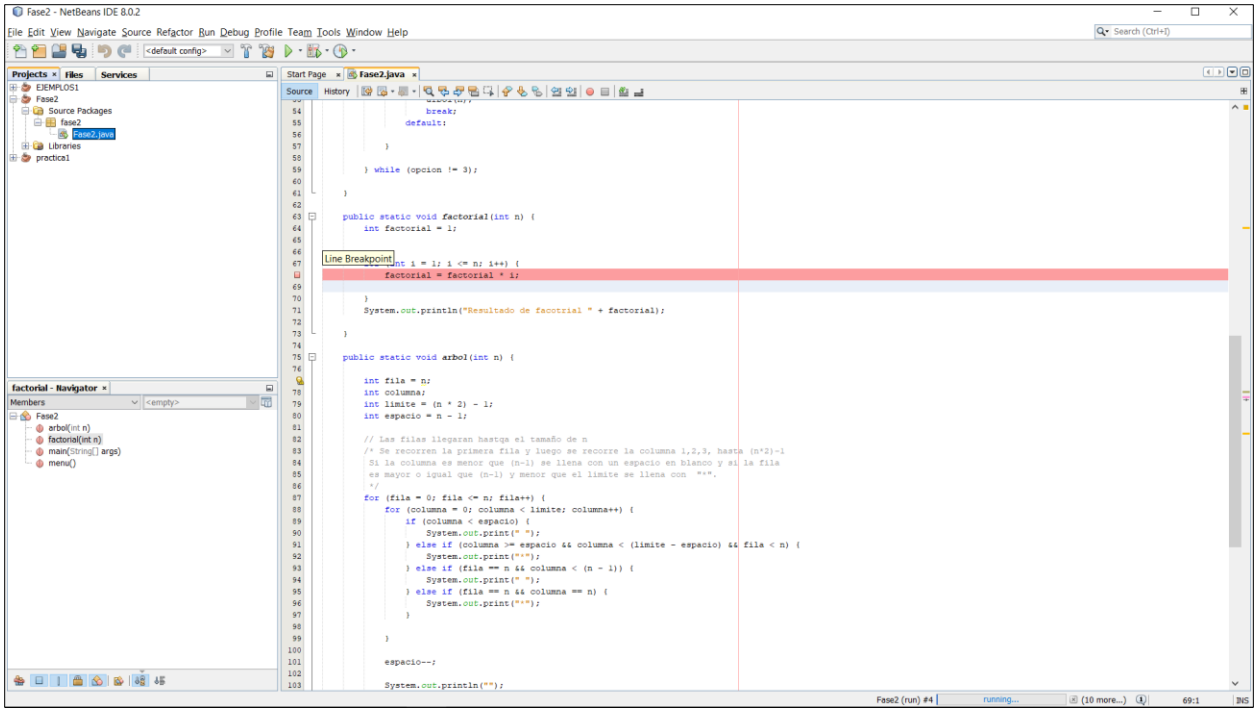
FACTORIAL – DEBUGGER EJEMPLO

1. Dar clic en Debug Proyect

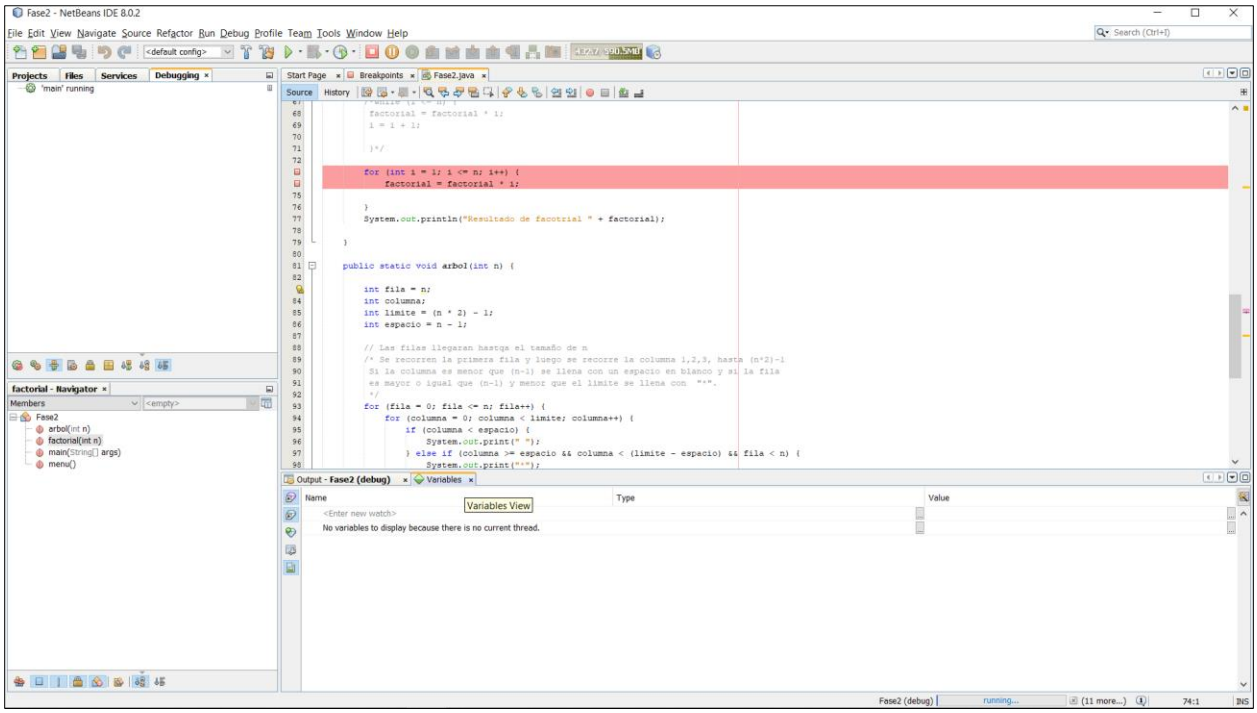


Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela Ciencias y Sistemas
Melissa Beatriz Liz Florecita Rivas Lucas
Carnet: 201504476

