

Lenguaje Java

conceptos básicos



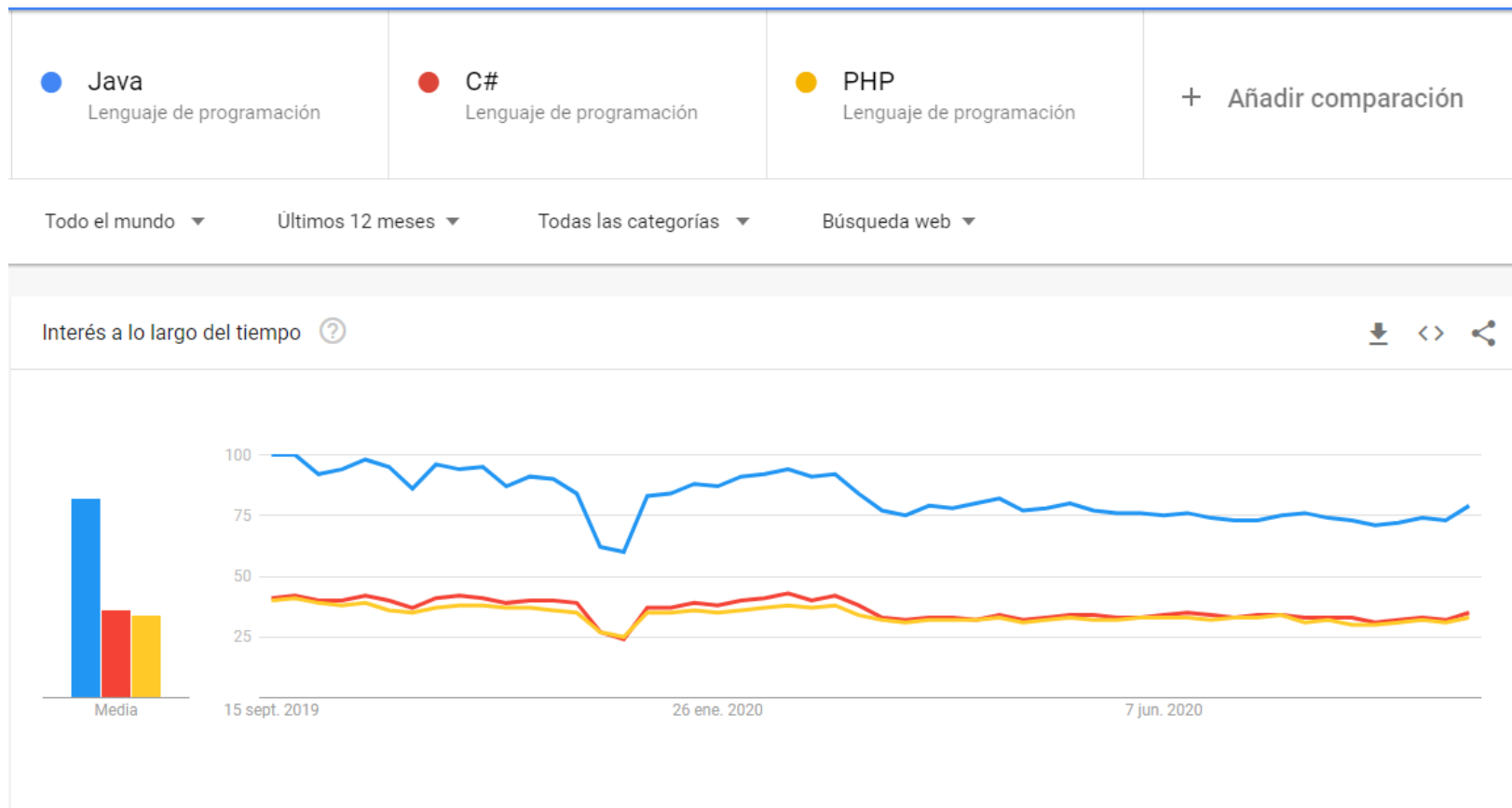
¿Cómo programamos?

