

Tipo	Tamaño	Rango
byte	8	-128 a 127
short	16	-32.768 a 32.767
int	32	-2.147.483.648 a 2.147.483.647
long	64	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.808
float	32	+/- 3.4E+38F (6-7 dígitos importantes)
double	64	+/- 1.8E+308 (15 dígitos importantes)
boolean	16	True o False
char	8	conjunto de caracteres UNICODE ISO

Las constantes se declaran así.

```
final int MAXIMO = 99999;
```

Asignación de variable a booleanos

Operador	Uso	Devuelve verdadero
&&(and)	condicion1&&condicion2	condicion1 y condicion2 verdaderas
(or)	condicion1 condicion2	condicion1 o condicion2 verdaderas
!(not)	!condicion	condicion falsa

```
float totaldescuento = total>=100 ? total-(total*(float)0.05) : total;
```

```
boolean salir = (lluvia == false && !tareas)||(!biblio);
System.out.println(salir);
```

formatear datos con printf

- %c: Escribe un carácter
- %s: Escribe una cadena de texto.
- %d: Escribe un entero.
- %f: Escribe un número en punto flotante.
- %e: Escribe un número en punto flotante en notación científica.

```
System.out.printf("El importe del iva es el siguiente: %,.2f%s\n", importeiva, "€");
```

suma de variable con ++

```
int x = 5, y = 5;
System.out.println(++x); // imprime 6
System.out.println(x); // imprime 6
System.out.println(y++); // imprime 5
System.out.println(y); // imprime 6
```

asignación de valores a variable

=	x=y	Asigna a x el valor de y
+=	x+=valor	Equivalente a x=x+valor
-=	x-=valor	Equivalente a x=x-valor
=	x=valor	Equivalente a x=x*valor
/=	x/=valor	Equivalente a x=x/valor

para usar scanner:

import java.util.Scanner;

```
public static void main(String[] args) {
    Scanner teclado = new Scanner(System.in);

    float base;
    System.out.println("Introduce la base imponible: ");
    base=teclado.nextFloat();
}
```

para usar parse

import javax.swing.JOptionPane;

```
public static void main(String[] args) {
    String texto; //variable para almacenar los datos recogidos por la caja de texto
    float precio;
    texto = JOptionPane.showInputDialog("Escriba el precio");
    precio = Float.parseFloat(texto);

    //o precio = Float.parseFloat(JOptionPane.showInputDialog("pon el precio"));
}
```

para usar system

```
InputStreamReader isr = new InputStreamReader(System.in);
BufferedReader br = new BufferedReader(isr);
System.out.print("Introduce el texto: ");
String cad = br.readLine();
```

```
int num = Integer.parseInt(cad));
```

Java math

import java.lang.Math;

Enum

```
public enum Meses {PEQUEÑO,MEDIANO,GRANDE}
```

```
enum Meses {
    ENERO, FEBRERO, MARZO, ABRIL, MAYO, JUNIO, JULIO, AGOSTO, SEPTIEMBRE, OCTUBRE,
    NOVIEMBRE, DICIEMBRE}
}
```

```
Meses mes = Meses.MARZO;
mes=Meses.valueOf("MARZO"); //asigno a esta variable la seleccion enum marzo
//a atraves de un string, es lo mismo que lo anterior pero en forma de texto
```

Creación de un constructor con parámetros.

```
//genero las variables
private int aceite, ruedas, contadorCambioTotales, contadorCambiosParciales;

//genero el constructor
public Taller(int inicialesBidonesAceite, int inicialesRuedas) {
    //cambio los bidones pasados por teclado a litros y meto los parametros a las variables principales
    this.aceite = inicialesBidonesAceite * 5;
    this.ruedas = inicialesRuedas;
}
```

constructor copia

```
// Constructor copia
public Taller(Taller objTaller){
    //referenciamos lo que tiene el objeto taller a la copia
    this.aceite=objTaller.aceite;
    this.ruedas=objTaller.ruedas;
}
```

Generar el objeto

```
// creo el objeto con el constructor por parametros
Taller davidMotor = new Taller(bidones, ruedas);
```

```
//creo el objeto copia
Taller copiaDavidMotor = new Taller(davidMotor);
```

	La misma clase	Otra clase del mismo paquete	Subclase de otro paquete	Otra clase de otro paquete
public	X	X	X	X
protected	X	X	X	
default	X	X		
private	X			

Remplazar una string

```
//creo el metodo
public String ocurrencias (String frase)
{
    //hago que se sustituya es por no por
    frase = frase.replaceAll("es", "no por");
    //devuelvo el resultado
    System.out.println(frase);
    return frase;
}
```

sacar la mitad de la frase de una string

```
//recogo el largo de la frase
int largo = frase.length();
//calculo el largo de la mitad de la frase
int largomitad = largo/2;

//guardo en frase la mitad de la frase
frase = frase.substring(0,largomitad);
//saco por teclado la informacion
System.out.println(frase);
return frase;
```

reemplazar vocales de una string

```
public String vocalesToO (String frase)
{
    frase = frase.replace('a','o');
    frase = frase.replace('e','o');
    frase = frase.replace('i','o');
    frase = frase.replace('u','o');
    System.out.println(frase);
    return frase;
}
```

sacar la longitud del string

```
//saco la longitud
System.out.println("la longitud del nombre es de :"+nombre.length());
```

concatenar string

```
nombre = nombre.concat(primerapellido);
nombre = nombre.concat(segundoapellido);
System.out.println(nombre);
```

comparar strings

```
System.out.println("¿Las cadenas son iguales?: "+texto1.equalsIgnoreCase(texto2));
```

startsWith(subcadena) y **endsWith(subcadena)**: para comprobar si una cadena comienza o finaliza con una subcadena determinada.

trim(): elimina los espacios en blanco de una cadena que tenga por delante o por detrás. No elimina los espacios intermedios.

toUpperCase() y **toLowerCase()**: me permite cambiar todos los caracteres por mayúsculas o minúsculas.

indexOf(cadenaABuscar): permite buscar una cadena dentro de otra.

indexOf(cadenaABuscar, posicion): igual que la anterior, pero desde una posición determinada.

replace(cadenaABuscar, cadenaSustituta): permite reemplazar una cadena por otra.

lastIndexOf (String cad): Retorna la posición de la ultima ocurrencia de la cadena dada como parámetro.

lastIndexOf (String cad, int ini): Retorna la posición de la última ocurrencia de la cadena dada como parámetro buscando en retroceso a partir de la posición dada como parámetro.

Teoría

Encapsulación : toda la información importante de un objeto dentro del mismo y solo expone la información seleccionada al mundo exterior.

Abstracción : permite representar datos y características de un objeto pero ocultando el proceso y la complejidad o el usuario interactúa solo con los atributos y métodos seleccionados de un objeto, utilizando herramientas simplificadas de alto nivel para acceder a un objeto complejo.

Herencia : significa que una clase hereda atributos y rasgos de su clase padre, de forma que atributos y métodos comunes puedan ser reutilizados. Las clases principales extienden atributos y comportamientos a las clases secundarias

Polimorfismo consiste en tener varios objetos de diferentes clases pero que se usan de la misma manera , aun dando resultados diferentes.

Modularidad : consiste en dividir el código en diferentes partes que pueden usarse de manera individual o conjunta