2. Se seleccionan las posiciones que se desean evaluar, se elige linea breakpoint.

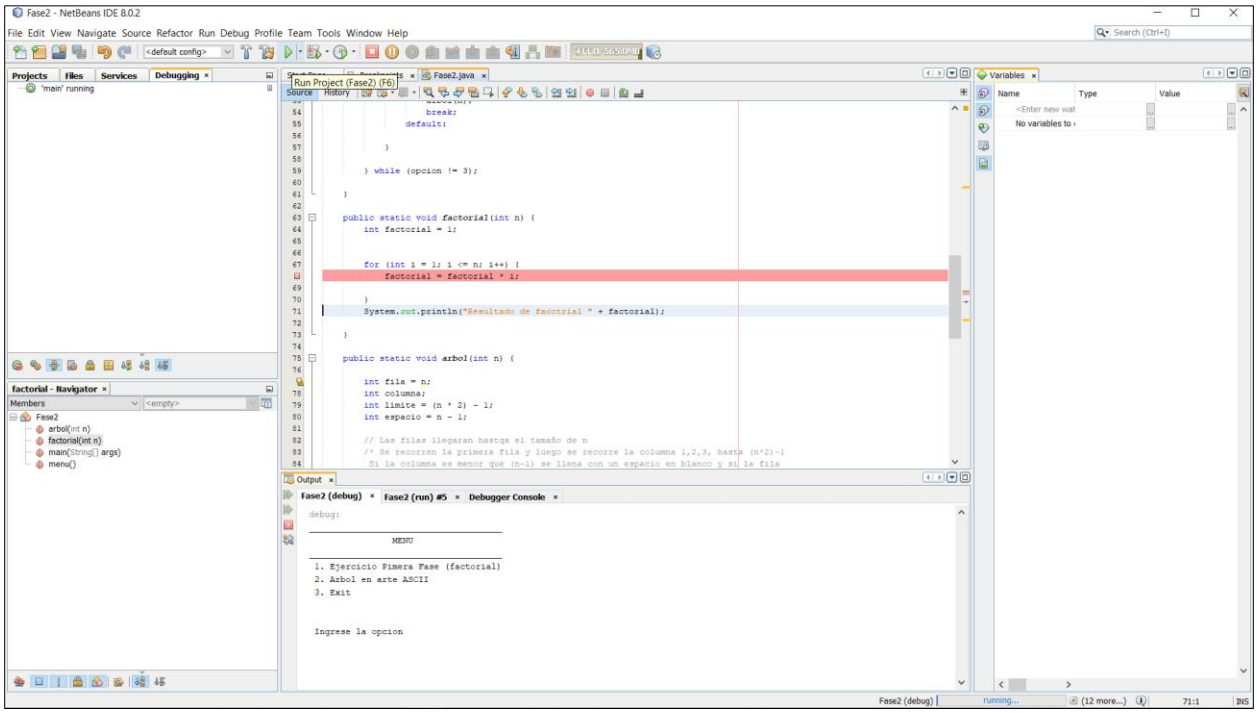


3. En esta sección se observan las variables del programa, y sus valores conforme la ejecución.

