

Clase 2

PROGRAMACIÓN 1

Objetivos del tema

- **Declaración de variables y constantes**
- **Describir los tipos de datos primitivos en el lenguaje de programación Java y su formato de representación**
- **Cargar datos de entrada y obtener salidas desde consola**

Datos y variables

Dato

Describe aquello con lo que opera el programa.

Está asociado a una variable, y la variable es la que se manipula para procesar el dato.

Nombre de variable: es lo que identifica a la variable o permite referenciarla.

Tipo de variable: define los valores posibles que puede tomar, y las operaciones que se van a poder realizar con ella.

Todo nuevo identificador o variable que se emplee en un programa Java debe definirse previamente a su utilización.

Nombre de variables o identificadores

Un identificador comienza por una letra, un caracter de subrayado (_) o un caracter de peso(\$).

No hay límite máximo de caracteres para un nombre de variable.

En los identificadores de un programa en Java se distinguen las mayúsculas de las minúsculas.

- Por ejemplo, casa, CASA y Casa son tres variables diferentes.

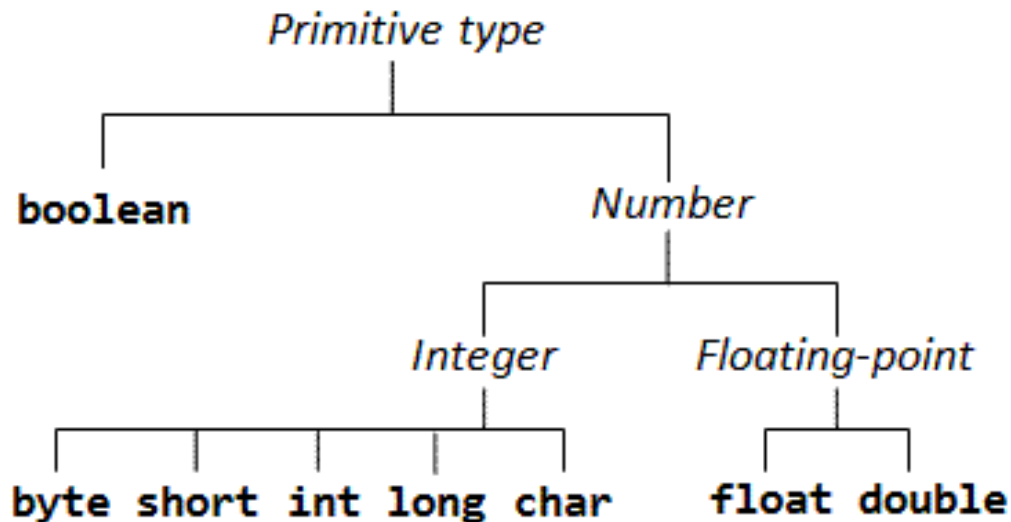
Variables: sirven para almacenar datos durante la ejecución del programa; el valor asociado puede cambiar varias veces durante la ejecución del programa.

Constantes o variables finales: también sirven para almacenar datos pero no puede modificarse posteriormente.

- Por ejemplo, el número PI y la aceleración de la gravedad G.

Tipos

- Un tipo es un “molde” que define los valores y operaciones posibles de las variables, y se dividen en dos grupos:
 - Tipos primitivos simples
 - Referencia: permiten almacenar un conjunto de elementos, pueden ser definidos por el usuario, etc.



Palabras reservadas

- Existe una serie de palabras reservadas que no pueden emplearse como identificadores o variables, y que son para otros usos.

<code>abstract</code>	<code>do</code>	<code>implements</code>	<code>protected</code>	<code>throw</code>
<code>boolean</code>	<code>double</code>	<code>import</code>	<code>public</code>	<code>throws</code>
<code>break</code>	<code>else</code>	<code>instanceof</code>	<code>rest</code>	<code>transient</code>
<code>byte</code>	<code>extends</code>	<code>int</code>	<code>return</code>	<code>true</code>
<code>case</code>	<code>false</code>	<code>interface</code>	<code>short</code>	<code>try</code>
<code>catch</code>	<code>final</code>	<code>long</code>	<code>static</code>	<code>void</code>
<code>char</code>	<code>finally</code>	<code>native</code>	<code>strictfp</code>	<code>volatile</code>
<code>class</code>	<code>float</code>	<code>new</code>	<code>super</code>	<code>while</code>
<code>const*</code>	<code>for</code>	<code>null</code>	<code>switch</code>	
<code>continue</code>	<code>goto*</code>	<code>package</code>	<code>synchronized</code>	
<code>default</code>	<code>if</code>	<code>private</code>	<code>this</code>	

