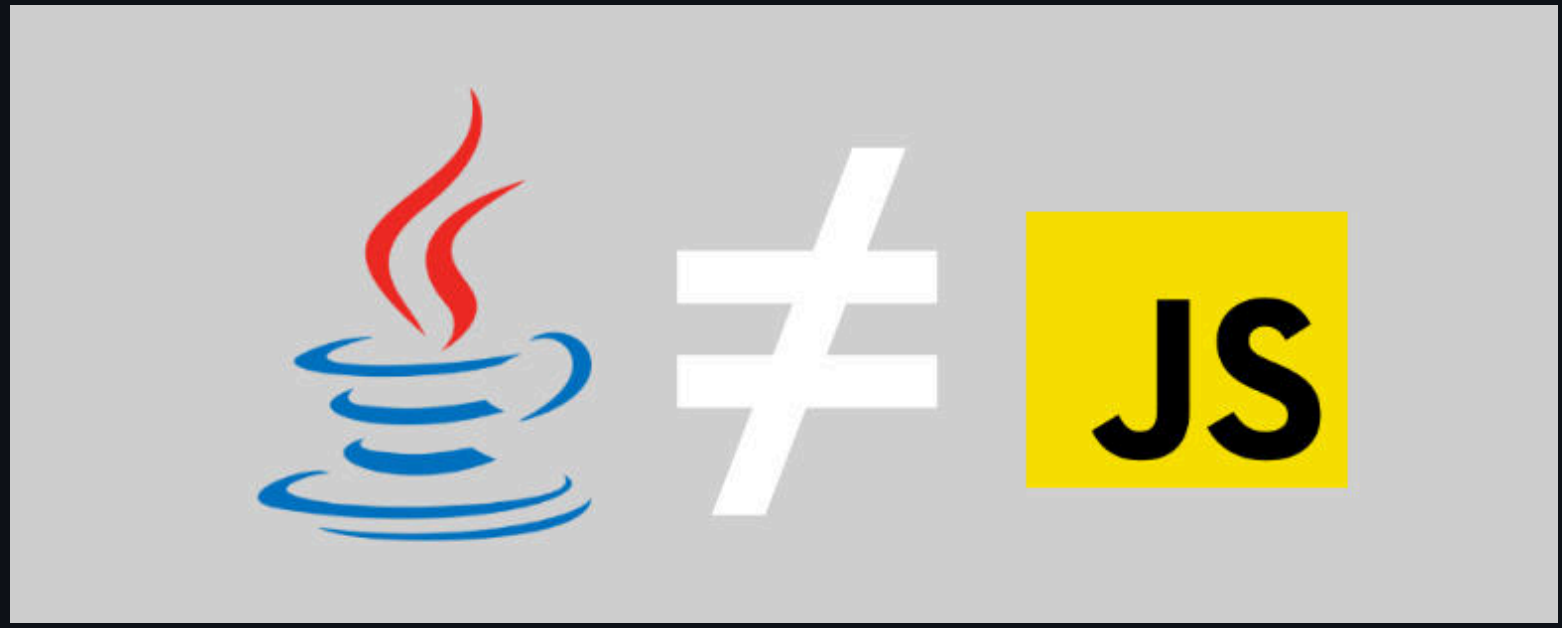
Estructura de datos y algoritmos

Rodrigo Alvarez

rodrigo.alvarez2@mail.udp.cl

Que es java?

Que es java?



Que es java?