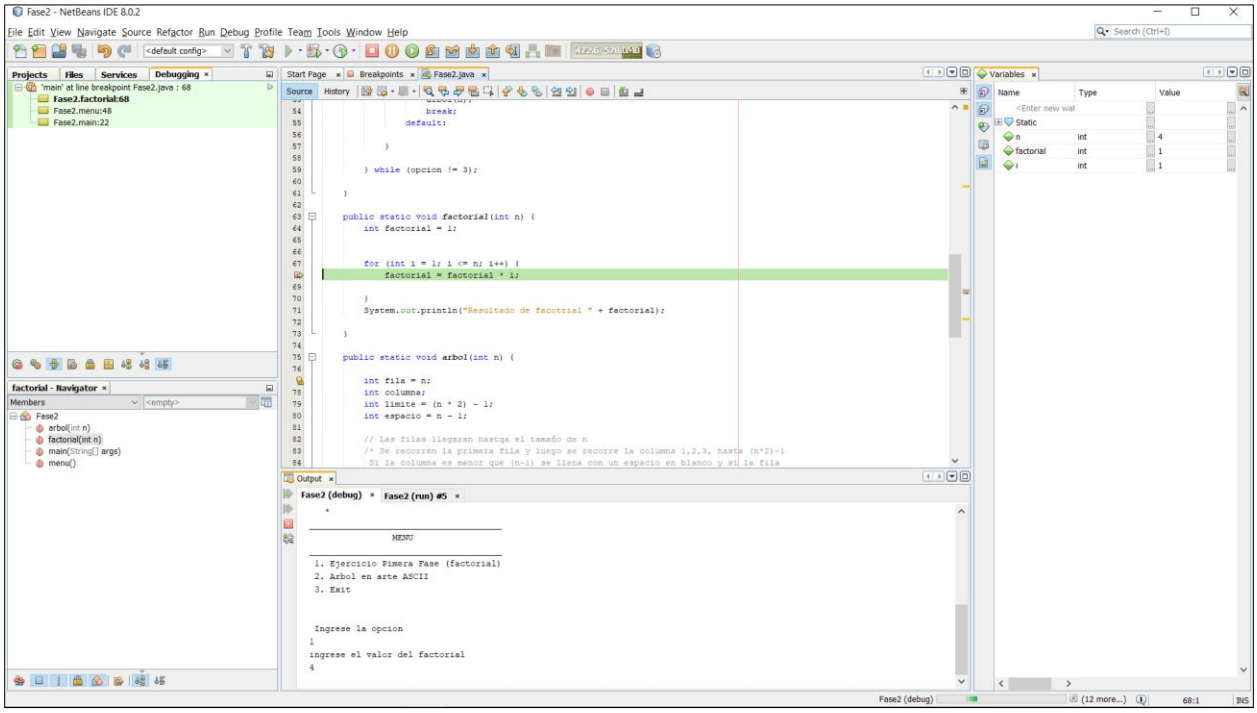


Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela Ciencias y Sistemas
Melissa Beatriz Liz Florecita Rivas Lucas
Carnet: 201504476