Declaración de variables

- Una variable corresponde a un dato cuyo valor puede modificarse durante la ejecución.
- Toda variable ha de declararse antes de ser usada en el código de un programa en Java.
- En la declaración de una variable debe indicarse el identificador y el tipo de dato asociado.

```
tipo_de_dato ident_1, ident_2, . . . , ident_n;
```

Por ejemplo:

```
int n;  
double x, y;  
int z;
```

Ejemplo

```
/**
 * Variables
 * Cuando se conocen los valores iniciales de las
 * variables se los puede poner en la declaración
 */
public class Programa {
    public static void main(String[] args) {
        double altura;
        int edad = 20;
        boolean existe = true;
        float peso = (float) 56.5; //56.6 es double
        char character = 'c';
        altura = 1.87;
    }
}
```


Declaración de constantes

- Las variables finales en Java son similares a las constantes empleadas en otros lenguajes de programación.
- Una vez inicializada una variable final su valor no puede ser modificado

Por ejemplo:

```
final double PI = 3.1415926;
```

```
final double g = 9.81;
```

Ejemplo

```
/**  
 * Variables y constantes  
 */  
public class Programa {  
    public static void main(String[] args) {  
        final double g = 9.81;//constante  
        double altura;  
        int edad = 20;  
        boolean existe = true;  
        float peso = (float) 56.5; //56.6 es double  
        char character = 'c';  
        altura = 1.87;  
        peso = (float) 60.0;  
        g = 6.3;//GENERA UN ERROR  
    }  
}
```

Literales

- Un literal es una constante cuyo nombre es la representación escrita de su valor pre-asignado en Java.
- Las constantes literales booleanas son false y true.
- Las constantes literales de tipo carácter aparecen entre comillas simples.
- Letras mayúsculas ('A', 'B', 'C',...), letras minúsculas ('a', 'b', 'c',...), signos de puntuación (',' ';' ':' ...), dígitos, símbolos especiales ('#', '&', '%',...) y caracteres de control (\n,\t,...).

Literal	Valor
\b	Retroceso o backspace
\t	Tabulación
\n	Nueva línea (enter)
\f	Salto de página

Ejemplo

```
/**  
 * Literales  
 */  
public class Programa {  
    public static void main(String[] args) {  
        System.out.println ("Hola Mundo. \nEstoy  
programando.");  
    }  
}
```

Ejemplo

Cargar una variable de tipo entero desde teclado e imprimirla por pantalla

```
import java.io.BufferedReader;    //contiene operaciones de E/S
import java.io.InputStreamReader; //contiene operaciones de E/S
public class Program1 {
    public static void main(String[] args){
        int    entero;
        try {          //try define un bloque de manejo de posibles excepciones
            BufferedReader entrada = new BufferedReader(new
                InputStreamReader(System.in));
            System.out.println ("Ingrese int: ");
            entero = new Integer(entrada.readLine()); //new Integer crea un int
            System.out.println("int : "    + entero);
        }
        catch (Exception exc ) { //se ejecuta si ocurre algún error de lectura
            System.out.println( exc );
        }
    }
}
```

Entrada y salida desde consola

- Importación de librerías
 - Una librería es un conjunto de utilitarios que se acceden mediante sentencias para hacer operaciones predefinidas.

Por ejemplo:

```
import java.io.BufferedReader;  
import java.io.InputStreamReader;
```

- Carga desde teclado de variables de distinto tipo a través de la sentencia **System.in**
- Salida/impresión de variables a través de la sentencia **System.out**

Ejemplo

Agregar la librerías a utilizar y comenzar a definir las variables

```
import java.io.BufferedReader;           //contiene operaciones de E/S
import java.io.InputStreamReader;       //contiene operaciones de E/S
public class Program1 {
    public static void main(String[] args){
        int    entero;
    }
}
```

Una vez definidas las variables agregar el bloque try catch para manejo de errores de ingreso de datos

```
import java.io.BufferedReader;           //contiene operaciones de E/S
import java.io.InputStreamReader;       //contiene operaciones de E/S
public class Program1 {
    public static void main(String[] args){
        int    entero;
        try {                             //try define un bloque de manejo de posibles excepciones
        }
        catch (Exception exc ) {          //se ejecuta si ocurre algún error de lectura
            System.out.println( exc );
        }
    }
}
```