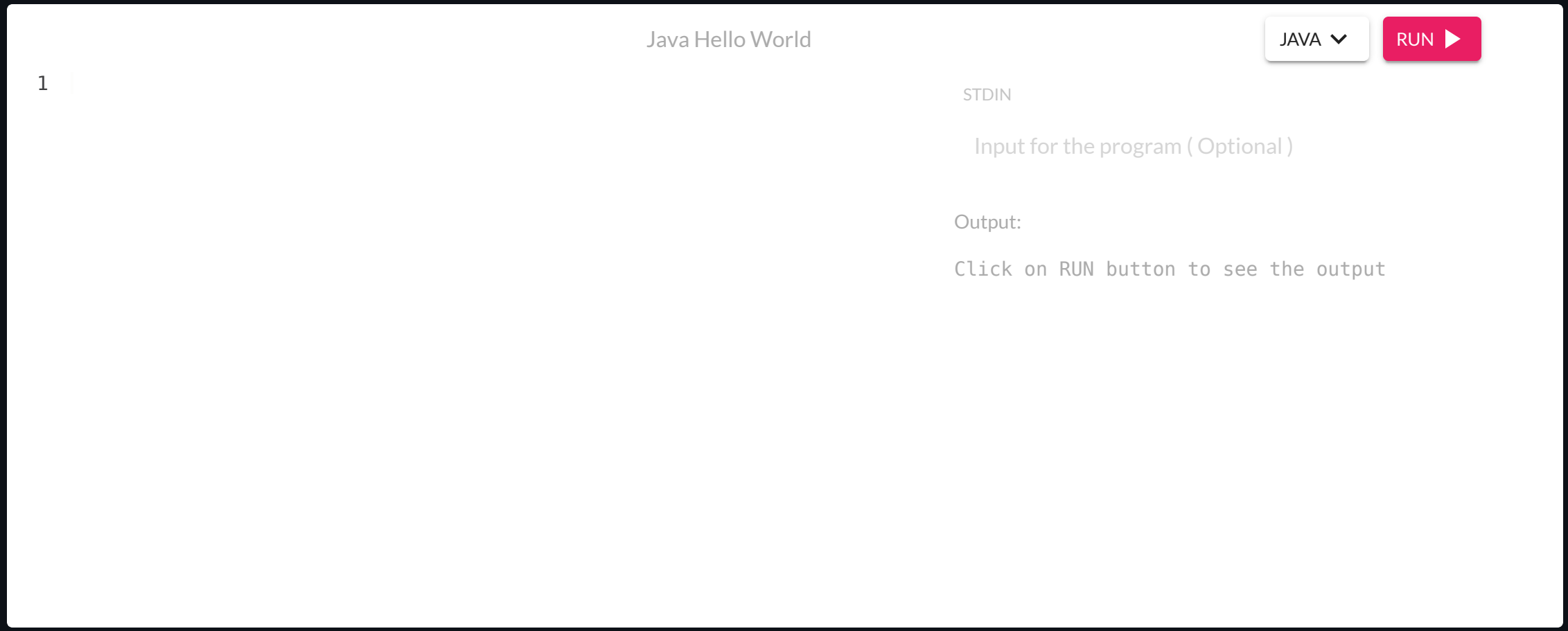
- Desarrollado por Sun Microsystems
- Lanzado en 1995
- Actualmente propiedad de Oracle
- Open Source

Que es java?

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


MAJOR COMPANIES
THAT USE
Java
LinkedIn
NETFLIX
Google
CapitalOne
amazon
slack
intel
NASA
ebay
android
Spotify
Square

Hello world


Java Hello World
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 digitos decimales
double	8 bytes	hasta 15 digitos 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
`==`	Equal to	`x == y`
`!=`	Not equal	`x != y`
`>`	Greater than	`x > y`
`<`	Less than	`x < y`
`>=`	Greater than or equal to	`x >= y`
`<=`	Less than or equal to	`x <= y`

Operadores lógicos

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

I/O: Input/Output en java

427p45t7h

JAVA

RUN

```
1
```

STDIN

```
2 3
```

Output:

```
Ingresa 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
`nextBoolean()`	Reads a `boolean` value from the user
`nextByte()`	Reads a `byte` value from the user
`nextDouble()`	Reads a `double` value from the user
`nextFloat()`	Reads a `float` value from the user
`nextInt()`	Reads a `int` value from the user
`nextLine()`	Reads a `String` value from the user
`nextLong()`	Reads a `long` value from the user

Scanner

427nv54rs

JAVA

RUN

```
1
```

STDIN

```
hola
3
```

Output:

```
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

427nvkqmd

JAVA

RUN

```
1
```

STDIN

```
17
```

Output:

```
the number is odd
numeros pares:
2
4
6
8
10
12
14
16
divisibles por 3:
3
6
9
```

Strings

String methods

Arrays

427p4c9xn

JAVA

RUN

```
1
```

STDIN

Input for the program (Optional)

Output:

4

Métodos

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

Métodos


427p5fbvj
JAVA
RUN
1
STDIN
Input for the program (Optional)
Output:
hi!

Clases

427pmavjh

JAVA

RUN

```
1
```

STDIN

Input for the program (Optional)

Output:

```
miauuuu
```

Herencia

427pqrv5m

JAVA

RUN

```
1
```

STDIN

Input for the program (Optional)

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

427pruj44

JAVA

RUN

```
1
```

STDIN

Input for the program (Optional)

Output:

```
The pig says: wee wee
Zzz
```

Generics

427psdcfw

JAVA

RUN

```
1
```

STDIN

Input for the program (Optional)

Output:

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.

- length of last word
- add binary

- Reto
- Por que hay tantos JDKs?