



Programación Orientada a Objetos con Java.



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



CONCEPTOS GENERALES



Cartagenag9

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



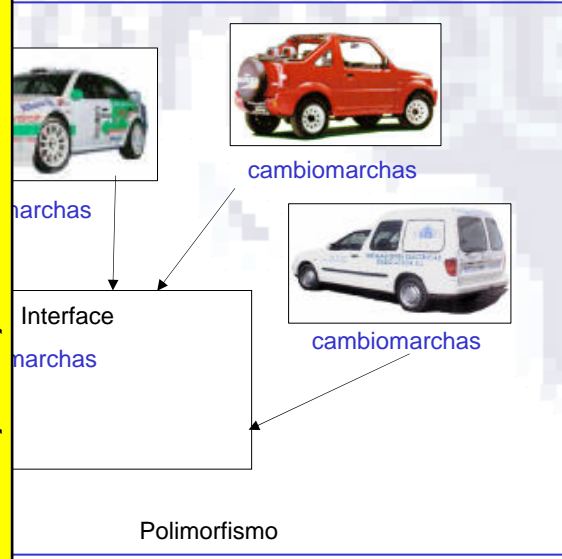
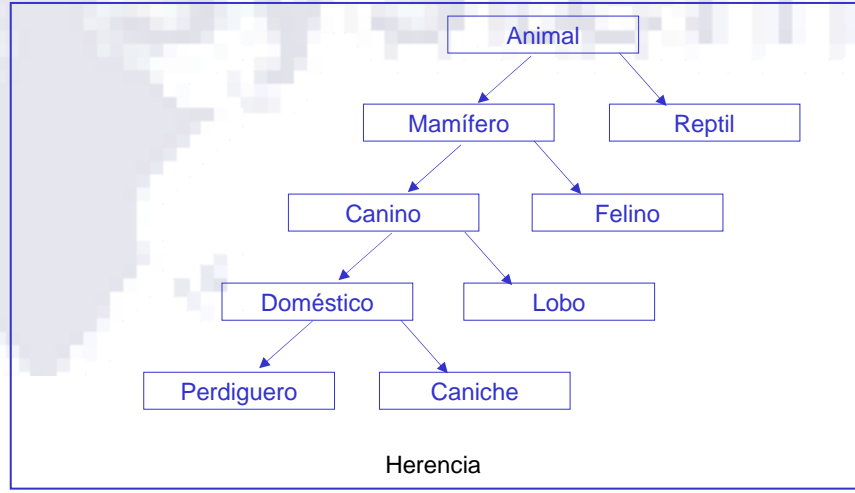
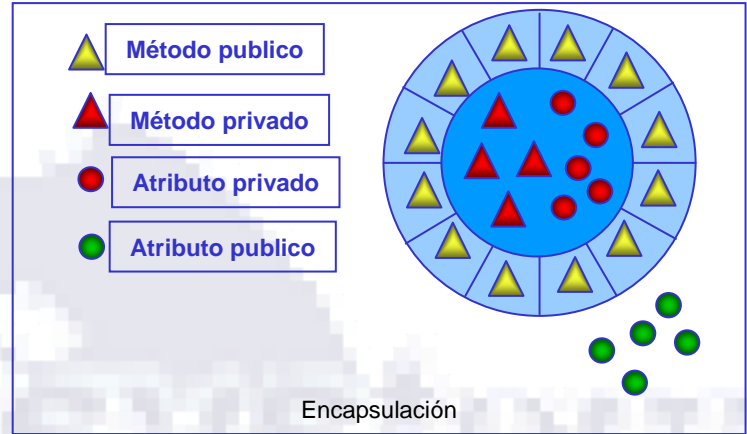
Introducción al lenguaje Java

Lenguaje orientado a objetos

Encapsulación

Herencia

Polimorfismo



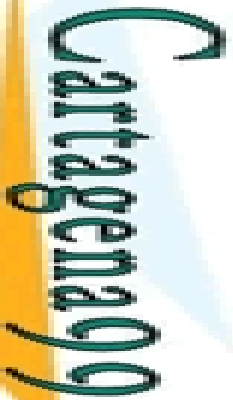
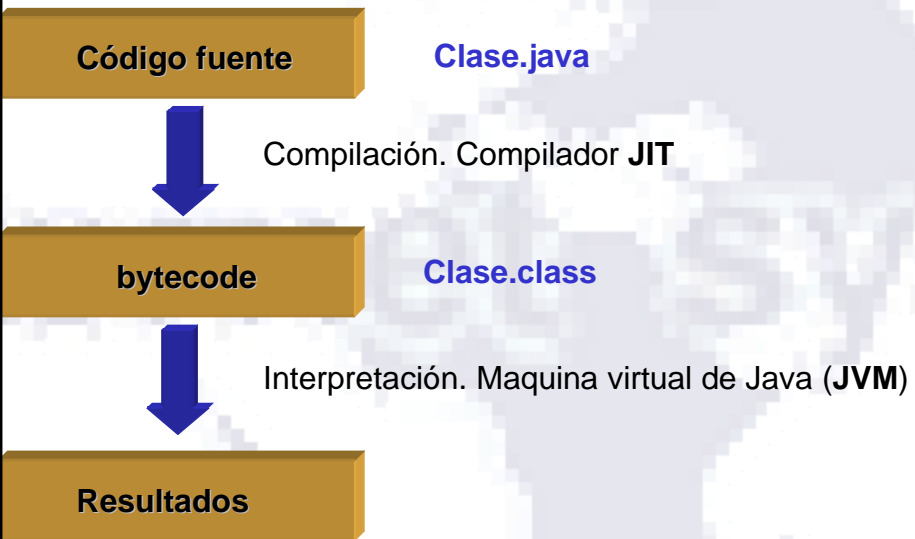
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70





Introducción al lenguaje Java

Lenguaje compilado e interpretado



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70





Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



Variables

variable es un **nombre** que contiene un valor que puede variar a lo largo del programa.

