

Unidad 02

Creación de mi primer programa



Módulo Programación
modalidad @vanza

UD02 – Creación de mi primer programa

1.- Introducción

2.- Las variables e identificadores

Una **variable** es una zona en la memoria del ordenador con un valor que puede ser almacenado para ser usado más tarde en el programa. Las variables vienen determinadas por:

- Un **nombre**, que permite al programa acceder al valor que contiene en memoria. Debe ser un *identificador* válido.
- Un **tipo de dato**, que especifica qué clase o tipo de información va a guardar
- Un **rango de valores** que puede admitir dicha variable.

2.1.- Identificadores

Un **identificador** en Java es una secuencia ilimitada sin espacios de letras y dígitos Unicode

UD02 – Creación de mi primer programa

2.2.- Convenios y reglas para nombrar variables

- Java **distingue las mayúsculas de las minúsculas**. Por ejemplo, Alumno y alumno son variables diferentes.
- **No se suelen utilizar** identificadores que comiencen con «\$» o «_», además el símbolo del dólar, por convenio, no se utiliza nunca.
- **No se puede utilizar** el valor booleano (true o false) ni el valor nulo (null).
- Los identificadores deben ser **lo más descriptivos posibles**. Convenios:
 - Identificador o nombre de **variable**. comenzar por letra minúscula, y si tienen más de una palabra se colocan juntas y el resto comenzando por mayúsculas. Ejemplo: numAlumnos, suma
 - Para los nombres de **constante**. En letras mayúsculas, separando las palabras con el guión bajo. Ejemplo: TAM_MAX, PI
 - Identificadores o nombre de una **clase**. Comienza por letra mayúscula. Ejemplos: String, MiTipo.
 - Identificadores o nombre de **función**. Comienza con letra minúscula. Ejemplo obtener_valor()

2.3.- Palabras reservadas:

son palabras utilizadas por el lenguaje de programación y, por tanto, no pueden utilizarse para crear identificadores.

UD02 – Creación de mi primer programa

2.4.-Tipos de Variables

Variables de tipos primitivos y variables referencia, según el tipo de información que contengan. En función de a qué grupo pertenezca la variable, tipos primitivos o tipos referenciados, podrá tomar unos valores u otros, y se podrán definir sobre ella unas operaciones u otras.

Variables y constantes, dependiendo de si su valor cambia o no durante la ejecución del programa.

- Variables**. Almacena datos durante la ejecución del programa, pueden estar formadas por cualquier tipo de dato primitivo o referencia. Su valor puede cambiar varias veces a lo largo de todo el programa.
- Constantes o variables finales**. Son aquellas variables cuyo valor no cambia a lo largo de todo el programa.

Variables miembro y variables locales, en función del lugar donde aparezcan en el programa.

- **Variables miembro**. Son las variables que se crean dentro de una clase, fuera de cualquier método. Pueden ser de tipos primitivos o referencias, variables o constantes.
- Variables locales**. Son las variables que se crean y usan dentro de un método o, en general, dentro de cualquier bloque de código. La variable deja de existir cuando la ejecución del bloque de código o el método finaliza.

UD02 – Creación de mi primer programa

2.4.1.- Tipos de variables. Crear variables y programas:

CREACIÓN DE MI PRIMER PROGRAMA



En netbeans->Crear proyecto; escribir el código; ejecutar

UD02 – Creación de mi primer programa

2.4.1.- Tipos de variables. Crear variables y programas:



UD02 – Creación de mi primer programa

2.4.1-Ejemplo:

```
public class ejemplovariables {  
    final double PI =3.1415926536; // PI es una constante  
    int x;                          // x es una variable miembro  
                                    // de clase ejemplovariables  
  
    int obtenerX(int x) {           // x es un parámetro  
        int valorantiguo = this.x; // valorantiguo es una variabe local  
        return valorantiguo;  
    }  
  
    // el método main comienza la ejecución de la aplicación  
    public static void main(String[] args) {  
        // aquí iría el código de nuestra aplicación  
  
    } // fin del método main  
  
} // fin de la clase ejemplovariables
```

UD02 – Creación de mi primer programa

3.- Tipos de datos

3.1.-Tipos de datos primitivos

Son datos sencillos que constituyen los tipos de información más habituales: números, caracteres y valores lógicos o booleanos.

