

UNIVERSIDAD NACIONAL DEL OESTE

PROGRAMACIÓN CON OBJETOS I



Profesor:

Mag. Ing. Pablo Pandolfo

Componentes del Lenguaje Java

TIPADOS VS NO TIPADOS

Los lenguajes estáticamente tipados requieren declarar las **variables con su tipo de dato**.

```
int a = 1;
String b = "hola";
bool c = true;
```



Más verboso, pero menos propenso a errores de sintaxis (el compilador los detecta).

Se puede saber qué tipo de dato retorna una función.

```
func saludar() string {
```



No se puede cambiar el tipo de dato después de declarada la variable.



Los lenguajes dinámicamente tipados declaran **variables sin necesidad de definir el tipo de dato** (el intérprete infiere el tipo).

```
let a = 1
let b = "hola"
let c = true
```



Código más legible y curva de aprendizaje más sencilla.

No sabes qué tipo de dato retorna una función.

```
def saludar():
```



Se puede cambiar el tipo de dato después de declarada la variable.





Componentes del Lenguaje Java

- ❑ Tipos:
 - ❑ Primitivos:
 - ❑ Numéricos enteros: **byte**, **short**, **int**, **long**
 - ❑ Numéricos reales: **float**, **double**
 - ❑ Lógico: **boolean**
 - ❑ Carácter: **char**
 - ❑ Objetos (Clases, Interfaces, Enums)



Componentes del Lenguaje Java



- Para los tipos primitivos existen clases “wrappers”:

- **int** Integer

- **boolean** Boolean

- **char** Character

- ...

- ¿Para qué utilizar las clases wrappers?

- Java no es un lenguaje orientado a objetos “puro”.

TIPOS DE DATOS EN JAVA

TIPOS PRIMITIVOS (sin métodos; no son objetos; no necesitan una invocación para ser creados)

NOMBRE	TIPO	OCUPA	RANGO APROXIMADO
byte	Entero	1 byte	-128 a 127
short	Entero	2 bytes	-32768 a 32767
int	Entero	4 bytes	2^{31}
long	Entero	8 bytes	Muy grande
float	Decimal simple	4 bytes	Muy grande
double	Decimal doble	8 bytes	Muy grande
char	Carácter simple	2 bytes	---
boolean	Valor true o false	1 byte	---

TIPOS OBJETO (con métodos, necesitan una invocación para ser creados)

Tipos de la biblioteca estándar de Java	String (cadenas de texto) Muchos otros (p.ej. Scanner, TreeSet, ArrayList...)
Tipos definidos por el programador/usuario	Cualquiera que se nos ocurra, por ejemplo Taxi, Autobus, Tranvia
arrays	Serie de elementos o formación tipo vector o matriz. Lo consideraremos un objeto especial que carece de métodos.
Tipos envoltorio o wrapper (Equivalentes a los tipos primitivos pero como objetos.)	Byte Short Integer Long Float Double Character Boolean

Componentes del Lenguaje Java

- ❑ Clase String:
 - ❑ Es una secuencia de caracteres.
 - ❑ No es un tipo primitivo.
 - ❑ Los String son instancias de la clase **java.lang.String**.
 - ❑ El compilador trata a los String como si fuesen tipos del lenguaje.
 - ❑ La clase tiene varios métodos para trabajar con ellos.
 - ❑ Son inmutables.
 - ❑ Como crear uno:

String saludo = "Hola";

String otroSaludo = new String("Como andás?");



Componentes del Lenguaje Java

□ Clase String:

□ Comportamiento:

saludo.length();
saludo.charAt(0);
saludo.substring(0,2);
saludo.compareTo("adios");
saludo.equals("Hol");
saludo.split("o");
saludo.trim();

saludo.indexOf("a");
saludo.indexOf("a", 2);
saludo.toLowerCase();
saludo.toUpperCase();
saludo.equalsIgnoreCase("h");
saludo.endsWith("la");

Componentes del Lenguaje Java

Operadores:

Asignación	<code>i = i + 1;</code> <code>j += k;</code>
Relacionales	<code>2 >= p;</code> <code>a == 5;</code> <code>b != 8;</code>
Aritméticos	<code>i + 4 * f;</code> <code>j - (8 / b);</code>
Lógicos	<code>a && b;</code> <code>d c;</code> <code>!a;</code>
Condicional	<code>(x < y) ? x : y;</code>
Otros	<code>[]</code> <code>.</code> <code>()</code> (refundición) new instanceof