- JDK y JRE
- <https://www.oracle.com/technetwork/es/java/javase/downloads/index.html>
- IDE (Netbeans, Eclipse)
- Netbeans
 - Apache NetBeans 12
(<https://netbeans.apache.org/download/index.html>)
 - Oracle NetBeans 8.2 with JDK 8u111(
<https://www.oracle.com/technetwork/es/java/javase/downloads/jdk-netbeans-jsp-3413139-esa.html>)

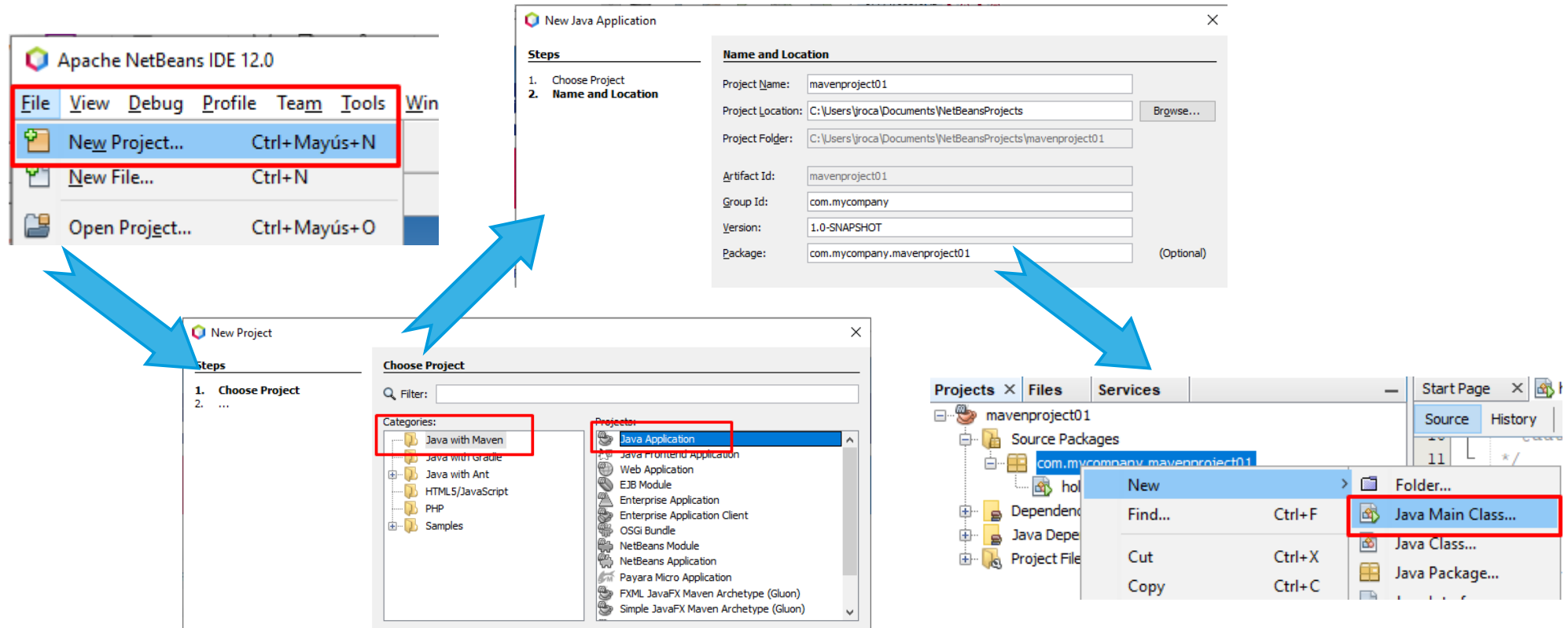
¿Por qué aprender Java?