`byte, short, int, long, float, double, char, boolean`

3.2.- Declaración e inicialización

Ej: `int numAlumnos = 15;`

3.3.- Tipos referenciados

A partir de los datos primitivos, se pueden construir otros tipos de datos, los referenciados, porque se utilizan para almacenar la dirección de los datos en la memoria del ordenador.

Ej: `int[] arrayDeEnteros;`

3.4.- Tipos enumerados

Una variable con un conjunto restringido de valores. Ej: `public enum Sexo(V, M);`

4.- Literales de los tipos primitivos.

UD02 – Creación de mi primer programa

3.4.- Tipos de datos. Ejemplos:

```
public class ejemplotipos {  
  
    // el método main inicia la ejecución de la aplicación  
    public static void main(String[] args) {  
        // Código de la aplicación  
        int i = 10;  
        double d = 3.14;  
        char c1 = 'a';  
        char c2 = 65;  
        boolean encontrado = true;  
        String msj = "Bienvenido a Java";  
  
        System.out.println("La variable i es de tipo entero y su valor es: " + i);  
        System.out.println("La variable f es de tipo double y su valor es: "+d);  
        System.out.println("La variable c1 es de tipo carácter y su valor es: "+c1);  
        System.out.println("La variable c2 es de tipo carácter y su valor es: "+c2);  
        System.out.println("La variable encontrado es de tipo booleano y su valor es: "+encontrado);  
        System.out.println("La variable msj es de tipo String y su valor es: " + msj);  
    } // fin del método main  
  
} // fin de la clase ejemplotipos
```

UD02 – Creación de mi primer programa

3.4.- Tipos de datos enumerados. Ejemplos:

```
10 public class tiposenumerados {
11     public enum Dias {Lunes, Martes, Miercoles, Jueves, Viernes, Sabado, Domingo};
12
13     public static void main(String[] args) {
14         // codigo de la aplicacion
15         Dias diaactual = Dias.Martes;
16         Dias diasiguiente = Dias.Miercoles;
17
18         System.out.print("Hoy es: ");
19         System.out.println(diaactual);
20         System.out.println("Mañana\nes\n"+diasiguiente);
21
22     } // fin main
23
24 } // fin tiposenumerados
```

UD02 – Creación de mi primer programa

5.- Operadores y expresiones.

5.1.- Operadores aritméticos: -, +, *, / y %

5.2.- Operadores de asignación: =, +=, -=, *=, /= y %=

5.3.- Operador condicional: ?:

5.4.- Operadores de relación: !=, >, <, >= y <=

5.5.- Operadores lógicos: !, &, |, ^, && y ||

5.6.- Operadores de bits: ~, &, |, ^, >>, << y >>>

5.7.- Trabajo con cadenas. `String cad1="uno"; ... cad1.length();`

5.8.- Precedencia de operadores.

Coinciden con las reglas de las expresiones del álgebra convencional.

UD02 – Creación de mi primer programa

5.1.- Operadores aritméticos. Ejemplo:

```
10 public class operadoresaritmeticos {
11     public static void main(String[] args) {
12         short x = 7;
13         int y = 5;
14         float f1 = 13.5f;
15         float f2 = 8f;
16         System.out.println("El valor de x es " + x + " y el valor de y es " + y);
17         System.out.println("El resultado de x + y es " + (x + y));
18         System.out.println("El resultado de x - y es " + (x - y));
19         System.out.printf("%s\n%s%s\n", "División entera:", "x / y = ", (x/y));
20         System.out.println("Resto de la división entera: x % y = " + (x % y));
21         System.out.printf("El valor de f1 es %f y el de f2 es %f\n", f1, f2);
22         System.out.println("El resultado de f1 / f2 es " + (f1 / f2));
23     } // fin de main
24 } // fin de la clase operadoresaritmeticos
```

UD02 – Creación de mi primer programa

5.2.- Operadores de asignación. Ejemplo:

```
public class operadoresasignacion {  
    // clase principal main que inicia la aplicación  
    public static void main(String[] args) {  
        int x;  
        int y;  
        x = 5; // operador asignación  
        y = 3; // operador asignación  
  
        //operadores de asignación combinados  
        System.out.printf("El valor de x es %d y el valor de y es %d\n", x,y);  
        x += y;  
        // podemos utilizar indistintamente printf o println  
        System.out.println(" Suma combinada: x += y " + " ..... x vale " + x);  
        x = 5;  
        x -= y;  
        System.out.println(" Resta combinada: x -= y " + " ..... x vale " + x);  
        x = 5;  
        x *= y;  
        System.out.println(" Producto combinado: x *= y " + " ..... x vale " + x);  
        x = 5;  
        x /= y;  
        System.out.println(" Division combinada: x /= y " + " ..... x vale " + x);  
        x = 5;  
        x %= y;  
        System.out.println(" Resto combinada: x %= y " + " ..... x vale " + x);  
    } // fin main  
} // fin operadoresasignacion
```