Existen dos tipos principales de variables:

Variables de **tipos primitivos**. Están definidas mediante un valor único.

Variables **referencia**. Se refieren a una información más compleja: **arrays** u **objetos** de una determinada clase

de acuerdo con el punto de vista de su papel en el programa, las variables pueden ser:

Variables **miembro** de una clase: Se definen en una clase, fuera de cualquier método. Pueden ser tipos primitivos o referencias.

Variables **locales**: Se definen dentro de un método o más en general dentro de cualquier bloque entre llaves `{ }`. Pueden ser también tipos primitivos o referencias.

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Variables

den ser cualquier conjunto de caracteres numéricos y alfabéticos, excepto algunos caracteres especiales como los operadores o separadores (, . + - * /).

Este conjunto incluye las **palabras reservadas** las cuales tienen un significado especial para **Java** y que no se pueden utilizar como nombres de variables.

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

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70





Tipos de datos

un lenguaje fuertemente tipado.

Existen dos tipos de datos que se pueden asignar a las variables.

Tipos Simples: Definidos por el lenguaje

Tipos Referencia: Apuntan a objetos por medio de sus posiciones de memoria.



Tipos de datos simples

Tamaño	Descripción
1 byte	Valores true y false
2 bytes	Comprende el código ASCII
1 byte	Valor entero entre -128 y 127
2 bytes	Valor entero entre -32768 y 32767
4 bytes	Valor entero entre -2.147.483.648 y 2.147.483.647
8 bytes	Valor entre -9.223.372.036.854.775.808 y 9.223.372.036.854.775.807
4 bytes	De -3.402823E38 a -1.401298E-45 y de 1.401298E-45 a 3.402823E38
8 bytes	De -1.79769313486232E308 a -4.94065645841247E-324 y de 4.94065645841247E-324 a 1.79769313486232E308

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 --
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Tipos de datos simples

boolean no es un valor numérico: sólo admite los valores **true** o **false**. El resultado de la expresión lógica que aparece como condición en un bucle o en una bifurcación debe ser **boolean**.

char contiene caracteres en código UNICODE (que incluye el código ASCII), y ocupan 16 bits por carácter.

byte, **short**, **int** y **long** son números enteros que pueden ser positivos o negativos, con distintos valores máximos y mínimos.

Los tipos **float** y **double** son valores reales con 6-7 (float) y 15 (double) cifras decimales.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Tipos de datos simples

Conversiones de tipos

Implícitas: Se realizan de modo automático **de un tipo a otro de más precisión**, por ejemplo de **int a long**, de **float a double**, etc.

```
byte a=40;
byte b=50;
byte c= 100;
int d=a * b / c;
```

a * b podría superar el rango definido para un byte. Java promociona automáticamente el resultado a int

Explícitas: Son las conversiones de un tipo de mayor a otro de menor precisión. A estas conversiones explícitas se les llama **cast**.

```
int a;
byte b;
b = (byte) a;
```

No se puede convertir un tipo numérico a **boolean**.

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 --
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Tipos de datos referencia

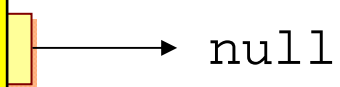
referencia es una variable que indica dónde está en la memoria del ordenador un objeto.

Al declarar una referencia todavía no se encuentra "apuntando" a ningún objeto en particular, luego se le asigna el valor **null**.

Si se desea que esta **referencia** apunte a un nuevo objeto es necesario utilizar el operador **new**. Este operador reserva en la memoria del ordenador espacio para ese objeto (variables y constantes).

También es posible igualar la **referencia** declarada a un objeto existente previamente.

```
Punto p1;
```



```
p1=new Punto();
```



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 --
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Tipos de datos especiales

```
ay {
static void main(String args[]) {
month_days[];
days = new int[12];
days[0] = 31;
days[1] = 28;
days[2] = 31;
days[3] = 30;
days[4] = 31;
days[5] = 30;
days[6] = 31;
days[7] = 31;
days[8] = 30;
days[9] = 31;
days[10] = 30;
days[11] = 31;
m.out.println("April has " + month_days[3] + " days.");
}
```