Componentes del Lenguaje Java

□ Arreglos:

- Todo arreglo es de un tipo o una clase.
- Se los instancia con el operador **new**
- Son “zero-based”
- Ejemplos:

```
char[] caracteres;  
String[] palabras = new String[10];  
float[] arreglo = {2, 4, 6};  
int[][] matriz = {{1, 7}, {3, 4}};  
caracteres = new char[]{ 'a', 'l' };  
palabras[2] = "chango";
```

Componentes del Lenguaje Java

- Arreglos:

- Operaciones con arreglos.

- Es necesario importar la clase `java.util.Arrays`.

- Ejemplos:

QuickSort
 $O(n \log n)$

```
int [] arreglo = new int [10000];  
...  
Arrays.sort(arreglo);  
Arrays.binarySearch(arreglo, 3);  
Arrays.fill(arreglo, 3);
```

Componentes del Lenguaje Java

□ Estructuras de control:

- **if:**
if (condición) {
 sentencias;
}
- **while:**
while (condición) {
 sentencias;
}
- **for:**
for (int i=0; i < 8; i++) {
 sentencias;
}
for (tipo elemento: colección)
{
 sentencias; // por c/
 elemento
}
- **if else:**
if (condición) {
 sentencias;
} else {
 sentencias;
}
- **switch:**
switch (i) {
 case 1:{sentencias; break;
 }
 case 2:{sentencias; break;
 }
 case 3:{sentencias; break;
 }
 [default: sentencias;
 }

Componentes del Lenguaje Java



```
boolean java = true;  
if(java == true){  
    System.out.println("Planeta Java");  
}
```

```
boolean java = true;  
if(java){  
    System.out.println("Planeta Java");  
}
```


Componentes del Lenguaje Java



```
public class JavaEnGeneral {  
  
    public static void main(String[] args) {  
        int i = (new Random()).nextInt(3) + 1;  
        if (i == 1) {  
            System.out.println("Java");  
        } else if (i == 2) {  
            System.out.println("En");  
        } else if (i == 3) {  
            System.out.println("General");  
        }  
    }  
}
```

```
public class JavaEnGeneral {  
  
    public static void main(String[] args) {  
        int i = (new Random()).nextInt(3) + 1;  
        switch (i) {  
            case 1:  
                System.out.println("Java");  
                break;  
            case 2:  
                System.out.println("En");  
                break;  
            case 3:  
                System.out.println("General");  
                break;  
        }  
    }  
}
```

Componentes del Lenguaje Java

- Comentarios:

- No aparecen en el programa ejecutable.

- Existen tres tipos:

- Por línea: //

- Bloque de código: /* */

- JavaDoc: /** */ Genera automáticamente la documentación a través de marcadores:

- @param

- @return

- @throws

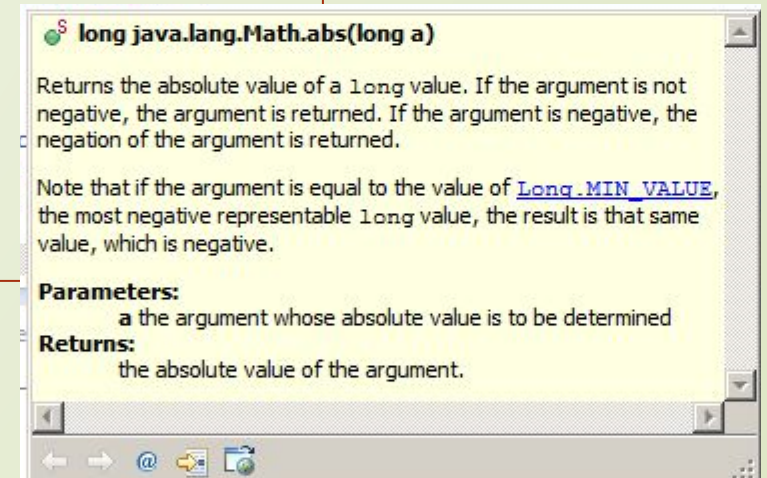
- @author

- @version

Componentes del Lenguaje Java

□ Ejemplo javadoc, en el método Math.abs

```
/**
 * Returns the absolute value of a {@code long} value.
 * If the argument is not negative, the argument is returned.
 * If the argument is negative, the negation of the argument is returned.
 *
 * <p>Note that if the argument is equal to the value of
 * {@link Long#MIN_VALUE}, the most negative representable
 * {@code long} value, the result is that same value, which
 * is negative.
 *
 * @param a the argument whose absolute value is to be determined
 * @return the absolute value of the argument.
 */
public static long abs(long a) {
    return (a < 0) ? -a : a;
}
```



