PROGRAMACIÓN.

U4 - Identificación de los elementos de un programa informático.

ÍNDICE

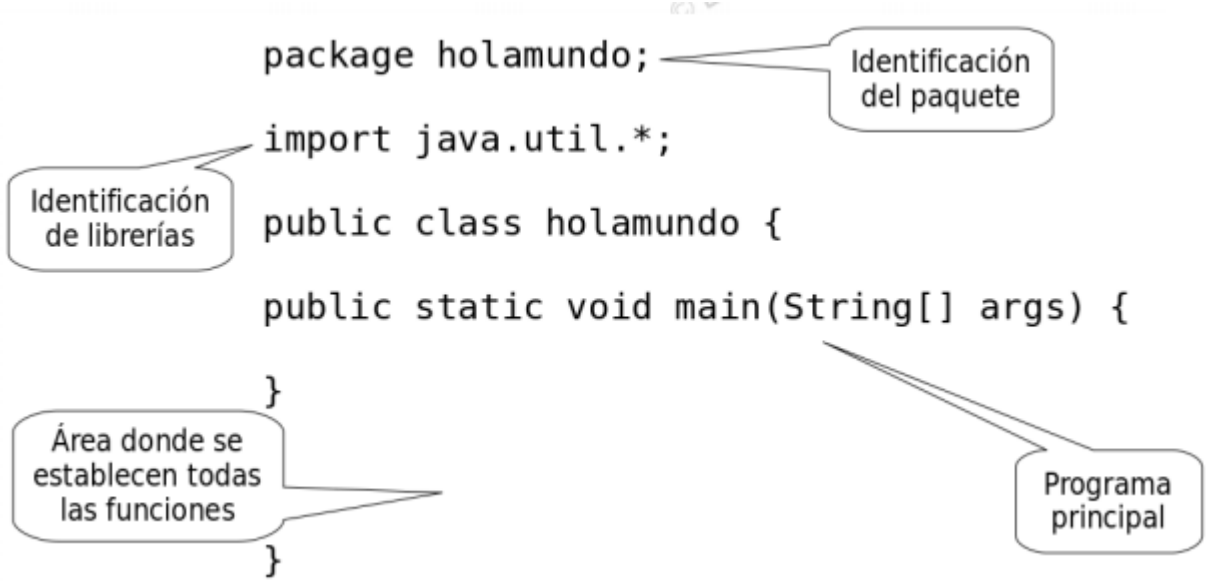
I. Instalación del IDE

- **PSeInt**. Herramienta para asistir a un estudiante en sus primeros pasos en programación. Mediante la integración de un simple e intuitivo pseudolenguaje y funcionalidades de diagrama de flujos. http://pseint.sourceforge.net/
- **Eclipse**. Eclipse es una plataforma de software compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar.

Instalación para su uso con Java:

* Eclipse: https://www.eclipse.org/
* JAVA JDK: https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

II. Estructura y bloques fundamentales



III. Identificadores. Palabras reservadas

Normas obligatorias:

- Diferenciar entre mayúsculas y minúsculas: No son los mismo los identificadores "NombreVariable" y "nombreVariable".
- Comenzar siempre con una letra, un subrayado (_) o símbolo de dólar ($)
- No usar como identificador una palabra reservada.
- No existe extensión máxima.

Normas recomendadas:

- Los nombres de clases empiezan con mayúsculas.
- La primera letra de las variables y los métodos con minúsculas.
- Los nombres compuestos se unen con letra mayúsculas en el comienzo de cada palabra.
- Se usan nombres largos y significativos.

PALABRAS RESERVADAS.

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

IV. Variables. Tipos de datos.

VARIABLE:

¿ Que es una variable ? Es una forma de almacenar información en nuestro ordenador. Las variables que hemos definido en un programa pueden ser accedidas por el nombre que le hemos dado. El ordenador hará el trabajo duro que hay almacenado en la variable deseada accediendo a la memoria RAM (random access memory) de nuestro ordenador.