- Lenguaje java fuertemente tipado
- Lenguaje Orientado a Objetos
- Está en todas partes (S.O., distintos dispositivos, etc)
- Sigue siendo popular.
- Ejemplo: <https://trends.google.es/trends/explore?date=today%205-y&q=%2Fm%2F07sbkfb,%2Fm%2F07657k,%2Fm%2F060kv>

Comparativa Java



Usando Netbeans



Variables: datos primitivos

Tipo	Tamaño	Valor min.	Valor máx.
byte	8 bits	-128	127
short	16 bits	-32768	32767
int	32 bits	-2147483648	2147483647
long	64 bits	-9223372036854775808	9223372036854775809
float	32 bits	-3.402823E+038	3.402823E+038
double	64 bits	-1.79769313486232e308	-1.79769313486232e308
char	16 bits	'\u0000'	'\uffff'
boolean	1 bit	0	1

Ejemplos:

- Ejemplos de valores primitivos en Java
- Byte, short, int y long

```
// valores primitivos:
byte mordida = 127;
System.out.println("\nValor de byte mordida: " + mordida);

short corto;
corto = 3200;
System.out.println("Valor de short corto: " + corto);

int entero;
entero = 2147483647;
System.out.println("Valor de int entero: " + entero);

long largo;
largo = 2147483647;
System.out.println("Valor de long largo: " + largo);
```

Ejemplos:

- Ejemplos de valores primitivos en Java
- Float, double, char y boolean

[illegible]

De tipo primitivo a envoltorio

- Cambiando nomenclatura de algunos tipos primitivos.
- Permite acceso a funciones internas pero ocupan más.
- `byte` → `Byte`
- `short` → `Short`
- `int` → `Integer`
- `long` → `Long`
- `float` → `Float`
- `double` → `Double`
- `boolean` → `Boolean`

```
Byte mordida = 12;  
System.out.println
```

`mordida.`

```
short corto;  
corto = 3200;  
System.out.println  
  
int entero;
```

• <code>byteValue()</code>	<code>byte</code>
• <code>compareTo(Byte b)</code>	<code>int</code>
• <code>doubleValue()</code>	<code>double</code>
• <code>equals(Object o)</code>	<code>boolean</code>
• <code>floatValue()</code>	<code>float</code>

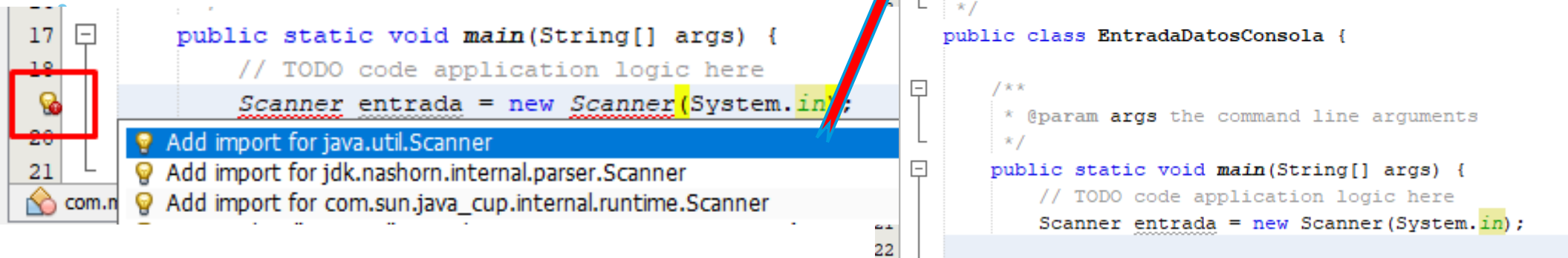
Constantes

- Se define con la palabra clave *final*
- *Mantener el valor de una variable*

```
final float pi = 3.1416f;  
System.out.println("Valor de constante pi: " + pi );
```

Entrada por consola

- Se define con la palabra clave `Scanner`
- Permite introducir información al programa por consola.
- Necesaria la clase `java.util.Scanner`.
- Las clases dentro de mi mismo paquete no es necesario importarlas.