4. Clic en Run Project y empieza a ejecutar

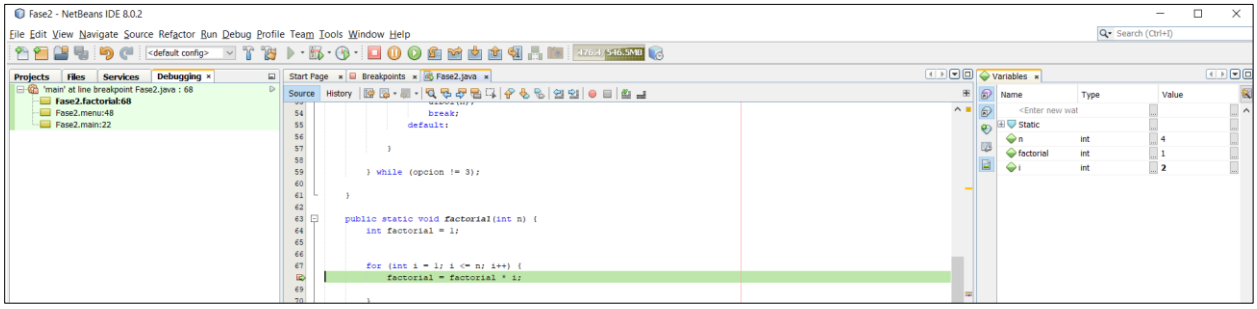


5. Ingreso la opción y luego el valor deseado, y se marca verde la línea que se está ejecutando. La variables tienen valor de i= 1 y factorial = 1.

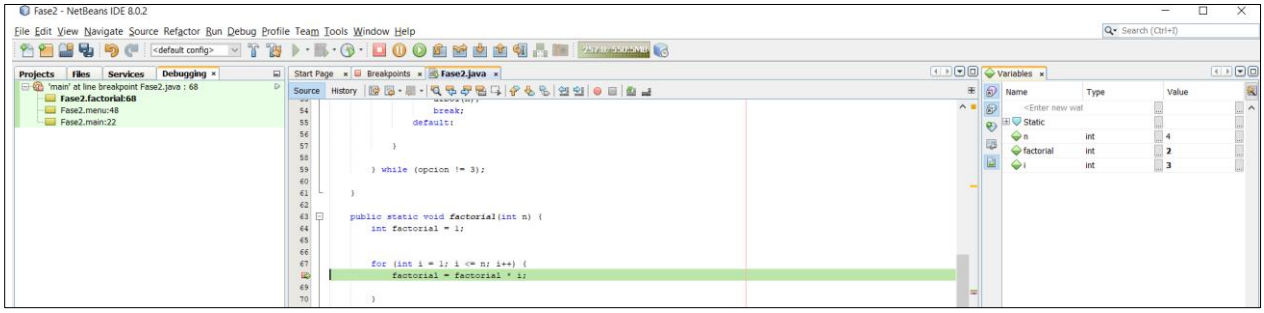


Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela Ciencias y Sistemas
Melissa Beatriz Liz Florecita Rivas Lucas
Carnet: 201504476

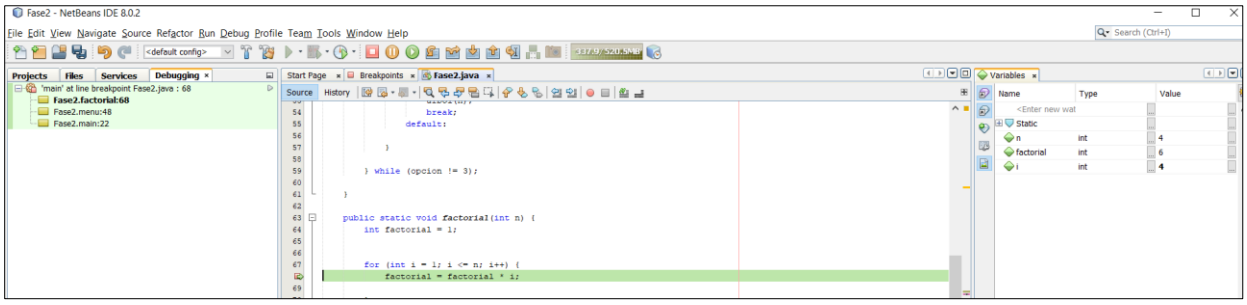
6. Le doy clic en continuar y ejecuta la siguiente instrucción. El valor se modifica I = 2.



7. Continuar y los valores son i= 3 y factorial = 2



8. Clic en Continuar y los valores son i = 4 y factorial = 6 (eso quiere decir que se opera (factorial * i)).



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela Ciencias y Sistemas
Melissa Beatriz Liz Florecita Rivas Lucas
Carnet: 201504476

9. Clic en Step Over y se finaliza el ciclo for, con el valor del factorial 24. Otra vez clic en step over y se ejecuta la linea de im