

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 tabs: "+", "Java Hello World", "NEW", "JAVA ▾", "RUN ►", and "...". Below the tabs, there are two input fields: "STDIN" and "Input for the program (Optional)". Underneath these fields, the word "Output:" is displayed, followed by the instruction "Click on RUN button to see the output".

```
Main.java
+
Java Hello World
NEW JAVA ▾ RUN ► ...
1
STDIN
Input for the program ( Optional )
Output:
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 |

especificación tamaños

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 "+", a file name "427nv54rs", and buttons for "NEW", "JAVA ▾", "RUN ▶", and a three-dot menu. The main area has two panes: "STDIN" on the left containing the input "hola" and "Output" on the right displaying the program's response. The code in "ScannerEx.java" is as follows:

```
1 import java.util.Scanner;
2
3 public class ScannerEx {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         System.out.print("Ingrese una palabra");
7         String palabra = sc.nextLine();
8         System.out.println("La palabra ingresada es: " + palabra);
9         System.out.print("Ingrese un numero");
10        int numero = sc.nextInt();
11        System.out.println("El numero ingresado es: " + numero);
12        System.out.print("Ingrese un numero flotante");
13        double numeroFlotante = sc.nextDouble();
14        System.out.println("El numero flotante ingresado es: " + numeroFlotante);
15    }
16 }
```

The "Output" pane shows the following text:
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. On the left, there is a file tab labeled "IfElseWhileForExample.java". To its right is a plus sign icon. Further right is a code identifier "427nvkqmd". At the top right, there are several buttons: "NEW", "JAVA ▾", "RUN ▶", and a three-dot menu icon.

The main area consists of two panes. The left pane is titled "STDIN" and contains the input value "17". The right pane is titled "Output" and displays the following text:

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

Strings

The screenshot shows a Java code editor interface. On the left, there is a file list with "StringExample.java" selected. In the main area, the code is as follows:

```
1 public class StringExample {  
2     public static void main(String[] args) {  
3         String str = "Hello World";  
4         System.out.println(str);  
5     }  
6 }
```

On the right, the execution environment is shown. It includes tabs for "NEW", "JAVA ▾", "RUN ▶", and a three-dot menu. The "STDIN" section contains the input "23". The "Output:" section shows the result of the program's execution: "H", "e", "l", "l", "o", " ", "W", "o", "r", "l", "d".

String methods

Arrays

A screenshot of a Java code editor interface. The top bar shows the file name "ArrayExample.java", a plus sign for adding new files, the 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 main code area is empty. To the right, there are two sections: "STDIN" which contains the placeholder "Input for the program (Optional)", and "Output:" which displays the number "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 tabs: "+", "427pqr5m", "NEW", "JAVA ▾", "RUN ►", and three vertical dots. The "JAVA ▾" tab is currently selected. 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 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".

```
Main.java
+
427pruj44
NEW JAVA ▾ RUN ► ...
1
STDIN
Input for the program ( Optional )
Output:
The pig says: wee wee
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, it shows the file name "427psdcfw" and has buttons for "NEW", "JAVA ▾", "RUN ▶", and a three-dot menu. Below these buttons, the "STDIN" section contains the placeholder text "Input for the program (Optional)". The "Output:" section 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.