TIPOS DE VARIABLES:

Tipo	Representación / Valor	Tamaño (en bits)	Valor mínimo	Valor máximo	Valor por defecto
`boolean`	`true` o `false`	1	N.A.	N.A.	`false`
`char`	Carácter Unicode	16	\u0000	\uFFFF	\u0000
`byte`	Entero con signo	8	-128	128	0
`short`	Entero con signo	16	-32768	32767	0
`int`	Entero con signo	32	-2147483648	2147483647	0
`long`	Entero con signo	64	-9223372036854775808	9223372036854775807	0
`float`	Coma flotante de precisión simple Norma IEEE 754	32	±3.40282347E+38	±1.40239846E-45	0.0
`double`	Coma flotante de precisión doble Norma IEEE 754	64	±1.79769313486231570E+308	±4.94065645841246544E-324	0.0

```
byte variable_byte = 5;
short variable_short = 5;
int variable_int = 5;
long variable_long = 5;
float variable_float = 5;
double variable_double = 5.0;
char variable_char = 'a';
boolean variable_boolean = true;
```

Definición de una variable en JAVA:

```
public class HolaMundo {

    public static void main(String[] args) {

        int miVariable = 10;

    }

}
```

IV. Conversión de tipos

CONVERSIÓN DE TIPOS.

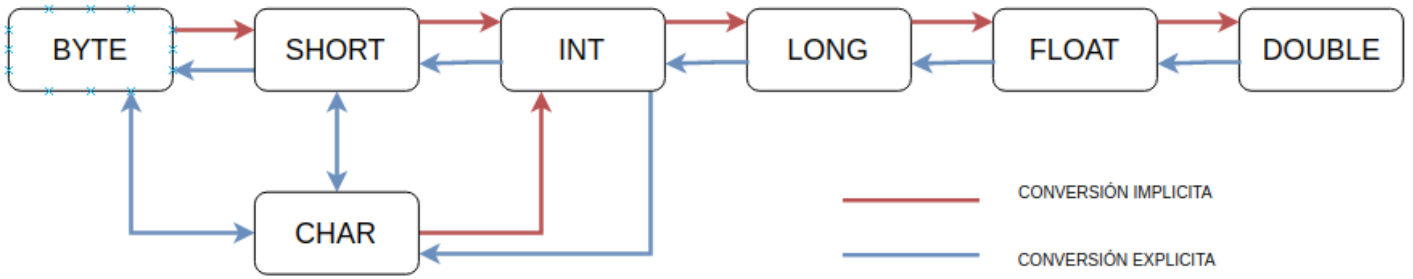
*** Conversiones implícitas. Automático.**

Requiere que la variable destino (la que se coloca a la izquierda) tenga más precisión que la variable origen (situada a la derecha).

*** Conversiones explícitas. Forzado por el programador.**

Mediante la operación llamada cast con el siguiente formato:

(tipo) expresión



V. CONSTANTES.

VARIABLE:

Las constantes se utilizan en datos que nunca van a variar.

Nos aseguramos que su valor no va a poder ser modificado nunca.

```
public class HolaMundo {

    public static void main(String[] args) {

        final double PI = 3.1416;

    }

}
```

Nota: Las constantes se definen con mayúsculas y las variables con minúsculas. Regla de estilo.

VI. Operadores y expresiones. Aritmética.

Operación en Java	Operador aritmético	Expresión algebraica	Expresión en Java
Suma	+	$f + 7$	`f + 7`
Resta	-	$p - c$	`p - c`
Multiplicación	*	bm	`b * m`
División	/	x/y o $\frac{x}{y}$ o $x \div y$	`x / y`
Residuo	%	$r \bmod s$	`r % s`

VI. Operadores y expresiones. Asignación

Operadores de asignación

Operador	Operación	Utilización	Equivalencia
=	Asignación	x = 5	
+=	Suma	x +=3	x = x + 3
-=	Resta	x -=3	x = x - 3
*=	Multiplicación	x *=3	x = x * 3
/=	División	x /=3	x = x / 3
%=	Módulo	x %=3	x = x % 3

VII. Operadores y expresiones. Toma de decisión

Operadores racionales

Operadores relacionales	
<	Menor
>	Mayor
<=	Menor o igual
>=	Mayor o igual

Operadores de comparación

Operadores de comparación	
==	Igual
!=	No igual

Operadores lógicos

Operadores lógicos	
&&	and
||	or

VI. Operadores y expresiones. Posfijos y prefijos

expr++	Posincremento: x=x+1 se puede escribir x+=1, pero también x++
expr--	Posdecremento: x=x-1 se puede escribir x-=1, pero también x--

++expr	Preincremento: x=x+1 se puede escribir x+=1, pero también ++x
--expr	Predecremento: x=x-1 se puede escribir x-=1, pero también --x
+expr	Número positivo: +5
-expr	Número negativo: -5
~	Complemento en bits
!	Negación booleana
?:	Condicional (se verá en el apartado de instrucciones)
new	Invocación al constructor del tipo
(tipo) expr	Conversión (casting) de un valor int x = (int) 3.28;

VIII. Comentarios

- Es un texto que introducimos para facilitar el entendimiento del código fuente.
- El compilador y el intérprete ignoran.