Entrada por consola (I)

- Usando el Scanner creado podemos leer valores.

```
//Definimos la variable Scanner
Scanner entrada = new Scanner(System.in);

int entero;
System.out.println("Escribe un entero: ");
entero=entrada.nextInt();

System.out.println("Escribe un decimal: ");
float valor;
valor=entrada.nextFloat();
System.out.println("Los números son : "+entero+" y "+valor);
```

Entrada por consola II

- Usando el Scanner creado podemos leer valores : Cadenas.
- Variable String.

```
String cadena;  
System.out.println("Escribe una cadena: ");  
cadena=entrada.nextLine();  
System.out.println("La cadena escrita es: "+cadena);
```

Operadores matemáticos

- Suma ($\text{var1} + \text{var2}$)
- Resta ($\text{var1} - \text{var2}$)
- Multiplicación ($\text{var1} * \text{var2}$)
- División ($\text{var1}/\text{var2}$)
- Residuo ($\text{var1}\%\text{var2}$)

```
Scanner entrada = new Scanner(System.in);

float num1, num2, suma, resta, mul, div, residuo;
System.out.println("Introduce 2 números:");
num1=entrada.nextFloat();
num2=entrada.nextFloat();
suma=num1+num2;
resta=num1-num2;
mul=num1*num2;
div=num1/num2;
residuo=num1%num2;
System.out.println("Suma: "+suma);
System.out.println("Resta: "+resta);
System.out.println("Multiplicación: "+mul);
System.out.println("División: "+div);
System.out.println("Residuo: "+residuo);
```

Comentarios en código

- Comentario de una sola línea
- //
- Comentario de varias líneas
- Todo lo comprendido entre /* código */

```
//Definimos la variable Scanner
//Scanner entrada = new Scanner(System.in);
/*
int entero;
System.out.println("Escribe un entero: ");
entero=entrada.nextInt();

System.out.println("Escribe un decimal: ");
float valor;
valor=entrada.nextFloat();
System.out.println("Los números son : "+entero+" y "
*/
```

Uso compacto de operadores

- Incremento var++;
- Decremento var--;
- Uso compacto de operadores
- var-=4; (var=var-4)
- var+=5; (var=var+5)
- var*=3; (var=var*3)

```
float num=5;  
//num=num+1;  
//num+=5;  
//num++;  
//num-=4;  
//num--;  
//num*=5;  
//num/=5;  
num%=5; I  
System.out.println("Res="+num);  
1
```


Condicional y operadores

- If(condicion)
 {código}
- if(condición)
 {código}
- else
 {código2}
- Op. de comparación
- == (iguales)
- != (distintos)
- > (mayor que)
- < (menor que)
- >= (mayor o igual)

```
Scanner entra=new Scanner(System.in);
int numUser,numSis;
numSis=(int)(Math.random()*10);
System.out.print("Ingresa un numero mayor a "+numSis+": ");
numUser=entra.nextInt();
if(numUser > numSis){
    System.out.println("Muy bien");
}
else{
    System.out.println("Muy mal");
}
```

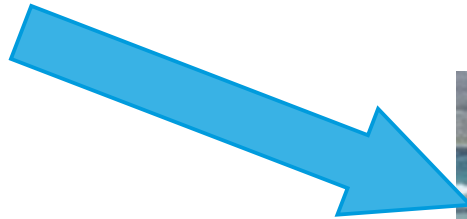
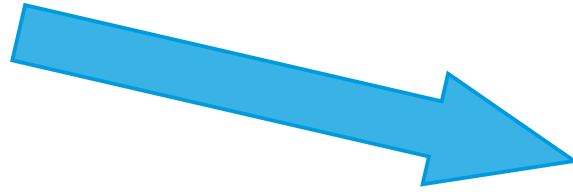
Bucles for y foreach

- for(inicio;final;incremento)
 {código}
- for(String variable:listado)
 {código}

```
//  
String [] perros={"caniche","bulldog","husky","dálmata"};  
for(int i=0;i<perros.length;i++){  
    System.out.println(perros[i]);  
}  
System.out.println("\n");  
for(String perro:perros){System.out.println(perro);}
```

POO

- Clase (Vehículo)
- Atributos (características)
- Marca
- Modelo
- Color
- Métodos (acciones)
- Enciende
- Acelera
- Frena
- Crear un objeto (instanciar la clase o crear una instancia)