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

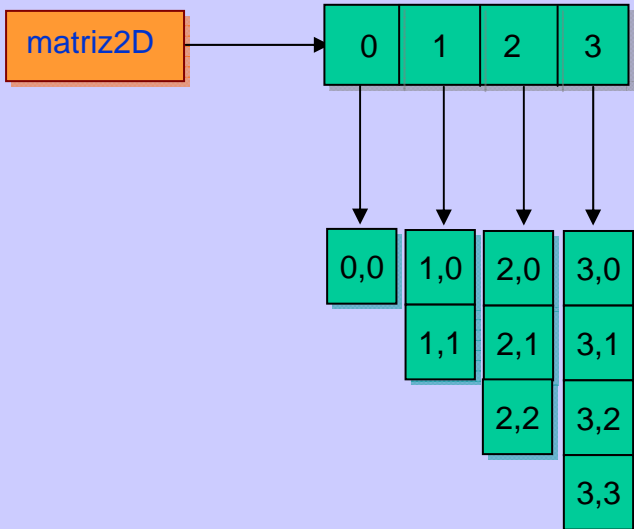


Tipos de datos especiales

```

dimensiones {
public void main(String args[]) {
    matriz2D[][] = new int[4][];
    matriz2D[0] = new int[1];
    matriz2D[1] = new int[2];
    matriz2D[2] = new int[3];
    matriz2D[3] = new int[4];
    for (int i = 0; i < 4; i++)
        for (int j = 0; j < i + 1; j++) {
            matriz2D[i][j] = k;
            k++;
        }
    for (int i = 0; i < 4; i++) {
        for (int j = 0; j < i + 1; j++)
            System.out.print(matriz2D[i][j] + " ");
        System.out.println();
    }
}
}

```



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70



Tipos de datos especiales

enas de caracteres.

String es el tipo cadena.

No es un tipo de datos simple.

Tampoco es un array de char.

Es un objeto con sus atributos, métodos y constructores.

String str = "esto es una cadena";

String str = new String ("esto es una cadena");



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 -- --
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Visibilidad de las variables

entiende por **visibilidad** o **scope** de una variable, la parte de la aplicación donde dicha variable es accesible.

Todas las variables deben estar incluidas en una clase.

Las variables declaradas dentro de unas llaves **{ }**, es decir dentro de un **bloque**, son visibles y existen dentro de estas llaves.

Las variables declaradas al principio de una función existen mientras se ejecute la función

Las variables declaradas dentro de un bloque **if** no serán válidas al finalizar las sentencias correspondientes a dicho **if**

Las variables miembro de una **clase** (es decir declaradas entre las llaves **{ }** de la clase pero fuera de cualquier método) son válidas mientras existe el objeto de la clase.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 --
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Operadores

OPERADORES ARITMÉTICOS

Los operadores aritméticos (operadores binarios que requieren siempre dos operandos) que realizan las operaciones aritméticas usuales .

Nombre	Representación
Suma	+
Resta	-
Producto	*
División	/
Resto de la div.	%

OPERADORES DE ASIGNACIÓN

Los operadores de asignación permiten asignar un valor a una variable .

Operador	uso	equivalente
+=	op1 += op2	op1= op1 + op2
-=	op1 -= op2	op1= op1 - op2
*=	op1 *= op2	op1= op1 * op2
/=	op1 /= op2	op1= op1 / op2
%=	op1 %= op2	op1= op1 % op2

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Operadores

nombre	uso	equivalente
preincremento	<code>++ i</code>	<code>i = i + 1</code>
predecremento	<code>-- i</code>	<code>i = i - 1</code>
postincremento	<code>i ++</code>	<code>i = i + 1</code>
postdecremento	<code>i --</code>	<code>i = i - 1</code>

OPERADORES INCREMENTALES
Incrementan o decrementan el
valor de una variable en una
unidad.

```
a = 5;
b = 2;
a = b ++;
```

Después de la ejecución

a = 2 y b = 3

```
b;
de la ejecución
y b = 3
```

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



Operadores

Operador	uso	nombre
>	op1 > op2	Mayor
>=	op1 >= op2	Mayor o igual
<	op1 < op2	Menor
<=	op1 <= op2	Menor o igual
==	op1 == op2	Igual
!=	op1 != op2	distinto

OPERADORES RELACIONALES

Operadores de comparaciones de igualdad y desigualdad y menor o mayor.

Cualquiera de estos operadores siempre un booleano.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Operadores

Nombre	Acción
AND	True si los dos son true
OR	True si uno de los dos es true
NOT	Lo contrario
AND en cortocircuito	True si los dos son true. Si el primero es false no se evalúa el segundo
OR en cortocircuito	True si uno de los dos es true. Si el primero es true no se evalúa el segundo

OPERADORES LÓGICOS.

Se utilizan para construir expresiones lógicas, combinando valores lógicos (true y/o false) o resultados de los operadores relacionales.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Operadores

OPERADOR CONDICIONAL.

Se utiliza para realizar bifurcaciones condicionales sencillas.

expresionBooleana ? res1 : res2

Se evalúa expresionBooleana y se devuelve **res1** si el resultado es true y **res2** si el resultado es false.

(x < y) ? x+3 : y+8;

Si x es mayor a z el valor 4, es decir x+3



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 --
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Conceptos generales

DECLARACIONES O EXPRESIONES

Una **expresión** es un conjunto de variables unidas por operadores.

Una **sentencia** es una expresión que acaba en punto y coma (;). Se permite incluir varias sentencias en una línea

`i = 0; j = 5; x = i + j;`

COMENTARIOS

Comentario de una línea `//`

Comentario de varias líneas `/*`

.....