- Tipos:
  - Comentario de una sola línea. (“//”)
  - Comentario de múltiples líneas. (“/* */”)
  - Comentario de documentación.
    - Javadoc (Herramienta de documentación)

```
// Comentario de una sola línea
```

```
/* Este es un comentario
     tradicional.  Puede
     dividirse en muchas líneas */
```

```
/**
 * Programa para imprimir texto.
 * @author ProfesorJava
 * @version 2010
 */
```

```
/**
 * Gets the geocoordinates of roadrunners based on your city and state.
 *
 * @param city the city you want to browse for roadrunners
 * @param state the state you want to browse for roadrunners
 * @return the coordinates of the roadrunner in your area
 * @throws IOException if you put integers instead of strings
 */
public String findRoadRunner(String city, String state) throws IOException {

    System.out.println("location: " + city + ", " + state);
    System.out.println("getting geocoordinates of roadrunner.... ");
    System.out.println("roadrunner located at " + LongLat);
    return LongLat;

}
```

IX. FLUJO DE ENTRADA Y SALIDA.

- FLUJO DE ENTRADA: MOSTRAR INFORMACIÓN


```
Escribir "Escribir valor de salida"
Escribir valor_salida
```

```
int valor_salida = 5;
System.out.println("Escribir valor de salida");
System.out.println(valor_salida);
```

- FLUJO DE SALIDA: RECIBIR INFORMACIÓN POR PARTE DEL USUARIO.

```
Leer valor_entrada

Leer cadena_entrada
```

```
// Creamos escaner de entrada.
Scanner escaner_entrada = new Scanner(System.in);

// Variables numéricas
boolean valor_entrada_boolean = escaner_entrada.nextBoolean();
byte valor_entrada_byte = escaner_entrada.nextByte();
short valor_entrada_short = escaner_entrada.nextShort();
int valor_entrada_int = escaner_entrada.nextInt();
long valor_entrada_long = escaner_entrada.nextLong();
float valor_entrada_float = escaner_entrada.nextFloat();
double valor_entrada_double = escaner_entrada.nextDouble();

//Cadena de caracteres
String valor_entrada_cadena = escaner_entrada.nextLine();
```

X. BUENAS PRÁCTICAS & ERRORES COMÚNES.

BUENAS PRÁCTICAS:

Identificar los paquetes, clases, métodos y variables según unas premisas recomendadas. UTILIZAR NOMBRES DE VARIABLES SIGNIFICATIVAS.
Siempre que utilizamos una llave de apertura ({) , debemos cerrarlo (}).
Aplicar las sangrías y la indexación del código de forma correcta.

ERRORES COMUNES:

No utilizar las llaves por pares.
Omitir el (;) tras una instrucción o sentencia de código.
Confunfir el operador de igualdad (==) con el de asignación (=).
No utilizar exactamente el mismo nombre de la variable. Recuerda es “Case sensitivity”, es decir, sensible a mayúsculas y minúsculas.

XI. EJ ERCICIO

Enunciado: Crear un programa en Java:

Creación del primer programa en Java. En este enlace, se puede ver la creación del programa que imprimirá “Hola Mundo” en la consola de salida (output).

Se pide:

De lo visto, nos interesa solamente la creación del proyecto, la creación de la clase ya la veremos en unidades posteriores. Cuando se crea el proyecto, que habrá de llamarse “holamundo”, se crea una fuente Java con la siguiente estructura:

```
/*
* To change this license header, choose License Headers in Project Properties.
* To change this template file, choose Tools | Templates
* and open the template in the editor.
*/
package holamundo;
/**
*
* @author autor
*/
public class Holamundo {
/**
* @param args the command line arguments
*/
public static void main(String[] args) {
// TODO code application logic here
}
}
```

¿ DUDAS ?

¿ Dudas ?


IMF
Formación
Profesional
MUCHAS GRACIAS