



1. Estructura y bloques fundamentales

 Date

Empty

 Status

Not started

 Type

Empty

Units



2. Estructura de un programa informático

Bloques

Identificadores

Estructuras y bloques fundamentales

Clases

Variables

Métodos

Palabras reservadas

Ejercicios clase

Bloques

Los bloques de código son partes del programa que tienen una ejecución específica. Dentro de los bloques encontramos los

estamentos que se encargan de realizar por ejemplo repeticiones, toma de decisiones, etc...

Identificadores

Un identificador es una palabra que representa elementos de un lenguaje de programación.

En java un identificador se define del siguiente modo:

- Comienza con una letra, un subrayado (_) o un símbolo de dólar (\$). Los siguientes caracteres pueden ser letras, dígitos y subrayado.
- Se distinguen las mayúsculas de las minúsculas.
- No hay una longitud máxima establecida para el identificador

Estructuras y bloques fundamentales

Los bloques fundamentales dentro de un programa son:

- Clases
- Variables
- Métodos
- Bloques

Existen más elementos que se explicarán más adelante pero los fundamentales son los comentados.

Clases

Representan el "molde" desde el cual se pueden "hacer" realidad cada uno de los elementos que formarán parte del programa informático. Para su creación se utiliza la palabra reservada `class`. La sentencia correcta para su creación es

```
modificador_acceso nombre_clase { }
```

```
public class Ejemplo{ }
```

Lo normal es la existencia de una clase por cada archivo, el cual debe tener el mismo nombre de la clase. Como mínimo siempre debe existir una clase en cada proyecto la cual tenga el método main que veremos en los siguientes puntos

Variables

Serán explicadas más adelante y tienen como objetivo guardar datos que serán utilizados durante la ejecución del programa. Su estructura básica es la siguiente:

```
tipo nombre = valor String palabra = "Hola"
```

Métodos

Son las funcionalidades individuales que pueden ser llamadas desde cualquier parte del código. Para poder ejecutar todo el interior del mismo simplemente hay que llamarlo por su nombre, con los parámetros que se indiquen. Su estructura o firma es la siguiente

```
modificador_acceso valor_retorno nombre_método(parámetro nomb  
re) {
```

En el caso de tener un valor de retorno es obligatorio que tenga la palabra reservada return

```
public int sumarNumeros(int op1, int op2){ int suma = op1+op  
2; return suma; } sumarNumeros(5,8);
```

```
public void mostrarMensaje(){ System.out.println("Sacar mensa  
je por consola"); } mostrarMensaje();
```

Existen también métodos estáticos, los cuales pueden ser llamados simplemente con el nombre de la clase.nombre_metodo

Su sintaxis es la siguiente:

```
static retorno nombre_método (parámetros) { }
```

El ejemplo más importante de método estático es el método main. Dicho método es la puerta de entrada del programa, el primer método que se ejecuta nada más arrancar el programa. El método tiene la siguiente firma:

```
public class Main { public static void main(String[] args) {  
    System.out.println("Hello world!"); } }
```

Como se puede comprobar, dicho método está englobado dentro de una clase con el nombre Main (este nombre puede cambiar), no retorna nada (void) y tiene como parámetros de entrada un array de palabras (String[]). Dentro del mismo se utiliza un método que imprime por consola la palabra Hello world!. Es muy importante tener en cuenta que solo puede haber un método main en el proyecto (ya que es desde donde se ejecutará el inicio del programa)

En una clase, podemos crear tantos métodos como queramos, lo único que hay que tener cuidado a la hora de decláralos, ya que si se quieren llamar desde otro método deberán ser declarados estáticos o utilizados a través del objeto relacionado (no os preocupéis porque esto lo explicaremos bien más adelante). Un ejemplo sería el siguiente:

```
public class Main { public static void main(String[] args) {  
    // ejecución por objeto new Main().metodoUno(); // ejecución  
    directa metodoDos(); } public void metodoUno(){  
    System.out.println("Ejecución del método uno"); } public  
    static void metodoDos(){ System.out.println("Ejecución del  
    método uno"); } }
```

Palabras reservadas

Las palabras reservadas son aquellas que tienen un significado específico y no pueden ser utilizadas para definir elementos como

específico y no pueden ser utilizadas para definir elementos como variables, métodos y/o clases. Las palabras reservadas en java son:

abstract	assert	boolean	break	byte
char	class	const	continue	default
else	enum	extends	final	finally
goto	ifv	implements	import	instanceof
long	native	new	package	private
return	short	static	strictfp	super
this	throw	throws	transient	try
while				

Ejercicios clase

En este ejemplo podemos ver los dos usos. El métodoUno se llama a través del objeto de la clase (new Main), mientras que el métodoDos se llama directamente por ser estático y estar en la misma clase de donde es llamado

1. Crea un nuevo programa que muestre ¡Hola mundo!, para la salida por consola se haga en un método sin datos de entrada ni salida de nombre "saludar".
2. Crea un método que se llame saludarPerso el cual admita un parámetro de tipo string y muestre por consola el mensaje "Hola NombreIntroducido".
3. Crea un programa en el se defina una constante global con el valor del número pi (3,1416) y se muestre por consola. Se definirá un método de nombre "mostrarPI" sin datos de entrada ni salida. (mostrarPI)
4. Crea un método que admita como parámetros dos números y muestre por consola su suma.