

Programando en Java

Sitio: [Agencia de Aprendizaje a lo largo de la Vida](#)
Curso: Técnicas de Programación - Turno mañana
Libro: Programando en Java

Imprimido por: Tomas Friz
Día: miércoles, 15 de septiembre de 2021, 13:12

Tabla de contenidos

- 1. Comencemos a programar!
- 2. Estructura de una clase y metodo MAIN
- 3. Uso básico de variables
- 4. Estructuras de Control - Secuenciales
- 5. Estructuras de Control - Repetitivas
- 6. Estructura repetitiva - DO WHILE
- 7. Estructuras de Control - Selección simple
- 8. Estructuras de Control - Selección doble
- 9. Estructuras de Control - Selección múltiple

1. Comencemos a programar!

Te piden realizar un programa para sumar dos números. ¿Por donde empezamos?

Veamos el problema por partes:

- 1. Comenzar describiendo el uso y las funciones que cumple el programa, el autor, la versión y alguna breve descripción.
- 2. Declarar las variables en primer lugar, que vamos a utilizar.
- 3. Usar los comandos o instrucciones que mostrarán algún mensaje descriptivo al operador describiendo el uso del programa.
- 4. Escribir las instrucciones que ejecutan el algoritmo de resolución.
- 5. Obtener el resultado.
- 6. Presentar el resultado por pantalla o impreso.

Ahora, veamos como podría quedar nuestro programa:

Pasos	Código de programación Java
1	<pre>/** * @author Diego Gomez Rivera * @version 1.0 * Ejercicio Sumardosnumeros * Sumar dos numeros * Mostrar el resultado */</pre>
2	<pre>int numero1 = 5; int numero2 = 3; int suma;</pre>
3	<pre>System.out.println("Suma de dos numeros");</pre>
4	<pre>suma = numero1 + numero2;</pre>
5 y 6	<pre>System.out.printf("El resultado es %d\n", suma);</pre>

Veamos el ejemplo completo de sumar dos números:

```
/**  
 * @author Diego Gomez Rivera  
 * @version 1.0  
 * Ejercicio Sumardosnumeros  
 * Sumar dos numeros enteros  
 * Mostrar el resultado  
 */  
  
public class Sumardosnumeros {  
  
    public static void main(String args[])  
    {  
        int numero1 = 5;  
        int numero2 = 3;  
        int suma;  
  
        System.out.println("Suma de dos numeros");  
        suma = numero1 + numero2;  
        System.out.printf("El resultado es %d\n", suma);  
    }  
}
```

Analizando el código....

- Comentarios. El asterisco * es utilizado para hacer comentarios en muchos lenguajes de programación.
- Declaración de variables: debemos declarar el tipo de variable. En este caso números enteros.
- Mostrar un mensaje: utilizamos System.out.print(). Esta instrucción puede tener distintas terminaciones (println, printf, etc.) de acuerdo a la función que vamos a realizar.

En su estructura encontramos:

- Una variable que toma el resultado llamada suma.
- Un símbolo de asignación =
- Una variable que contiene el primer número a sumar numero1
- Una variable que contiene el segundo número a sumar numero2
- Es decir, que el signo igual se lee: asignamos la suma de numero1 y numero2 a la variable suma.
- No está indicando una igualdad. Para hacer comparaciones entre valores, se utilizará el doble signo igual ==
- Finalmente, mostramos el resultado con el comando System.out.println().

Como habrá notado, en Java nos encontramos con una estructura que contiene a nuestro algoritmo de resolución, necesita empezar con los siguientes comandos:

```
public class Sumardosnumeros {  
  
    public static void main(String args[])  
    {  
        aquí va nuestro código...
```

```
    Y finalizar cerrando ambas llaves.  
    }  
}
```

Es decir que todo lo que está contenido entre la segunda llave, en este caso destacada en color verde, es nuestro código. De otra forma nuestro programa no podrá ejecutarse.

Además si nos fijamos bien, cada fin de línea termina con un punto y coma ;

2. Estructura de una clase y metodo MAIN

Ejercicio básico para explicar la estructura de una clase, el método main, la salida a consola y el uso del javadoc.

