

UT 1. POO, EDA y API JAVA.

SINTAXIS LENGUAJE JAVA

Sintaxis Lenguaje Java

- Sentencias y bloques de código.
 - Una sentencia Java es una instrucción que termina con el carácter “;”.
 - Un bloque es un grupo de sentencias, delimitadas por los caracteres “{ “ y “}”.
 - Tipos de sentencia:
 - De creación.
 - De asignación.
 - De llamada a una función.
 - De control de flujo de programa
 - El código debe indentarse para ganar en claridad.

```
public class HolaMundo{  
    public static void main(String[] args) {  
        System.out.println("Hola mundo");  
    }  
}
```

```
int i=0; //Sentencia de creación de variable  
i = teclado.nextInt(); //Sentencia de asignación  
boolean esPar = i % 2 == 0 ? true : false; //Sentencia de  
asignación  
if( esPar == true){ //sentencia de decisión simple  
    System.out.println("Se ha tecleado un número par");  
}else{  
    System.out.println("Se ha tecleado un número par");  
}
```

Sintaxis del Lenguaje Java.

- La función main.
 - Se define función como aquella porción de código que es ejecutado cuando es llamado por un nombre determinado.
 - Las funciones presentan dos partes diferenciadas:
 - **Definición:** bloque de sentencias que se ejecutan.
 - **Llamada a la función:** provoca que se ejecuten las sentencias.
 - La función main representa el programa Java.
 - Las funciones devuelven un valor cuando acaba su ejecución. La función main no devuelve ningún valor (*void*).
 - En Java cuando una función no necesita un objeto para ser invocada, se usa el modificador “static” en su definición 7 función o atributo de clase.
 - Para poder usar una función o dato fuera del ámbito de definición se usa el modificador “public”
 - Las funciones en su definición pueden presentar una lista de parámetros.

```
public static void main(String[] args)
```

Sintaxis de Java

-
- Palabras clave e identificadores.
 - Al igual que en otros lenguajes de programación, en Java los identificadores son conjuntos de caracteres que sirven para identificar los elementos del programa.
 - Palabras clave o reservadas: son los identificadores que forman parte del lenguaje Java, y no pueden ser usados por el programador.
-

Sintaxis de Java

- Palabras reservadas del lenguaje Java.

<code>abstract</code>	<code>continue</code>	<code>for</code>	<code>new</code>	<code>switch</code>
<code>assert^{***}</code>	<code>default</code>	<code>goto[*]</code>	<code>package</code>	<code>synchronized</code>
<code>boolean</code>	<code>do</code>	<code>if</code>	<code>private</code>	<code>this</code>
<code>break</code>	<code>double</code>	<code>implements</code>	<code>protected</code>	<code>throw</code>
<code>byte</code>	<code>else</code>	<code>import</code>	<code>public</code>	<code>throws</code>
<code>case</code>	<code>enum^{****}</code>	<code>instanceof</code>	<code>return</code>	<code>transient</code>
<code>catch</code>	<code>extends</code>	<code>int</code>	<code>short</code>	<code>try</code>
<code>char</code>	<code>final</code>	<code>interface</code>	<code>static</code>	<code>void</code>
<code>class</code>	<code>finally</code>	<code>long</code>	<code>strictfp^{**}</code>	<code>volatile</code>
<code>const[*]</code>	<code>float</code>	<code>native</code>	<code>super</code>	<code>while</code>

^{*} not used
^{**} added in 1.2
^{***} added in 1.4
^{****} added in 5.0

FIGURA 2.3: LISTA DE PALABRAS RESERVADAS DE JAVA.

Sintaxis de Java

- Reglas para la creación de identificadores.
 - El primer carácter tiene que ser una letra, el carácter `_` o el carácter `$`.
 - Puede incluir números, pero no empezar por uno de ellos.
 - No se pueden incluir espacios en blanco.
 - Se distingue entre mayúsculas y minúsculas.
 - No se pueden usar palabras reservadas.

