

Estructura de datos y algoritmos

Rodrigo Alvarez

rodrigo.alvarez2@mail_udp.cl

Que es java?

Que es java?



Que es java?

- Desarrollado por Sun Microsystems
- Lanzado en 1995
- Actualmente propiedad de Oracle
- Open Source



Que es java?

- Lenguaje compilado a bytecode
(corre encima de la jvm)
- Fuertemente orientado a objetos
- Sintaxis similar a C++
- Multiplataforma



Domina todos lenguajes de programación estudiando en EDteam

ed.team/cursos



MAJOR COMPANIES THAT USE



Hello world

The screenshot shows a Java code editor interface. On the left, there is a code editor window titled "Main.java" containing the number "1". To the right of the code editor are several toolbars and buttons. The top toolbar includes a "+" button for creating new files, the title "Java Hello World", a "NEW" button, a "JAVA" dropdown menu, a pink "RUN ▶" button, and a three-dot menu icon. Below the toolbar, there are two input fields: "STDIN" and "Input for the program (Optional)". Underneath these fields is a large text area labeled "Output:" which contains the instruction "Click on RUN button to see the output".

Tipo dato	Tamaño	Descripción
byte	1 byte	números del -128 al 127
short	2 bytes	-32,768 al 32,767
int	4 bytes	-2,147,483,648 al 2,147,483,647
long	8 bytes	-9,223,372,036,854,775,808 al 9,223,372,036,854,775,807
float	4 bytes	números racionales, hasta 7 dígitos decimales
double	8 bytes	hasta 15 dígitos decimales
boolean	1 bit	true o false
char	2 bytes	un único carácter

Operadores aritméticos

Operator	Name	Description	Example
+	Addition	Adds together two values	x + y
-	Subtraction	Subtracts one value from another	x - y
*	Multiplication	Multiplies two values	x * y
/	Division	Divides one value by another	x / y
%	Modulus	Returns the division remainder	x % y
++	Increment	Increases the value of a variable by 1	++x
--	Decrement	Decreases the value of a variable by 1	--x

Operadores de asignación

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3

Operadores de comparación

Operator	Name	Example
<code>==</code>	Equal to	<code>x == y</code>
<code>!=</code>	Not equal	<code>x != y</code>
<code>></code>	Greater than	<code>x > y</code>
<code><</code>	Less than	<code>x < y</code>
<code>>=</code>	Greater than or equal to	<code>x >= y</code>
<code><=</code>	Less than or equal to	<code>x <= y</code>

Operadores lógicos

Operator	Name	Description	Example
&&	Logical and	true if both statements are true	<code>x < 5 && x < 10</code>
	Logical or	true if one of the statements is true	<code>x < 5 x < 4</code>
!	Logical not	Reverse the result	<code>!(x < 5 && x < 10)</code>

I/O: Input/Output en java

The screenshot shows a Java code editor interface. The top bar includes tabs for "lo.java" and "+", a file identifier "427p45t7h", and buttons for "NEW", "JAVA ▾", "RUN ▶", and a three-dot menu. The code area contains a single line of code: "1". To the right, under the "STDIN" section, the input "2 3" is shown. Below it, the "Output:" section displays the program's response:

```
Ingresá dos ints:  
el primer numero es: 2  
el segundo numero es: 3  
La suma de los numeros es: 5
```

Scanner class

La clase `Scanner` se utiliza para obtener la entrada del usuario, y es parte del paquete `java.util`.

Method	Description
<code>nextBoolean()</code>	Reads a <code>boolean</code> value from the user
<code>nextByte()</code>	Reads a <code>byte</code> value from the user
<code>nextDouble()</code>	Reads a <code>double</code> value from the user
<code>nextFloat()</code>	Reads a <code>float</code> value from the user
<code>nextInt()</code>	Reads a <code>int</code> value from the user
<code>nextLine()</code>	Reads a <code>String</code> value from the user
<code>nextLong()</code>	Reads a <code>long</code> value from the user

Scanner

The screenshot shows a Java code editor interface. The top bar includes tabs for "ScannerEx.java" and "427nv54rs", along with buttons for "NEW", "JAVA ▾", "RUN ▶", and a three-dot menu. The main area has two panes: "STDIN" on the left and "Output" on the right. In the "STDIN" pane, the user has typed "hola". In the "Output" pane, the program's response is displayed in a vertical list:

```
Ingrese una palabra  
La palabra ingresada es: hola  
Ingrese un numero  
El numero ingresado es: 3  
Ingrese un numero flotante  
El numero flotante ingresado es: 3.14
```

if, else, while, for

The screenshot shows a Java code editor interface. The left panel displays the code file `IfElseWhileForExample.java`. The right panel shows the output of running the code with input `17`.