Se explicó en la clase anterior, el uso de las llaves y la estructura de un programa en Java.

Aquí podemos observar que // se usa como comentario de línea.

El programa mostrará un mensaje en pantalla.

```
/**
 * Ejercicio Nro. 2
 * Salida x consola
 * Imprimir un mensaje en la consola.
 * @author Diego Gomez Rivera
 * @version:1.0
 *
 */
public class Ej02 {
/**
 * Nombre de parámetro y descripción, para javadoc
 * @param
 * el método main empieza la ejecución de la aplicación en Java
 */
public static void main(String [] args) {
    System.out.println("Bienvenido a la programacion en Java!");
} // fin del método main

} // fin de la clase Ej02
```

3. Uso básico de variables

Ejercicio con uso de variables, numéricas y alfanuméricas.

```
/**
 *
 * @author Diego Gomez Rivera
 * @version 1.0
 * Programa para uso básico de variables
 */
public class Ej03 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {

        String nombre;
        int edad;
        String mensaje1 = "Tu nombre es... ";
        String mensaje2 = "Tu edad es... ";
        nombre = "Juan";
        edad = 27;

        System.out.println("Hola " + nombre);
        System.out.println("Tu edad es ... " + (edad * 2));
        System.out.println("Perdon " + nombre + " pense que eras mas grande...");
        System.out.println("Entonces debes tener ... " + (edad/3));
        System.out.println("Me equivoque de nuevo no " + nombre + "?");
        System.out.println("Tu nombre es " + nombre + " y tu edad es... " + edad);

    }

}
```

4. Estructuras de Control - Secuenciales

Ejercicio para el uso del método para Ingresar datos, uso de SCANNER

En este caso utilizamos Scanner, para obtener una entrada de teclado, guardar cada entrada de datos en variables enteras y luego calcular el producto entre dos números enteros ingresados.

```
/**
 *
 * @author Diego Gomez Rivera
 * @version 1.0
 * Ejercicio Nro. 11
 * uso de Scanner
 * Multiplicar dos numeros
 * Pedir un numero
 * Ingresarlo
 * Pedir otro numero
 * Ingresarlo
 * Mostrar el resultado
 */

import java.util.Scanner;

public class Ej11 {

    public static void main(String args[])
    {
        //Crear un objeto scanner para permitir la entrada de datos
        Scanner entrada = new Scanner(System.in);

        int numero1;
        int numero2;
        int producto;

        System.out.println("Producto de dos numeros");

        System.out.print("Ingrese el primer numero entero: ");
        numero1 = entrada.nextInt();

        System.out.print("Ingrese el segundo numero entero: ");
        numero2 = entrada.nextInt();

        producto = numero1 * numero2;

        System.out.printf("El resultado es %d\n", producto);

    }

}
```

5. Estructuras de Control - Repetitivas

Uso de la sentencia repetitiva FOR

Como hemos aprendido en lecciones anteriores en la sección Algoritmos, a veces la suma de muchos números, el ingreso de los datos de un grupo grande de personas, las notas de toda una comisión de alumnos, etc. implica por ejemplo, la repetición de instrucciones de forma innecesaria.

Para eso, utilizamos las iteraciones o bucles que repiten la misma secuencia hasta que se cumpla una condición.

El comando a utilizar en este primer ejemplo es el FOR.

Además en el caso de esta instrucción, todo el núcleo de la iteración se puede escribir en la misma línea de código, identificada mediante paréntesis, luego de la instrucción FOR.

Vamos a contar una secuencia de números:

Uso de CONTADORES

En el siguiente ejemplo, existe un contador, que va acumulando cada ciclo, hasta que se cumple la condición contador <=10.