..... */

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Bifurcaciones

Esta estructura permite ejecutar un conjunto de sentencias en función del valor que tenga la expresión de comparación (se ejecuta si la expresión de comparación tiene valor true).

```
if (ExpresiónBooleana) {
    sentencias;
}
```

Las llaves { } sirven para agrupar en un bloque las sentencias que se van a ejecutar. No son necesarias si sólo hay una sentencia dentro del bloque.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Bifurcaciones

else

Análoga a la anterior, de la cual es una ampliación. Las sentencias incluidas en el **else** se ejecutan en el caso de no cumplirse la expresión de comparación (**false**).

```
if (ExpresiónBooleana) {
    sentencias1;
} else {
    sentencias2;
}
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Bifurcaciones

```
o = 61; // "numero" tiene dos dígitos
Math.abs(numero) < 10) // Math.abs() calcula el valor absoluto. (false)
m.out.println("Numero tiene 1 digito ");
Math.abs(numero) < 100) // Si numero es 61, estamos en este caso
m.out.println("Numero tiene 2 digitos ");
// Resto de los casos
m.out.println("Numero tiene mas de 2 digitos ");
m.out.println("Se ha ejecutado la opcion por defecto ");
}
```

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
--
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



Bifurcaciones

```
(char)(Math.random()*26+'a'); // Generación aleatoria de letras
// minúsculas

t.println("La letra " + c );
) {
case 'a': // Se compara con la letra a
case 'e': // Se compara con la letra e
case 'i': // Se compara con la letra i
case 'o': // Se compara con la letra o
case 'u': // Se compara con la letra u

System.out.println(" Es una vocal ");

break;
default:

System.out.println(" Es una consonante ");
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Bucles

Las sentencias se ejecutan mientras la **Expresión booleana** sea **true**.

```
while (Expresión booleana ) {  
    sentencias;  
}
```



Bucles

```
i = 1, j = i + 10; i < 5; i++, j = 2*i) {  
System.out.println(" i = " + i + " j = " + j);
```

la salida

```
= 1 j = 11  
= 2 j = 4  
= 3 j = 6  
= 4 j = 8
```



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



Bucles

while

Es similar al bucle **while** pero con la particularidad de que el control está al final del bucle (lo que hace que el bucle se ejecute al menos una vez, independientemente de que la condición se cumpla o no).

```
do {
    sentencias
} while (Expresión booleana );
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Sentencias especiales

Sentencias **break** y **continue**

La sentencia **break** es válida tanto para las bifurcaciones como para los bucles. Hace que se salga inmediatamente del bucle o bloque que se está ejecutando sin finalizar el resto de las sentencias.

La sentencia **continue** se utiliza en los bucles (no en bifurcaciones). Finaliza la iteración "i" que en ese momento se está ejecutando (no ejecuta el resto de sentencias que hubiera hasta el final del bucle). Vuelve al comienzo del bucle y comienza la siguiente iteración (i+1).

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



Sentencias especiales

Sentencia **return**

Es otra forma de salir de un bucle o de un método. A diferencia de **continue** o **break**, la sentencia **return** sale también del método o función.

En el caso de que la función devuelva alguna variable, este valor se deberá poner a continuación del **return** (**return value;**).

que **try {...} catch {...} finally {...}**

Se utilizan en la gestión de excepciones

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



CLASES



Cartagenag9

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



Conceptos básicos

una clase es una agrupación de **datos** (variables o campos) y **funciones** (métodos) que operan sobre esos datos.

```
[public] class Classname {
    // definición de variables y métodos
    ...
}
```

La palabra **public** es opcional: si no se pone, la clase sólo es visible para las demás clases del **package**. Todos los métodos y variables deben ser definidos dentro del **bloque** {...} de la clase.

Un **objeto** (instancia) es un ejemplar concreto de una clase. Las **clases** son como tipos de variables, mientras que los **objetos** son como variables concretas de un tipo determinado.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Conceptos básicos

una clase proporciona una definición para un tipo particular de objeto: define sus datos y la funcionalidad sobre dichos datos.

una clase es una plantilla.

se crea una copia de esa clase por programa pero muchos objetos de esa clase en el mismo programa.

los métodos definen las operaciones que se pueden realizar, esto es lo que hace la clase. no pueden existir fuera de las clases.

todas las variables y funciones de Java deben pertenecer a una clase. No hay variables y funciones globales.

Java tiene una jerarquía de clases estándar de la que pueden derivarse las clases que crean los usuarios.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Conceptos básicos

En Java no hay herencia múltiple. Si al definir una clase se especifica de qué clase deriva, por defecto la clase deriva de Object.

La clase Object es la base de toda la jerarquía de clases en Java.

En un fichero se pueden definir varias clases, pero en un fichero no puede haber más que una clase public. El nombre del fichero se debe llamar como la clase public que contiene con extensión ***.java**. Con algunas excepciones, lo habitual es escribir una sola clase por fichero.

Las clases se pueden agrupar en paquetes, introduciendo una línea al comienzo del fichero (`package nombrepaquete;`).

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



Ejemplo de clase

```
class Punto {
    public double x;
    public double y;
    public Punto(double a, double b) {
        x=a;
        y=b;
    }
    public double calcularDistancia() {
        double z;
        z=Math.sqrt((x*x)+(y*y));
        return z;
    }
}
```

atributos

constructor

Clase Punto.class

método

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Elementos de una clase. Atributos

atributos definen la estructura de los objetos que encierran una clase.

Un **objeto** que se crea de una clase tiene **su propia copia** de los atributos y de los métodos. Por ejemplo, cada objeto de la clase **Punto** tiene sus propias coordenadas **x** e **y**.

Para acceder al valor de un atributo se escribe:

Objeto.atributo;

```
p1.x=5;
p1.y=7;
System.out.println(p1.x+" , "+p1.y);
```

Para aplicar un método a un objeto concreto :

Objeto.método();

```
double d = p1.calcularDistancia();
System.out.println("distancia: "+d);
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Elementos de una clase. Métodos

métodos son funciones definidas dentro de una clase. Los métodos **static** o de clase, se aplican siempre a un objeto de la clase por medio del **operador punto** (.).