Componentes del Lenguaje Java

- Entrada de datos por teclado:

```
Scanner in = new Scanner(System.in);  
in.nextLine();           // lee una línea de entrada  
in.next();               // lee una sola palabra  
in.nextInt();            // lee un entero  
in.nextDouble();        // lee número de coma flotante  
in.hasNext();            // si hay o no otra palabra  
in.hasNextInt();         // si hay o no otro entero  
in.hasNextDouble();      // si hay o no otro número de coma flotante
```

Componentes del Lenguaje Java

□ Salida de datos a consola:

```
System.out.print("Hola mundo");  
System.out.println("Hola mundo");
```

```
Double x = 10000.0/3.0;  
System.out.println(x);           //3333.3333333  
System.out.printf("%8.2f", x); //3333.33  
System.out.printf("%.2f", x); //3,333.33
```

```
String.format("Hola, %s. El año que viene tendrás %d", nombre, edad);
```

Ejercicios:

La prueba “Fizz-Buzz”

SE TRATA DE UN EJERCICIO USADO EN ENTREVISTAS LABORALES PARA EL PUESTO DE DESARROLLADOR, QUE APUNTA A FILTRAR A QUIENES NO TIENEN LA CAPACIDAD MÍNIMA NECESARIA PARA RESOLVER PROBLEMAS. UNA VEZ RESUELTO ESTE SENCILLO EJERCICIO, COMIENZA “LA VERDADERA ENTREVISTA”.

La
prueba
es la
siguiente:

Imprimir los números del 1 al 100. Pero, en lugar de los números múltiplos de 3 se debe imprimir “Fizz” y en lugar de los múltiplos de 5 se debe imprimir “Buzz”. En lugar de los números que sean múltiplos tanto de 3 como de 5, imprimir “FizzBuzz”.

NORMALMENTE SE DA A ELEGIR EL LENGUAJE, PERMITIENDO INCLUSO EL PSEUDOCÓDIGO.

Ejercicios:

PROBLEMA DE PROGRAMACIÓN

Intercambiar el valor de dos variables de tipo entero sin utilizar una tercera variable.

Elabore un script que te pida un número del 1 al 10 y te dé como resultado la cruz de números inferiores

Si ingresas:2 1 121 1	Si ingresas 3 1 2 12321 2 1	Si ingresas 4 1 2 3 1234321 3 2 1
Si ingresas 5 1 2 3 4 123454321 4 3 2 1		

PREGUNTA DE ENTREVISTA LABORAL · DESARROLLADOR

Escribir una función para invertir un string. Puede estar escrita en cualquier lenguaje, pero no es posible usar una biblioteca que lo haga automáticamente.



Encriptar un mensaje usando el método de "la cifra del César", que consiste en correr cada letra –considerando la posición de cada una en el alfabeto– una determinada cantidad de lugares. Ejemplo: si el corrimiento es de 2 lugares, la palabra "HOLA" se transforma en "JQNC".

Si el alfabeto termina antes de poder correr la cantidad de lugares necesarios, se vuelve a comenzar desde la letra 'a'.

Dar una implementación en Python de la función `primeros_N` que reciba un entero positivo, `n`, como único argumento. Luego la función debe imprimir los primeros `n` cuadrados perfectos que no sean números pares. Ejemplo: si `n` es 2, debería imprimir los cuadrados perfectos 1 y 9.