Usamos el ++ para ir acumulando las veces que se repite cada ciclo y en pantalla mostramos el estado actual del contador.

```
/**
 *
 * @author Diego Gomez Rivera
 * @version 1.0
 * Ejercicio Nro. 13
 * Uso de For
 * Contadores
 */

public class Ej13 {

    public static void main(String args[])
    {
        //Crear un contador del 1 al 10 con la instrucción FOR

        System.out.println("Uso de FOR. Contadores");

        for (int contador = 1;
            contador <= 10;
            contador++)
            System.out.printf("%d ", contador);

        System.out.println();

    }

}
```

Uso de ACUMULADORES

En este caso la variable acumulador retiene el valor original, suma el valor de la variable contador, y en el ciclo siguiente repite el proceso. El resultado equivale a decir que cualquier número entero ingresado se multiplica por 10, ya que el contador para a los 10 ciclos.

```
/**
 *
 * @author Diego Gomez Rivera
 * @version 1.0
 * Ejercicio Nro. 14
 * uso de Scanner
 * Uso de For
 * Acumuladores
 */

import java.util.Scanner;
```



```
public class Ej14 {

    public static void main(String args[])
    {
        //Crear un contador del 1 al 10 con la instrucción FOR

        //Crear un objeto scanner para permitir la entrada de datos
        Scanner entrada = new Scanner(System.in);

        int acumulador;
        int numero;
        acumulador = 0; //Limpia la variable acumulador

        System.out.println("Vamos a acumular un numero entero 10 veces...");

        System.out.print("Ingrese el numero entero\n que quiere acumular: ");
        numero = entrada.nextInt();

        for (int contador = 1; contador <= 10;contador++ )
            acumulador= acumulador + numero;
            System.out.println("El total acumulado es: " + acumulador);

        System.out.println();

    }

}
```

6. Estructura repetitiva - DO WHILE

Repetición controlada por contador

La instrucción de repetición while, lleva a cabo una repetición controlada por contador, que requiere:

- 1. una variable de control (o contador de ciclo)
- 2. el valor inicial de la variable de control
- 3. el incremento con el que se modifica la variable de control cada vez que pasa por el ciclo (lo que también se conoce como cada iteración del ciclo)
- 4. la condición de continuación de ciclo, que determina si el ciclo debe continuar.

Como se puede observar, a diferencia del FOR, la instrucción WHILE, necesita más de una línea de código para el mismo proceso.

Instrucción DO WHILE

```
/**
 *
 * @author Diego Gomez Rivera
 * @version 1.0
 * Ejercicio Nro. 18
 * DO ... WHILE
 */

public class Ej18 {

    public static void main(String args[]) {
        // Uso del Do While
        {
            int x = 1;

            do
            {
                System.out.println("valor de x : " + x);
                x++;
            }
            while (x < 10);
        }
    }
}
```

Uso de WHILE

A diferencia de la sintaxis anterior, primero podemos asignar el valor de la variable que utilizaremos dentro del bucle WHILE. Eso nos da más versatilidad en algunos casos.

```
/**
 * @author Diego Gomez Rivera
 * @version 1.0
 * Ejercicio Nro. 18
 * DO ... WHILE
 */

public class Ej19 {

    public static void main(String args[]) {
        // Uso del Do While

        int x = 1;

        while (x < 10)
        {
            System.out.println("VALOR DE x : " + x);
            x++;
        }
    }
}
```


7. Estructuras de Control - Selección simple

Estructuras de Control Selectivas

Selección simple

Uso de IF y Operadores lógicos

```
/**
 *
 * @author Diego Gomez Rivera
 * @version 1.0
 * Ejercicio Nro. 15
 * uso de Scanner
 * Uso de operadores lógicos
 * Uso de IF
 */
import java.util.Scanner;

public class Ej15 {

    public static void main(String args[]) {
        // crea objeto Scanner para obtener la entrada de la ventana de comandos
        Scanner entrada = new Scanner( System.in );

        int numero1; // primer numero a comparar
        int numero2; // segundo numero a comparar

        System.out.print( "Escriba el primer entero: " ); // indicador
        numero1 = entrada.nextInt(); // lee el primer numero del usuario

        System.out.print( "Escriba el segundo entero: " ); // indicador
        numero2 = entrada.nextInt(); // lee el segundo numero del usuario

        if ( numero1 == numero2 )
            System.out.printf( "%d == %d\n", numero1, numero2 );

        if ( numero1 != numero2 )
            System.out.printf( "%d != %d\n", numero1, numero2 );

        if ( numero1 < numero2 )
            System.out.printf( "%d < %d\n", numero1, numero2 );

        if ( numero1 > numero2 )
            System.out.printf( "%d > %d\n", numero1, numero2 );

        if ( numero1 <= numero2 )
            System.out.printf( "%d <= %d\n", numero1, numero2 );

        if ( numero1 >= numero2 )
            System.out.printf( "%d >= %d\n", numero1, numero2 );

    }
}
```