El primer objeto es su **argumento implícito**. Esto implica que se puede hacer referencia a sus atributos dentro del método. De estas formas, también se puede acceder a ellas mediante la referencia **this**.

Los métodos pueden además tener otros **argumentos explícitos** que van entre paréntesis, a continuación del nombre del método.

La primera línea de la definición de un método se llama **declaración**.

El **valor de retorno** puede ser un valor de un **tipo primitivo** o una **referencia**.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70



Elementos de una clase. Métodos

uede haber más que un único valor de retorno .

uede devolver como valor de retorno un objeto de la ma clase que el método o de una sub-clase, pero nunca de super-clase.

métodos pueden definir **variables locales**. Su visibilidad rá limitada al propio método. Las variables locales no se alizan por defecto.

```

public Punto elPuntoMasLejano(Punto c) {
    int a;
    if (this.x >= c.x)
        return this;
    else
        a++; //error: a no está inicializada
    return c;
}

```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Elementos de una clase. Constructores

Un **constructor** es un método que se llama automáticamente a la vez que se crea un objeto de una clase. La principal función del **constructor** es reservar memoria e inicializar las variables miembro de la clase.

Se usan para la inicialización correcta de objetos.

Los **constructores** no tienen valor de retorno (ni siquiera `void`) y se tienen que llamar igual que la clase.

Una clase puede tener **varios constructores**, que se diferencian por el tipo y número de sus argumentos (**sobrecargados**).

Se llama **constructor por defecto** al constructor que no tiene argumentos. Cuando no hay constructor, el compilador genera uno por defecto.

Los **constructores** sólo pueden ser llamados por otros **constructores** o por métodos **static**. No pueden ser llamados por los **métodos de objeto** de la clase.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

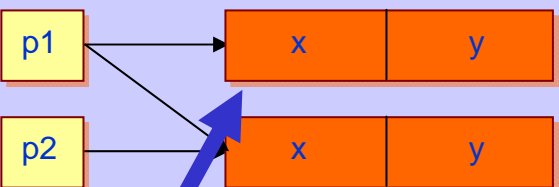
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Elementos de una clase. Destruectores

```
=new Punto(2,3);
```

```
=new Punto(5,7);
```



**Garbage
collector**

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

Sobrecarga

```
class Principal {
    public static void main(String args[]) {
        SobrecargaDemo ob = new SobrecargaDemo();
        double res;
        ob.test();
        ob.test(10);
        ob.test(10, 20);
        result = ob.test(123.2);
        System.out.println(res);
    }
}
```

```
SobrecargaDemo {
    public SobrecargaDemo() {
        System.out.println("vacio");
    }
}
```

```
public void test(int a) {
    System.out.println("a: " + a);
}
```

```
public void test(int a, int b) {
    System.out.println("a y b. " + a + " " + b);
}
```

```
public void test(double a) {
    System.out.println("double a: " + a);
}
return *a;
```



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Sobrecarga de constructores

```

Caja{
    le alto;
    le ancho;
    le largo;
    (double a, double b, double c){
        alto = a;
        ancho = b;
        largo = c;
    }
    {
        alto = -1;
        ancho = -1;
        largo = -1;
    }
    (double longitud) {
        alto = ancho = largo = longitud;
    }
}

```

```

class Principal{
    public static void main(String args[]){
        Caja c1=new Caja(10,20,15);
        Caja c2=new Caja();
        Caja c3=new Caja(5);
        .
        .
    }
}

```

Diagram illustrating constructor overloading in the `Caja` class. The `Principal` class contains a `main` method that creates three `Caja` objects: `c1` (with parameters 10, 20, 15), `c2` (with no parameters), and `c3` (with parameter 5). Green arrows point from these object creation calls to the corresponding constructor definitions in the `Caja` class: `c1` to the 3-parameter constructor, `c2` to the no-parameter constructor, and `c3` to the 1-parameter constructor.

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 --
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Paso de objetos como parámetros

Objetos también pueden pasar a un método como parámetros, ya son variables del tipo definido por su clase.

```

Caja{
    le alto;
    le ancho;
    le largo;
    ( double a, double b, double c ) {
        alto = a;
        ancho = b;
        largo = c;
    }
    double ob ) {
        alto = ob.alto;
        ancho = ob.ancho;
        largo = ob.largo;
    }
}

```

```

class Principal{
    public static void main(String args[]){
        Caja cl=new Caja(10,20,15);
        Caja cajaclonada=new Caja(cl);
        .
        .
        .
    }
}

```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



so de variables por valor y por referencia

variables de
arán de difer
o de origen
Por valor:

```
class Principal{
    public static void main(String args[]){
        Test ob=new Test();
        int a=15;
        int b=20;
        S.o.p("a y b antes:"+a+" "+b);
        ob.metodo(a,b);
        S.o.p("a y b depues:"+a+" "+b);
    }
}
```

```
Test {
    metodo( int i, int j ){
        i=i*2;
        j=j/2;
    }
}
```

a y b antes: 15 20
a y b después: 15 20

Las operaciones que se realizan sobre tipos de datos simples no afectan a las variables