Tipo de identificador	Convención	Ejemplo
nombre de una clase	comienza por letra mayúscula	String, System, HolaMundo
nombre de función	comienza con letra minúscula	getEdad, setNombre, calculaArea
nombre de variable	comienza por letra minúscula	edad, nombre, i
nombre de constante	en letras mayúsculas	X_MAX, PI, VERDE

Sintaxis de Java

- Tipos de datos primitivos.
 - Java esta fuertemente tipado, es decir realiza una comprobación estricta de tipos.
 - Existen 9 tipos llamados primitivos.

Tipo	Representación	Bytes	Rango	Valor por omisión	Clase envoltorio
char	Caracter Unicode	2	\u0000 a \uFFFF	\u0000	Character
byte	Entero con signo	1	-128 a 127	0	Byte
short		2	-32768 a 32767	0	Short
int		4	-2147483648 a 2147483647	0	Integer
long		8	-9223372036854775808 a 9223372036854775807	0	Long
float	Punto flotante precisión simple	4	$\pm 3.4 \cdot 10^{-38}$ a $\pm 3.4 \cdot 10^{38}$	0.0	Float
double	Punto flotante precisión doble	8	$\pm 1.8 \cdot 10^{-308}$ a $\pm 1.8 \cdot 10^{308}$	0.0	Double
boolean	Lógico	1	true o false	false	Boolean
void	-	-	-	-	Void

Sintaxis de Java

- Literales.
 - Los literales son valores fijos o constantes, representados en forma legible para las personas.
 - Cada tipo primitivo tiene su forma de representar sus literales.

Tipos	Representación	Ejemplos
char	Encerrados entre ''	'a', '1', '\n', '\u0037
byte, short, int	Los dígitos del número, sin parte fraccionaria. Por omisión todos son de tipo int.	1245, -345678, 3_456_789 El carácter _ hace de separador de miles
long	Como los anteriores pero terminados en l o L	34567898923L, -383838383l, 89_678_578_976L
double	Con parte decimal, por omisión todos los punto flotante son double	0.0, -12345e-125, .0456
float	Como los anteriores seguidos de f o F	0.0F, -1245.35f, 345.43e12F, .0f
boolean	true o false	
String	Aunque no es un tipo primitivo, sus literales tienen representación como caracteres delimitados por ""	"Hola mundo", "\nNombre: "

Sintaxis de Java

- Variables, ámbito y duración.

- Una variable en Java es una posición de memoria en la que se almacena un valor de un tipo primitivo.
- Las variables se declaran, se inicializan y se usan.
- Declaración de una variable:

```
<tipo> <identificador> [= <valor_inicial>];
```

- Declaración de un grupo de variables del mismo tipo:

```
<tipo> <identificador> [= <valor_inicial>]{ <identificador>  
[= <valor_inicial>],};
```

```
int i=21;  
char c='A';  
String nombre = "Pepe";    //en realidad nombre  
                             es una referencia  
                             //a un objeto String  
float sueldo=1245f, complemento=321.5f;  
short a, c=4, d=6; f;  
long numero= 4_567_890;  
float base=6.0f, altura=14.3f;  
double area = base * altura / 2;
```

Sintaxis de Java

-
- Existen tres tipos de variables en Java:
 - Locales: se declaran dentro de los bloques de una función, su ámbito es el bloque en el que se crean y su duración desde que se crean hasta que finaliza el bloque.
 - De instancia: se declaran en el bloque de una clase, se instancian cuando se crea un objeto de esa clase. Su ámbito es el definido con los modificadores *public*, *protected* o *private*. Se acceden siempre a través del objeto al que pertenecen.
 - De clase: espacios de memoria compartidos por todos los objetos de una clase. Se declaran en el ámbito de una clase pero precedidas por el modificador *static*
-

Sintaxis de Java

- Operadores y Tipos.

- Un operador es un símbolo que relaciona elementos mediante una funcionalidad. Indican al compilador que realice una operación. Los elementos que relaciona un operador son los operandos.
- Tipos de operadores:
 - Unarios.
 - Binarios.
 - Ternarios.