Ejemplo

Crear el buffer donde se almacenarán los datos por teclado

```
import java.io.BufferedReader;                //contiene operaciones de E/S
import java.io.InputStreamReader;            //contiene operaciones de E/S
public class Program1 {
    public static void main(String[] args){
        int    entero;
        try {                                //try define un bloque de manejo de posibles excepciones
            BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
        }
        catch (Exception exc ) {             //se ejecuta si ocurre algun error de lectura
            System.out.println( exc );
        }
    }
}
```

Leer desde teclado y convertirlo al tipo de variable a asignar

```
import java.io.BufferedReader;                //contiene operaciones de E/S
import java.io.InputStreamReader;            //contiene operaciones de E/S
public class Program1 {
    public static void main(String[] args){
        int    entero;
        try {                                //try define un bloque de manejo de posibles excepciones
            BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
            System.out.println ("Ingrese int: ");
            entero = new Integer(entrada.readLine());
        }
        catch (Exception exc ) {             //se ejecuta si ocurre algun error de lectura
            System.out.println( exc );
        }
    }
}
```


Ejemplo

Generar la salida del programa por consola

```
import java.io.BufferedReader;    //contiene operaciones de E/S
import java.io.InputStreamReader; //contiene operaciones de E/S
public class Programal {
    public static void main(String[] args){
        int    entero;
        try {           //try define un bloque de manejo de posibles excepciones
            BufferedReader entrada = new BufferedReader(new
                InputStreamReader(System.in));
            System.out.println ("Ingrese int: ");
            entero = new Integer(entrada.readLine()); //new Integer crea un int
            System.out.println("int : "    + entero);
        }
        catch (Exception exc ) { //se ejecuta si ocurre algún error de lectura
            System.out.println( exc );
        }
    }
}
```

Cargar y visualizar más variables

Tipo	Nombre
String	linea;
float	flotante;
double	doble_prec;
int	entero;
char	caracter;

Ejemplo de carga con más variables

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.util.Locale; //se usa para setear separadores decimales y de listas
public class Programa {
    public static void main(String[] args){
        String linea;
        float flotante;
        double doble_prec;
        int entero;
        char caracter;
        Locale.setDefault(new Locale("en", "US")); //define . como separador decimal
        try {
            BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
            System.out.println ("Ingrese string: ");
            linea = entrada.readLine();
            System.out.println ("Ingrese float: ");
            flotante = new Float(entrada.readLine());
            System.out.println ("Ingrese double: ");
            doble_prec = new Double(entrada.readLine());
            System.out.println ("Ingrese int: ");
            entero = new Integer(entrada.readLine());
            System.out.println ("Ingrese char: ");
            caracter = (char) (entrada.readLine().charAt(0));
            System.out.println("string : " + linea);
            System.out.println("float : " + flotante);
            System.out.println("double : " + doble_prec);
            System.out.println("int : " + entero);
            System.out.println("char : " + caracter);
        }
        catch (Exception exc ) {
            System.out.println( exc );
        }
    }
}
```

Resolución y salida

```
try {  
    BufferedReader entrada = new BufferedReader(new  
        InputStreamReader(System.in));  
    System.out.println ("Ingreso string: ");  
    linea = entrada.readLine();  
    System.out.println ("Ingreso float: ");  
    flotante = new Float(entrada.readLine());  
    System.out.println ("Ingreso double: ");  
    doble_prec = new Double(entrada.readLine());  
    System.out.println ("Ingreso int: ");  
    entero = new Integer(entrada.readLine());  
    System.out.println ("Ingreso char: ");  
    caracter = (char) (entrada.readLine().charAt(0));  
    System.out.println("string : " + linea);  
    System.out.println("float : " + flotante);  
    System.out.println("double : " + doble_prec);  
    System.out.println("int : " + entero);  
    System.out.println("char : " + caracter);  
}
```

Práctico

- Escribir un programa que permita ingresar 5 números de a uno por vez y que los muestre por pantalla en orden inverso:

Ej:

Ingreso		se debe mostrar
23		3
4		100
2	=>	2
100		4
3		23

- Pedir y luego informar por consola los valores necesarios para dibujar un círculo.
- Pedir y luego informar por consola los valores necesarios para dibujar un triángulo.

Práctico

- Escribir un programa que solicite Nombre, edad, altura, ocupación, y los imprima por pantalla.
- Escribir un programa que pida que se ingresen datos necesarios para emitir una factura por la compra de dos artículos de librería (tipo factura, número, nombre cliente, producto 1, importe 1, producto 2, importe 2, importe total). Como salida debe imprimir por pantalla la factura en un formato similar al del siguiente ejemplo:

Factura	C	N 221
Jorge Rodriguez		
Producto		importe
Pala		12.2
Alambre		20.0
Importe Total		32.2