Uso de IF y FOR

```
/**
 *
 * @author Diego Gomez Rivera
 * @version 1.0
 * Ejercicio Nro. 16
 * uso de Scanner
 * Uso de For
 * uso de IF
 * Acumuladores
```

```
*/
import java.util.Scanner;

public class Ej16 {

    public static void main(String args[]) {
        // crea objeto Scanner para obtener la entrada de la ventana de comandos
        Scanner entrada = new Scanner(System.in);

        int numero1; // numero a ingresar
        int aux, mayor;
        mayor = 0;
        aux = 0;

        for (int contador = 1;
            contador <= 10;
            contador++) {
            aux++;
            System.out.println("Escriba un numero entero: "); // indicador
            numero1 = entrada.nextInt(); // lee el primer numero del usuario
            if (numero1 >= mayor) {
                mayor = numero1;
            }
            System.out.printf("El numero mayor es %d\n", mayor);
            if (numero1 == 0) {
                break;
            }
        }

        System.out.printf("Ud ingreso  %d\n", aux);
        System.out.printf("El numero mayor es %d\n", mayor);
    }
}
```

8. Estructuras de Control - Selección doble

Estructuras de Control Selectivas

Selección doble

Uso de IF, ELSE, ENDIF, BREAK

```
/**
 *
 * @author Diego Gomez Rivera
 * @version 1.0
 * Ejercicio Nro. 17 uso de Scanner Uso de For uso de
 * IF y ELSE IF y ELSE IF anidados uso de BREAK y CONTINUE Acumuladores
 */
import java.util.Scanner;

public class Ej17 {

    public static void main(String args[]) {
        // crea objeto Scanner para obtener la entrada de la ventana de comandos
        Scanner entrada = new Scanner(System.in);

        int numero1; // numero a ingresar
        int aux, mayor, confirma;
        mayor = 0;
        aux = 0;
        confirma = 0;

        for (int contador = 1;
            contador <= 10;
            contador++) {
            aux++;
            System.out.println("Escriba un numero entero: "); // indicador
            numero1 = entrada.nextInt(); // lee el primer numero del usuario
            if (numero1 >= mayor) {
                mayor = numero1;
            }
            System.out.printf("El numero mayor es %d\n", mayor);
            if (numero1 == 0) {
                System.out.println("Desea SALIR? CONTINUAR = 1 / SALIR = 0 ");
                confirma = entrada.nextInt(); // lee un valor para la salida
                if (confirma == 0) {
                    break; // sale del bucle
                } else if (confirma == 1) {
                    continue; // continua
                } else {
                    System.out.println("Eligio otra opcion... entonces continuamos.. ");
                }
            }
        }
        System.out.printf("Ud ingreso  %d\n", aux);
        System.out.printf("El numero mayor es %d\n", mayor);

    }
}
```

9. Estructuras de Control - Selección múltiple

Estructuras de Control Selectivas

Selección múltiple

Uso de SWITCH CASE

Realiza distintas acciones, con base en los posibles valores de una expresión entera constante o de un string.

```
/**
 *
 * @author Diego Gomez Rivera
 * @version 1.0
 */
public class Ej22 {

    /**
     * @author Diego Gomez Rivera
     * Ejercicio Nro. 22 SWITCH con SCANNER
     * EVALUAR UNA NOTA
     */
    public static void main(String[] args) {
        // TODO code application logic here
        {
            char nota = 'C';

            switch (nota)
            {
                case 'A':
                    System.out.println("Excelente!");
                    break;
                case 'B':
                    System.out.println("Muy Bueno");
                    break;
                case 'C':
                    System.out.println("Bien");
                    break;
                case 'D':
                    System.out.println("Aprobado");
                case 'F':
                    System.out.println("Desaprobado");
                    break;
                default:
                    System.out.println("Nota invalida");
            }
            System.out.println("Su nota es " + nota);
        }

    }
}
```