- **Aritméticos:** +, -, *, /, %, ++, --
 - **De asignación:** =, +=, -=, *=, /=, %=
 - **Relacionales:** ==, !=, <, >, <=, >=
 - **Lógicos:** ^, &&, ||, !, &, |
 - **Concatenación:** +
 - **Nivel u orden de bits:** &, !, ^, <<, >>
 - **Especiales:** ?, new, [], ()
-

Expresiones y precedencia de operadores.

-
- Una expresión es un conjunto de operandos y operadores que produce un resultado.
 - Cuando se utilizan dos o más operandos hay que tener en cuenta la prioridad de evaluación:
 - Los operadores se evalúan de izquierda a derecha.
 - No todos los operadores tienen la misma prioridad.
 - El orden de evaluación solo es aplicable para los operadores que tengan la misma prioridad.
 - Para cambiar el orden de evaluación se usan los paréntesis. Siempre se evalúa primero lo que está entre paréntesis.
 - Los operadores de prioridad son los que menos prioridad tienen.
 - A igualdad de prioridad se evalúa de izquierda a derecha.
-

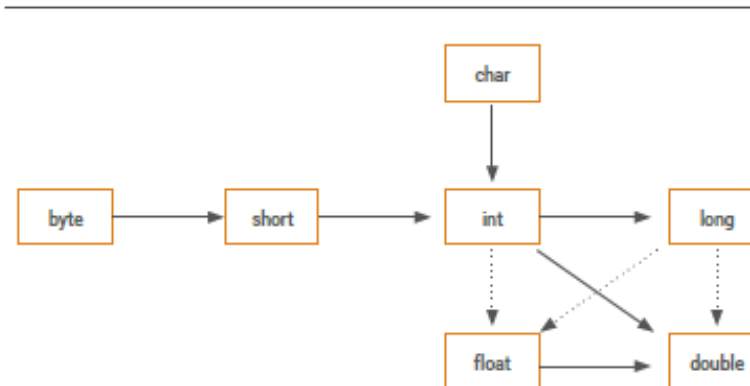
Sintaxis de Java

- Orden de prioridad de los operadores.

Operadores	Tipo	Asociatividad
[] . () y llamada a método	Especiales	De izquierda a derecha
! ~ ++ -- +(unario) -(unario) (moldeo) new	Unarios, moldeo y creación	De derecha a izquierda
* / %	Aritméticos	De izquierda a derecha
+ -	Aritméticos	De izquierda a derecha
<< >> >>>	Nivel de bits	De izquierda a derecha
< <= > >=	Relacionales	De izquierda a derecha
== !=		De izquierda a derecha
&	Lógicos	De izquierda a derecha
^		De izquierda a derecha
		De izquierda a derecha
&&		De izquierda a derecha
		De izquierda a derecha
?:	Ternario	De izquierda a derecha
Todos los de asignación		De derecha a izquierda

Sintaxis de Java

- Conversión de tipos.
 - El tipo de variable y el valor a asignar debe ser compatible. Todos los tipos numéricos son compatibles, pero debemos tener cuidado con el tamaño de los tipos, ya que se puede perder precisión.
 - Las conversiones validas entre tipos son las siguientes:



Las flechas continuas implican conversiones sin pérdida de información.
Las flechas discontinuas implican conversiones en las que puede haber pérdida de información.

Sintaxis de Java

-
- Conversión de tipos.
 - Cuando en expresiones se relacionan, valores numéricos con operadores aritméticos, debemos tener en cuenta las siguientes reglas:
 - Si alguno de los operandos es double, el otro se convierte a double.
 - Si ninguno es double, pero uno es float, el otro se convierte a float.
 - Si ninguno es double o float, pero uno es long, el otro se convierte a long.
 - En cualquier otro caso se convierte a int.
 - Estas conversiones son todas implícitas. Podemos realizar conversiones de manera explícita. `variable_destino = (tipo_destino) variable_origen;`
 - A esta operación se le denomina casting.
-