En McDonald's se pueden comprar patitas de pollo en paquetes de 6, 9 o 20 unidades. Escribir una función en Python que reciba un parámetro entero, *num*, y decida si es o no posible comprar *num* cantidad de patitas.

```
public static void main(String[] args) {
    Scanner leer=new Scanner(System.in);
    String genero;
    double talla,promediov,promediom,totalv,totalm;
    int xv=0,xm=0;totalv=0;totalm=0;
    do {
        System.out.print("Ingrese genero varon-mujer:");
        genero=leer.next().toUpperCase();
        System.out.print("Ingresar talla:");
        talla=leer.nextDouble();
        if(genero.equalsIgnoreCase("Varon")==true){
            totalv=totalv+talla;    xv=xv+1;
            if(talla==0) xv=xv-1;
        }
        if(genero.equalsIgnoreCase("Mujer")==true){
            totalm=totalm+talla;    xm=xm+1;
            if(talla==0) xm=xm-1;
        }
        totalv=totalv+talla;
        xv=xv+1; //TALLA FUERA DE RANGO ES 0 PARA FINALIZAR
    }
    while(talla!=0);
    promediov=totalv/xv;
    promediom=totalm/xm;
    System.out.println("promedio varon="+promediov);
    System.out.println("promedio mujer="+promediom);
}
```



```
1. class Main {
2.     static String metodo(String[] arreglo) {
3.         String mas_l = null;
4.         int long_mas_l = 0;
5.         for (String cadena : arreglo) {
6.             if (cadena.length() > long_mas_l) {
7.                 long_mas_l = cadena.length();
8.                 mas_l = cadena;
9.             }
10.        }
11.        return mas_l;
12.    }
13.    public static void main(String[] args) {
14.        String[] a = {"mayo", "abril", "julio", "junio"};
15.        System.out.println(metodo(a));
16.    }
17. }
```

¿Qué salida produce este programa?

Ejercicios:

Pseudocódigo

Algoritmo testParImpar

 Escribir "Indique un número: "

 Leer numero

 Si numero % 2 = 0 Entonces

 Escribir numero, " es par"

 SiNo

 Escribir numero, " no es par"

 Fin Si

FinAlgoritmo

```
1 Algoritmo Primo
```

```
2     Definir x,n como entero
```

```
3     Escribir "Dame un número"
```

```
4     leer n
```

```
5     x=1
```

```
6     mientras x <= n hacer
```

```
7         si n mod x == 0 entonces
```

```
8             contador = contador + 1
```

```
9             FinSi
```

```
10            x=x+1
```

```
1 FinMientras
```

```
2     si contador == 2 entonces
```

```
3         escribir "El numero, ",n, " es primo "
```

```
4     SiNo
```

```
5         escribir "El numero, ",n, " no es primo "
```

```
6     FinSi
```

```
7 FinAlgoritmo
```

```
8
```

```
9
```

Ejercicios:

ALGORITMOS DE BÚSQUEDA

La búsqueda secuencial en un arreglo desordenado de N elementos recorre el arreglo, comparando elemento a elemento. En el peor caso (no encontrar el elemento) se habrán hecho N comparaciones.

ALGORITMO EN C

Los parámetros corresponden al arreglo (a), el valor buscado (v), el índice donde comenzará la búsqueda (l) y el índice hasta donde se debe buscar (r).

```
int buscar(int a[], int v, int l, int r) {
    int i;
    for (i = l; i <= r; i++)
        if (v == a[i])
            return i;
    return -1;
}
```

La búsqueda binaria depende de que el arreglo esté ordenado. Sucesivamente compara el valor buscado, sólo en una mitad del arreglo. En el peor caso se realizan $\lceil \log N \rceil + 1$ comparaciones.

ALGORITMO EN C

Cuando el arreglo está ordenado, se puede terminar la búsqueda al encontrar un valor mayor que el buscado. El arreglo se divide en dos sucesivamente, descartando la mitad donde se sabe que no se encontrará el valor buscado.

```
int buscar(int a[], int v, int l, int r) {
    while (r >= l) {
        int m = (l+r)/2;
        if (v == a[m])
            return m;
        if (v < a[m])
            r = m-1;
        else
            l = m+1;
    }
    return -1;
}
```

Algoritmo Ordenamiento_burbuja

```
Definir a,b,c,x,temp, vector Como Entero
Dimension vector(5)
para b = 1 hasta 5 hacer
    Escribir "Dame el numero ",b
    leer vector(b)
FinPara
```

```
para x = 2 hasta 5 hacer
    para a=1 hasta 4 hacer
        si vector(a) > vector(a+1)
            temp = vector(a)
            vector(a) = vector(a+1)
            vector(a+1) = temp
        FinSi
    FinPara
FinPara
```

```
para c = 1 hasta 5 Hacer
    Escribir vector(c)
FinPara
```

FinAlgoritmo