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



so de variables por valor y por referencia

```
class Principal{
    public static void main(String args[]){
        Test ob=new Test(15,20);
        S.o.p("ob.a y ob.b antes:"+ob.a+" "+ob.b);
        ob.metodo(ob);
        S.o.p("ob.a y ob.b despues:"+ob.a+" "+ob.b);
    }
}
```

Por referenc

```
Test{
    a,b;
    ( int i, int j ){
        a=i;
        b=j;
    }
    metodo( Test ob ){
        ob.a *=2;
        ob.b /=2;
    }
}
```

ob.a y ob.b antes: 15 20
ob.a y ob.b después: 30 10

Las operaciones que se realizan sobre objetos
 si afectan a los atributos de estos

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Control de acceso

```

class Test {
    // acceso por defecto
    int b; // acceso publico
    private int c; // acceso privado

    setc(int i) {
        c = i;
    }

    getc() {
        return c;
    }
}

```

pueden estar precedidos por

```

class Acceso {
    public static void main(String args[]) {
        Test ob = new Test();

        ob.a = 10;
        ob.b = 20;

        ob.c = 100; // Error!

        ob.setc(100); // OK

        System.out.println("a, b, y c: " +
            ob.a + " " + ob.b + " " + ob.getc());
    }
}

```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



static

```
oStatic {
    int c;
    static int a = 42;
}

ncipal {
    static void main(String args[]) {
        DemoStatic ob1=new DemoStatic();
        DemoStatic ob2=new DemoStatic();
        ob1.c =5;
        ob1.a=7;
        System.out.println("ob1.c = " + ob1.c);
        System.out.println("ob1.a = " + ob1.a);
        ob2.c =15;
        ob2.a=10;
        System.out.println("ob2.c = " + ob2.c);
        System.out.println("ob2.a = " + ob2.a);
        System.out.println("ob1.a = " + ob1.a);
        System.out.println("a = " + DemoStatic.a);
    }
}
```



--

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



static

```
emoStatic {
    static int a = 42;
    static int b = 99;

    c {
        System.out.println("cargando clase...");
        b = a * 4;
    }

    c void llamada() {
        System.out.println("a = " + a);
    }
}

Principal {
    c static void main(String args[]) {
        System.out.println("antes de llamar al metodo");
        DemoStatic.llamada();
        System.out.println("b = " + DemoStatic.b);
    }
}
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



final

utiliza para declarar variables de manera que su contenido no pueda ser modificado.

La variable **final** tiene que ser inicializada en su declaración.

La variable **final** se convierte en una constante.

Por convenio, el identificador de una variable **final**, debe escribirse en mayúsculas.

```
final int NUEVO_ARCHIVO=3;
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



HERENCIA



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

```
class Principal {  
    public static void main(String args[]) {  
        A superOb = new A();  
        B subOb = new B();  
        superOb.i = 10;  
        superOb.j = 20;  
        System.out.println("superOb: ");  
        superOb.mostrarij();  
        System.out.println();  
  
        subOb.i = 7;  
        subOb.j = 8;  
        subOb.k = 9;  
        System.out.println("subOb: ");  
        subOb.mostrarij();  
        subOb.mostrark();  
        System.out.println();  
  
        System.out.println("suma en subOb:");  
        subOb.sum();  
    }  
}
```

```
        mostrarij() {  
            y j: + i + " " + j);
```

```
        ds A {
```

```
        rk() {  
            " + k);
```

```
        {  
            j+k: " + (i+j+k));
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



```
{
;

extends A {
;

a, int b) {

    a;
    super.i = a;
    b;

mostrar() {
    tem.out.println("i de la superclase: " + super.i);
    tem.out.println("i de la subclase: " + i);

principal {
    c static void main(String args[]) {
        subOb = new B(1, 2);
        Ob.mostrar();
```

¿a que i hace referencia?

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

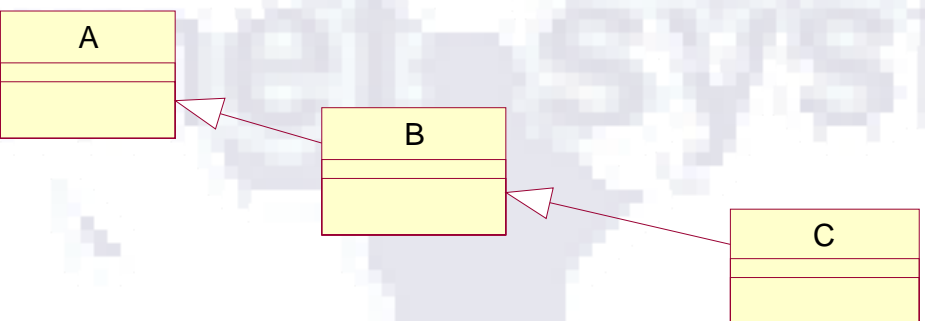




Herencia

Jerarquía multinivel:

La herencia se puede definir en varios niveles: Si B hereda de A y C hereda de B => C hereda de A y de B





Referencia. Orden de ejecución de los constructores

```

System.out.println(" constructor de A");

class B
extends A {
    B() {
        System.out.println(" constructor de B");
    }
}

class C
extends B {
    C() {
        System.out.println("constructor de C");
    }
}

public class Principal {
    public static void main(String args[]) {
        new C();
    }
}

```

La salida que genera es:

```

constructor de A
constructor de B
constructor de C

```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



Herencia. Sobreescritura de métodos

```

{
    j;
    a, int b) {
        a; j = b;

    mostrar() {
        System.out.println("i y j: " + i + " " + j);
    }
}

class B extends A {
    B(int a, int b, int c) {
        super(a, b);
        k = c;
    }

    mostrar() {
        System.out.println("k: " + k);
    }
}

```

```

class Principal{
    public static void main(String args[]) {
        B subOb = new B(1, 2, 3);
        subOb.mostrar(); }
}

```

Se ejecuta el mostrar de B que sobreescrive al de A.

Para llamar al mostrar de A se usa la instrucción:

```
super.mostrar();
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70



Herencia. Polimorfismo

```
s {
1;
2;
uble a, double b) {
;
;
a() {
ut.println("Area no
");
;
gulo extends Figuras {
(double a, double b) {
b);
a() {
ut.println("Area del
lo.");
dim1 * dim2;
```

```
class Triangulo extends Figuras {
    Triangulo(double a, double b) {
        super(a, b);
    }
    double area() {
        System.out.println("Area del triángulo.");
        return dim1 * dim2 / 2;
    }
}

class Principal{
    public static void main(String args[]) {
        Figuras f = new Figuras(10, 10);
        Rectangulo r = new Rectangulo(9, 5);
        Triangulo t = new Triangulo(10, 8);
        Figuras fig;
        fig = r;
        System.out.println("Area :" + fig.area());
        fig = t;
        System.out.println("Area :" + fig.area());
        fig = f;
        System.out.println("Area :" + fig.area());
    }
}
```



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Herencia. Clases abstractas

```
public class A {
    abstract void llamada() ;
}

class B implements A {
    noAbstracto() {
        System.out.println("Este es un método no abstracto.");
    }

    llamada() {
        System.out.println("Implementación del método abstracto");
    }
}

public class Principal {
    public static void main(String args[]) {
        B b = new B();
        b.llamada() ;
        b.noAbstracto();
    }
}
```



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



PAQUETES

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



Cartagena99





Paquetes

Los contenedores de clases que se utilizan para mantener el espacio de nombres de clases dividido en departamentos.

Definición:

Se pone antes de definir la clase, e indica el paquete al que pertenece la clase. Si ya está creado la añade, si no lo crea.

```
package p1;
```

```
package p1.p2;
```

Ejemplo:

```
import p1.*;
```

```
import p1.p2.*;
```

```
import p1.Clase;
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Paquetes

paquetes tienen importancia para la accesibilidad a una e.

Identificador de visibilidad **protected** permite mostrar miembros de una clase fuera del paquete actual pero sólo a las subclases directas de su clase.

acceso	private	Sin modificador	protected	public
se	Si	Si	Si	Si
del mismo paquete	No	Si	Si	Si
se del mismo paquete	No	Si	Si	Si
de diferente paquete	No	No	Si	Si
se de diferente paquete	No	No	No	Si



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



Cartagenag9

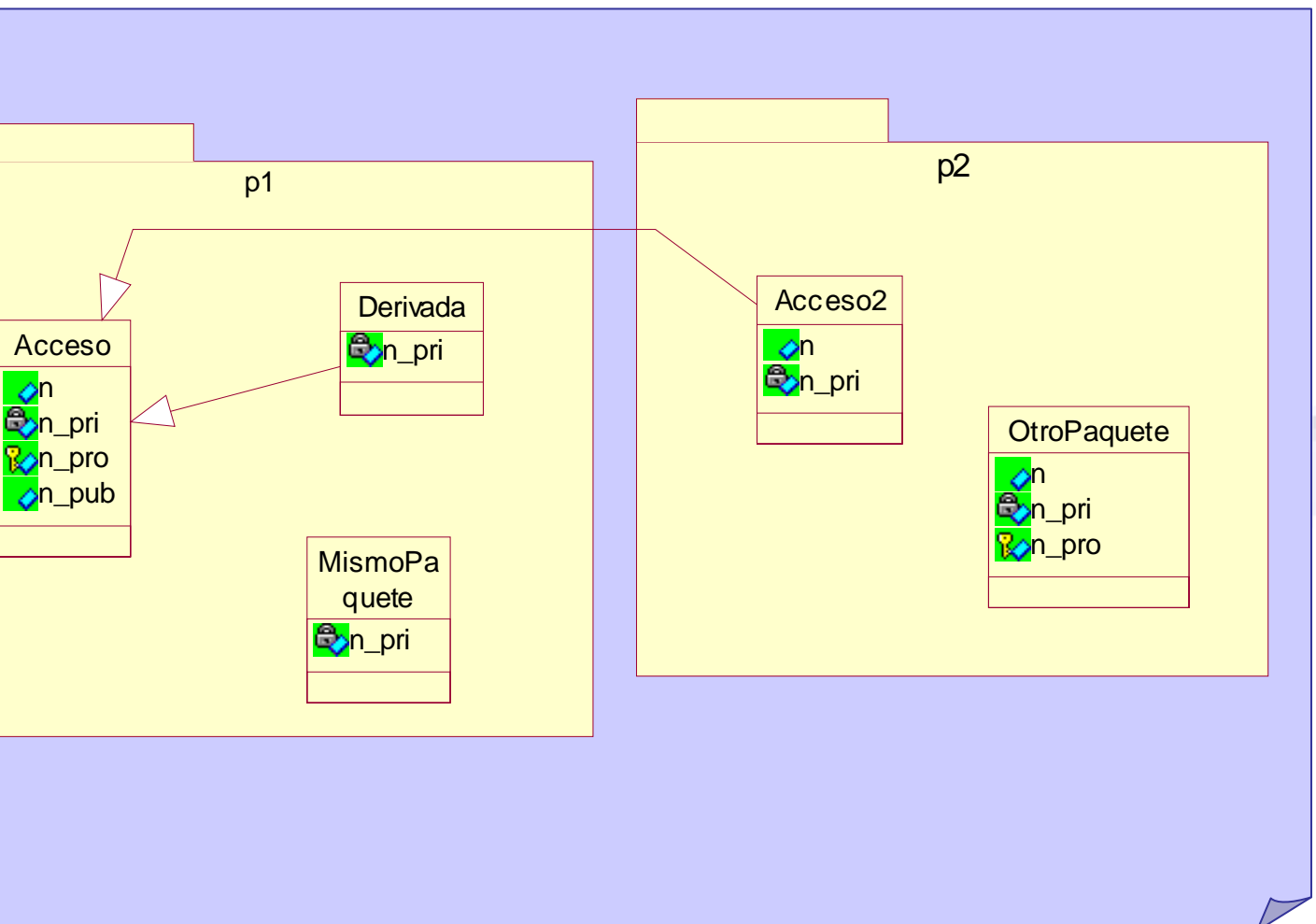
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



Paquetes. Ejemplo



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



INTERFACES



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



Interfaces

Interface son estructuras de código similares a las clases que contienen la definición de un conjunto de funcionalidades, pero no su implementación.

Las clases que implementen un interface están obligadas a implementar todos los métodos de interface.

Un interface no puede contener métodos implementados.

Un interface no se puede instanciar.