Sintaxis de Java

-
- Entrada y Salida Básica.
 - En Java todo el sistema de entrada y salida se basa en una jerarquía de clases.
 - Todo el sistema de E/S se realiza a través de flujos.
 - Para escribir en la consola se utiliza el flujo “out”, este flujo consume información.
 - Funciones:
 - println
 - print.
-

Sintaxis de Java

- Para introducir información a través del teclado se utiliza el flujo “in”.
 - Podemos usar un objeto de la clase Scanner.

```
//Se crea objeto Scanner para el flujo "in"  
Scanner teclado = new Scanner(System.in);  
//La función nextLine de Scanner devuelve todos los  
//caracteres tecleados hasta el carácter "fin de línea"  
String cadena = teclado.nextLine();
```

- Otras funciones de Scanner para leer información son las siguientes:
 - Next(): lee como String lo tecleado hasta encontrar un espacio en blanco.
 - nextXXX(): donde xxx es uno de los tipos primitivos excepto char.
- Si se utiliza la función next o cualquier nexXXX, el carácter fin de línea no lo vacían del buffer quedando en espera, si la siguiente sentencia utiliza nextLine(), esta saca del buffer dicho carácter y como es un carácter terminador para ella no lee la cadena que estuviera en la línea siguiente.

Sintaxis de Java

- Uso de la función nextInt y la función nextLine.

```
System.out.print("Teclea un número: ");

//Teclear número y almacenarlo en variable
n=teclado.nextInt();

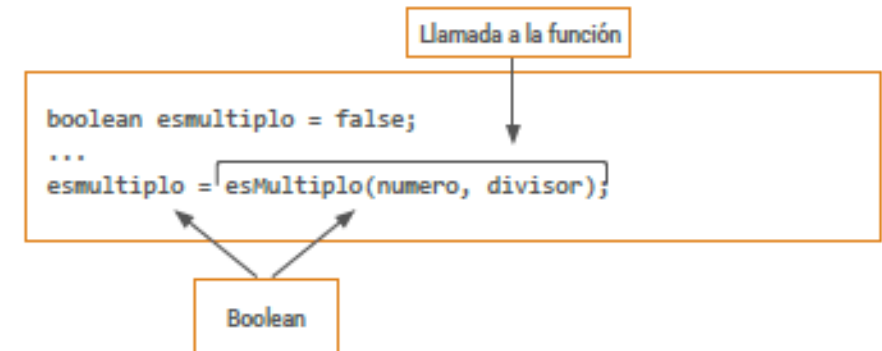
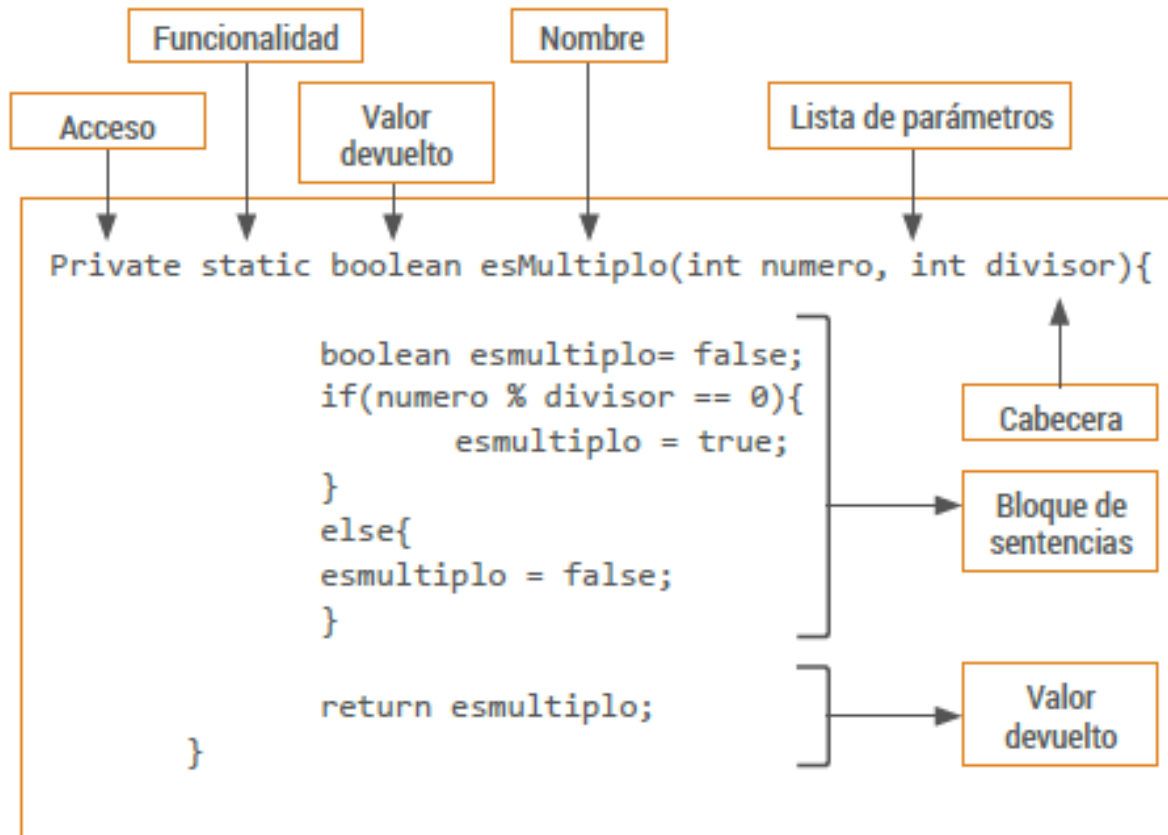
//Declarar una referencia para cadenas de caracteres
String nombre;
System.out.print("Teclea tu nombre: ");

//Teclear una cadena de caracteres
//Hay que recoger el CR (fin de línea) pulsado para introducir número
teclado.nextLine();
nombre=teclado.nextLine();
```