`IfElseWhileForExample.java`

```
1
```

427nvkqmd

NEW JAVA ▾ RUN ▶ :

STDIN

17

Output:

the number is odd
numeros pares:
2
4
6
8
10
12
14
16

Strings

The screenshot shows a Java code editor interface. The top bar includes tabs for "StringExample.java" and "427nu2xbm", along with buttons for "NEW", "JAVA", "RUN", and a menu icon. The main area has two panes: "STDIN" on the left and "Output:" on the right. In the STDIN pane, the number "23" is entered. In the Output pane, the string "Hello world" is displayed, with each character appearing on a new line.

```
StringExample.java + 427nu2xbm NEW JAVA RUN :  
1  
23  
Output:  
H  
e  
l  
l  
o  
  
W  
o  
r  
l  
d
```

String methods

Arrays

The screenshot shows a Java code editor interface. The top bar includes tabs for "ArrayExample.java" and "+", a file identifier "427p4c9xn", and buttons for "NEW", "JAVA ▾", "RUN ▶", and a three-dot menu. The main area has a dark background. On the left, a vertical toolbar shows the number "1". The central workspace contains the following code:

```
1
```

The right side of the interface displays the execution environment. It shows "STDIN" and an optional input field containing "Input for the program (Optional)". Below that, it shows "Output:" followed by the value "4".

Métodos

```
[modificador de acceso] [tipo_retorno] nombre_funcion(tipo_1 parametro_1, tipo_2 parametro_2, ...){  
    // instrucciones  
    return valor;  
}
```

Métodos

Main.java Greeter.java + 427p5fbvj NEW JAVA ▾ RUN ► :

1

STDIN

Input for the program (Optional)

Output:

hi!

Clases

The screenshot shows a Java code editor interface. At the top, there is a header bar with the following elements from left to right: "ClassExample.java", a "+" button, "427pmavjh", "NEW", "JAVA ▾", "RUN ▶", and a three-dot menu icon. The main area is divided into two vertical sections. The left section is a code editor with a single line of code: "1". The right section contains two panels: "STDIN" which has a placeholder "Input for the program (Optional)", and "Output:" which displays the text "miauuuu".

Herencia

The screenshot shows a Java code editor interface. On the left, there is a code editor window titled "Car.java" with a line number "1". To the right of the code editor are several buttons: "+", "427pqr5m", "NEW", "JAVA ▾", "RUN ▶", and three vertical dots. The main area is divided into two sections: "STDIN" and "Output". The "STDIN" section contains the placeholder text "Input for the program (Optional)". The "Output" section displays the following text:
Output:
Tuut, tuut!
Ford 2023 4

Interfaces

Las interfaces en Java son una colección de métodos abstractos y constantes que pueden ser implementados por cualquier clase que las implemente.

Una interfaz define un conjunto de métodos y su firma, pero no proporciona implementaciones de los métodos.

Interfaces

The screenshot shows a Java code editor interface. On the left, there is a code editor window titled "Main.java" with a line number "1". To the right of the code editor are several tabs: "+", "427pruj44", "NEW", "JAVA ▾", "RUN ►", and "...". Below the tabs, there are two sections: "STDIN" and "Output". The "STDIN" section contains the placeholder text "Input for the program (Optional)". The "Output" section displays the text "The pig says: wee wee" followed by "Zzz".

Generics

The screenshot shows a Java code editor interface. On the left, there is a code editor window titled "Main.java" containing the following code:

```
1 public class Main {
```

On the right, there is a preview area. At the top of the preview area, it says "427psdcfw". Below that, there are tabs for "NEW", "JAVA ▾", "RUN ►", and a three-dot menu. The preview area has two sections: "STDIN" and "Output". Under "STDIN", it says "Input for the program (Optional)". Under "Output", it shows the number "3".

- `ArrayList`: Implementa un **TDA List** en array manipulando el tamaño de forma dinámica.
- `LinkedList`: Implementa un **TDA List** sobre una lista de nodos.
- `Queue`: Implementa un **TDA FIFO** en una lista de nodos.
- `Stack`: Implementa un **TDA LIFO** en una lista de nodos.
- `Map`: Implementa una estructura **TDA key-value** en un árbol o una tabla hash.
- `Set`: Implementa una estructura **TDA unique value** en un árbol o una tabla hash.