POO

•Clase (Vehículo)

```
public class Vehiculo {  
    String marca;  
    String modelo;  
    String color;  
  
    public void enciende() {  
        System.out.println("Vehiculo encendido, run, run");  
    }  
    public void acelera() {  
        System.out.println("Acelerando, velocímetro a 80 km/h");  
    }  
    public void frena() {  
        System.out.println("Frenando, velocímetro a 0 km/h");  
    }  
}
```

```
public static void main(String[] args) {  
  
    Vehiculo testarrossa = new Vehiculo();  
    testarrossa.color = "rojo";  
    testarrossa.marca = "ferrari";  
    testarrossa.modelo = "testarrossa";  
  
    testarrossa.enciende();  
    testarrossa.acelera();  
    testarrossa.frena();  
  
    System.out.println("características : "+testarrossa.color);  
}
```

•Instanciar el monovolumen para la Clase Vehículo

POO

- Clase calculadora. Métodos que devuelven valores.
- Crear nueva clase Calculadora
- Crear método suma
- En la Clase main crear instancia de Calculadora.
- Pedir 2 datos y realizar el cálculo.

```
public class Calculadora {  
    public int suma(int a, int b) {  
        int resultado = 0;  
        if(a>0 && b>0)  
            resultado=a+b;  
        return resultado;  
    }  
}
```

- Crear los métodos para
- Resta
- Multiplicación
- División

EJEMPLO CALCULADORA

• Clase Calculadora

Llamada a la clase

```
public class Calculadora {  
    public int suma(int a, int b){  
        int resultado = 0;  
        if(a>0 && b>0)  
            resultado=a+b;  
        return resultado;  
    }  
    public int resta(int a, int b){  
        int resultado = 0;  
        if(a>0 && b>0)  
            resultado=a-b;  
        return resultado;  
    }  
    public int multiplicacion(int a, int b){  
        int resultado = 0;  
        if(a>0 && b>0)  
            resultado=a*b;  
        return resultado;  
    }  
    public int division(int a, int b){  
        int resultado = 0;  
        if(a>0 && b>0)  
            resultado=a/b;  
        return resultado;  
    }  
}
```

```
//Creamos objeto Calculadora  
Calculadora cal = new Calculadora();  
Scanner entrada = new Scanner(System.in);  
System.out.println("Introduce dos valores para sumar:");  
int resultado=cal.suma(entrada.nextInt(), entrada.nextInt());  
System.out.println("Este es el resultado de la suma = "+resultado);  
  
System.out.println("Introduce dos valores para restar:");  
int datol = entrada.nextInt();  
int dato2 = entrada.nextInt();  
int resultado=cal.resta(datol, dato2);  
System.out.println("Este es el resultado de la resta = "+resultado);  
resultado=0;  
System.out.println("Introduce dos valores para multiplicar:");  
int resultado=cal.multiplicacion(datol, dato2);  
System.out.println("Este es el resultado de la multiplicacion = "+resultado);  
  
System.out.println("Introduce dos valores para dividir:");  
int resultado=cal.division(datol, dato2);  
System.out.println("Este es el resultado de la division = "+resultado);
```

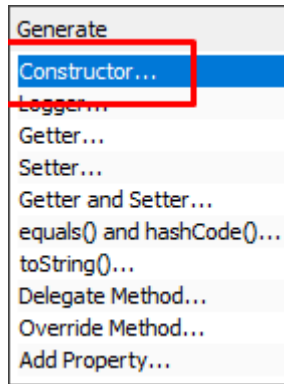
CONSTRUCTORES

- Se trata de definir cómo se puede crear un objeto.
- La sobrecarga nos permite crear el mismo método pero que acepte distintos parámetros.

```
public class Persona {  
    String nombre;  
    int edad;  
    //CONSTRUCTOR  
    public Persona() {  
        this.nombre="john Doe";  
        this.edad=0;  
    }  
    //CONSTRUCTOR  
    public Persona(String _nombre, int _edad) {  
        this.nombre=_nombre;  
        this.edad=_edad;  
    }  
}
```

CONSTRUCTORES

- NetBeans nos ayuda con ciertos códigos.
- Botón derecho
- Insertar código



ARRAYLIST

- Listas dinámicas con objetos.