UD02 – Creación de mi primer programa

5.3 y 5.4.- Operadores relacional y condicional. Ejemplo:

```
public class ejemplosRelacionales {  
  
    public static void main( String args[] )    {  
  
        int x=2,y=3;  
        String cadena;  
        boolean resultado;  
  
        cadena=(x==y)?"iguales":"distintos";  
        System.out.printf("Los números %d y %d son %s\n",x,y,cadena);  
        resultado=(x!=y);  
        System.out.println("x != y // es " + resultado);  
        resultado=(x < y );  
        System.out.println("x < y // es " + resultado);  
        resultado=(x > y );  
        System.out.println("x > y // es " + resultado);  
        resultado=(x <= y );  
        System.out.println("x <= y // es " + resultado);  
        resultado=(x >= y );  
        System.out.println("x >= y // es " + resultado);  
    }  
}
```


UD02 – Creación de mi primer programa

5.5.- Operadores lógicos. Ejemplo:

```
public class operadoreslogicos {  
    public static void main(String[] args) {  
        // TODO code application logic here  
        System.out.println("OPERADORES LÓGICOS");  
  
        System.out.println("Negacion:\n ! false es : " + (! false));  
        System.out.println(" ! true es : " + (! true));  
  
        System.out.println("Operador AND (&):\n false & false es : " + (false & false));  
        System.out.println(" false & true es : " + (false & true));  
        System.out.println(" true & false es : " + (true & false));  
        System.out.println(" true & true es : " + (true & true));  
  
        System.out.println("Operador OR (|):\n false | false es : " + (false | false));  
        System.out.println(" false | true es : " + (false | true));  
        System.out.println(" true | false es : " + (true | false));  
        System.out.println(" true | true es : " + (true | true));  
  
    } // fin main  
} // fin operadoreslogicos
```

UD02 – Creación de mi primer programa

5.7.- Trabajo con cadenas. Ejemplo:

```
public class ejemplocadenas {
    public static void main(String[] args)
    {
        String cad1 = "CICLO DAM";
        String cad2 = "ciclo dam";

        System.out.printf( "La cadena cad1 es: %s y cad2 es: %s", cad1,cad2 );

        System.out.printf( "\nLongitud de cad1: %d", cad1.length() );

        // concatenación de cadenas (concat o bien operador +)
        System.out.printf( "\nConcatenación: %s", cad1.concat(cad2) );

        //comparación de cadenas
        System.out.printf("\ncad1.equals(cad2) es %b", cad1.equals(cad2) );
        System.out.printf("\ncad1.equalsIgnoreCase(cad2) es %b", cad1.equalsIgnoreCase(cad2) );
        System.out.printf("\ncad1.compareTo(cad2) es %d", cad1.compareTo(cad2) );

        //obtención de subcadenas
        System.out.printf("\ncad1.substring(0,5) es %s", cad1.substring(0,5) );

        //pasar a minúsculas
        System.out.printf("\ncad1.toLowerCase() es %s", cad1.toLowerCase() );

        System.out.println();
    } // fin main
} // fin ejemplocadenas
```


UD02 – Creación de mi primer programa

6.- Conversión de tipo.

- Conversiones automáticas.

Cuando asignamos el valor de un tipo de datos pequeño a un tipo de datos más grande.

```
public static void main(String[] args) {  
    int a=1;  
    double b=2.5;  
  
    b=a;  
    System.out.println(b);  
}
```

- Conversiones explícitas. **Casting.**

Al asignar un valor de tipo de dato más grande a un tipo de dato más pequeño.

```
public static void main(String[] args) {  
    int a=1;  
    double b=2.5;  
  
    a=(int)b;  
    System.out.println(a);  
}
```

UD02 – Creación de mi primer programa

7.- Comentarios.

- **Comentarios de una sola línea.** Utilizaremos el delimitador //
- **Comentarios de múltiples líneas.** /* al principio del párrafo y */ al final del mismo.
- **Comentarios Javadoc.** Utilizaremos los delimitadores /** y */