Sintaxis de Java

-
- Funciones y parámetros.
 - Se trata de un bloque de sentencias que se ejecutan cuando es llamada devolviendo un resultado.
 - En la definición de una función distinguimos distintas partes:
 - <acceso> <funcionalidad> <tipo_devuelto> <nombre_función> (<parámetros>
 - El bloque de sentencias a ejecutar se encierra entre llaves.
 - El valor devuelto se realiza mediante la sentencia return.
 - El ámbito de una función, es desde que es llamadas hasta que retorna el valor.
 - Una función define un ámbito local.
-

Sintaxis de Java



Sintaxis de Java

- Comentarios.
 - Los comentarios son anotaciones ignoradas por el compilador.
 - Pueden ser de dos tipos:
 - Comentarios de una sola línea: marcados por los caracteres //.
 - De varias líneas: acotados por los caracteres /*.....*/

```
int n; //Crear una variable
System.out.print("Teclea un número: ");
//Teclear número y almacenarlo en variable
n=teclado.nextInt();
//Declarar una referencia para cadenas de caracteres
String nombre;
System.out.print("Teclea tu nombre: ");
/*Teclear una cadena de careacteres
Hay que recoger el CR pulsado para introducir número*/
teclado.nextLine();
nombre=teclado.nextLine();
```

Comentario de línea

Comentario de varias líneas

Sintaxis de Java

-
- Ejercicios de Programación en Java
-

Ejercicios de Programación en Java

- Ejercicio_1.
 - Hacer un programa que calcule e imprima la suma de tres notas.
 - Ejercicio_2.
 - Hacer un programa que calcule e imprima el salario semanal de un empleado a partir de sus horas semanales trabajadas y de su salario por hora.
 - Ejercicio_3.
 - Guillermo tiene N euros. Luis tiene la mitad de lo que posee Guillermo. Juan tiene la mitad de lo que poseen Luis y Guillermo juntos. Hacer un programa que calcule e imprime la cantidad de dinero que tienen entre los tres.
 - Ejercicio_4.
 - Una compañía de venta de coches usados paga a su personal de ventas un salario de 1000€ mensuales, más una comisión de 150€ por cada coche vendido, más el 5% del valor de la venta del coche. Hacer un programa que calcule e imprima el salario mensual de un vendedor dado.
-