- Clase

- *import java.util.ArrayList;*

- Declaración

- *ArrayList <tipo> nombreArrayList = new ArrayList<tipo>();*

ARRAYLIST: Métodos (II)

Método	Descripción
<ul style="list-style-type: none">• <code>add("elemento");</code>• <code>add(n, "elemento");</code>	<ul style="list-style-type: none">• Añadir elemento al arrayList <p>Añadir elemento al arrayList en la posición n</p>
<ul style="list-style-type: none">• <code>size();</code>• <code>get(n);</code>	<ul style="list-style-type: none">• Numero de elementos• Devuelve el elemento de la posición n
<ul style="list-style-type: none">• <code>contains("elemento");</code>	<ul style="list-style-type: none">• Comprobar si existe la ocurrencia "elemento"
<ul style="list-style-type: none">• <code>indexOf("elemento");</code>	<ul style="list-style-type: none">• Primera posición de la ocurrencia "elemento"
<ul style="list-style-type: none">• <code>lastIndexOf("elemento");</code>	<ul style="list-style-type: none">• Última posición de la ocurrencia "elemento"
<ul style="list-style-type: none">• <code>remove(posicion);</code><ul style="list-style-type: none">• <code>clear();</code>• <code>isEmpty();</code>	<ul style="list-style-type: none">• Borra el elemento en "posicion".• Borra todos los elementos del arrayList• <code>True</code> i la lista está vacía

ARRAYLIST (III)

• Método	• Descripción
<ul style="list-style-type: none">• <code>ArrayList arrayL2 = (ArrayList) arrayL1.clone();</code>• <code>Object [] array = arrayL3.toArray();</code>	<ul style="list-style-type: none">• Copiar un arrayList a otro arrayList. <p>Pasar el ArrayList a un Array</p>

ARRAYLIST Iteradores

- Sirven para recorrer los ArrayList y trabajar con ellos.

- Clase

- *import java.util.Iterator;*

- Declaración

- *Iterator<Stringtipo> nombreArrayList = new ArrayList<tipo>();*

ARRAYLIST Practica 1

- | • Método | • Descripción |
|----------------------------|--------------------------------------|
| • <code>.hasNext();</code> | • Comprobar si quedan elementos. |
| • <code>.next();</code> | • Siguiendo elemento del iterador. |
| • <code>.remove();</code> | • Eliminar el elemento del iterador. |

PRACTICA:

- Crear un ArrayList.
- Añadir 10 elementos (usar un bucle) del tipo "Elemento n".
- Sustituir el 2er elemento por "Elemento 3".
- Declarar un Iterador
- Recorrer el ArrayList con el iterador para mostrar los elementos.
- NOTA: usar bucle while() {}



ARRAYLIST Ejemplos

```
//Declaración del ArrayList lista
ArrayList <String> lista=new ArrayList<String>();
//Añadimos 10 elementos en el ArrayList
for(int i=1;i<=10;i++){
    lista.add("Elemento "+i);
}

//Añadimos nuevo elemento en la posición 2
lista.add(2,"Elemento 3");

//Declaramos Iterador e imprimimos
Iterator<String> Iter = lista.iterator();

while(Iter.hasNext()){
    String elemento = Iter.next();
    System.out.print(elemento+" / ");
}
```



ARRAYLIST Practica

PRACTICA2: (continuación del anterior)

- Mostrar el número de elementos (cantidad de elementos) del ArrayList.
- Eliminar el primer elemento.
- Eliminar los elementos iguales a “Elemento 3”



ARRAYLIST Ejemplos-Solución

```
//Mostrar número de elementos
System.out.println("El ArrayList tiene "+lista.size()+" elementos.");
//Eliminar el primero
System.out.println("Eliminamos el primer elemento.");
lista.remove(0);
.....
Iter = lista.iterator();
while (Iter.hasNext()) {
    String elemento2 = Iter.next();
    System.out.println(elemento2+" / ");
}

System.out.println("Eliminamos todos los elementos 3.");
```



ARRAYLIST Ejemplos-Solución

```
//Eliminar elementos iguales a Elemento 3
Iter = lista.iterator();
while(Iter.hasNext()){
    String elemento = Iter.next();
    if (elemento.equals("Elemento 3")){
        Iter.remove();
    }
}
Iter = lista.iterator();
while(Iter.hasNext()){
    String elemento2 = Iter.next();
    System.out.println(elemento2+" / ");
}
```



GRACIAS