```
Llamada{
    llamar(int param);
}
```

```
class Cliente implements Llamada{
    public void llamar(int p){
        S.o.p("Llamando a llamar"+p);
    }
}
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Interfaces

clase puede implementar varios interfaces

class Cliente implements Llamada, Llamada2{....}

métodos de las clases que implementan un interface
deben ser públicos.

```
class Principal{
    public static void main(String args[]){
        llamada c=new Cliente( );
        c.llamar(5);
    }
}
```

se pueden declarar objetos a través de una variable de
referencia a un interface. *

variables de un interface son constantes dentro de la clase
que implementa al interface.

interfaces se pueden heredar.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Cartagena99

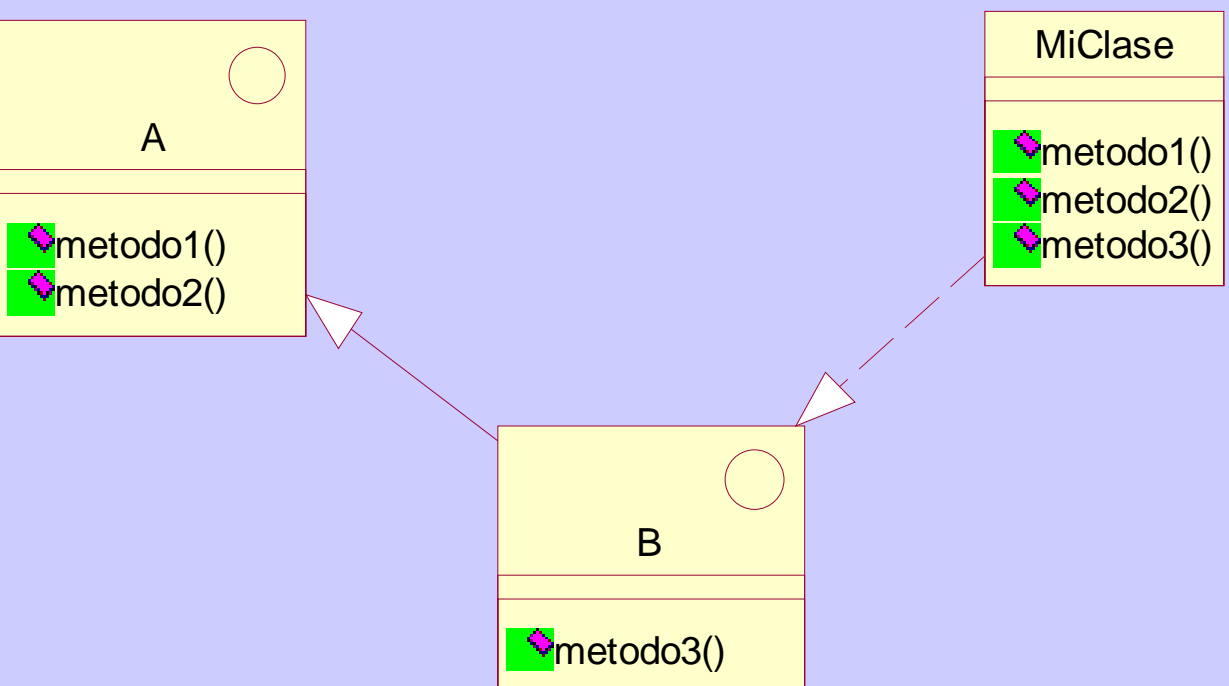
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



Interfaces



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



EXCEPCIONES



Cartagenag9

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

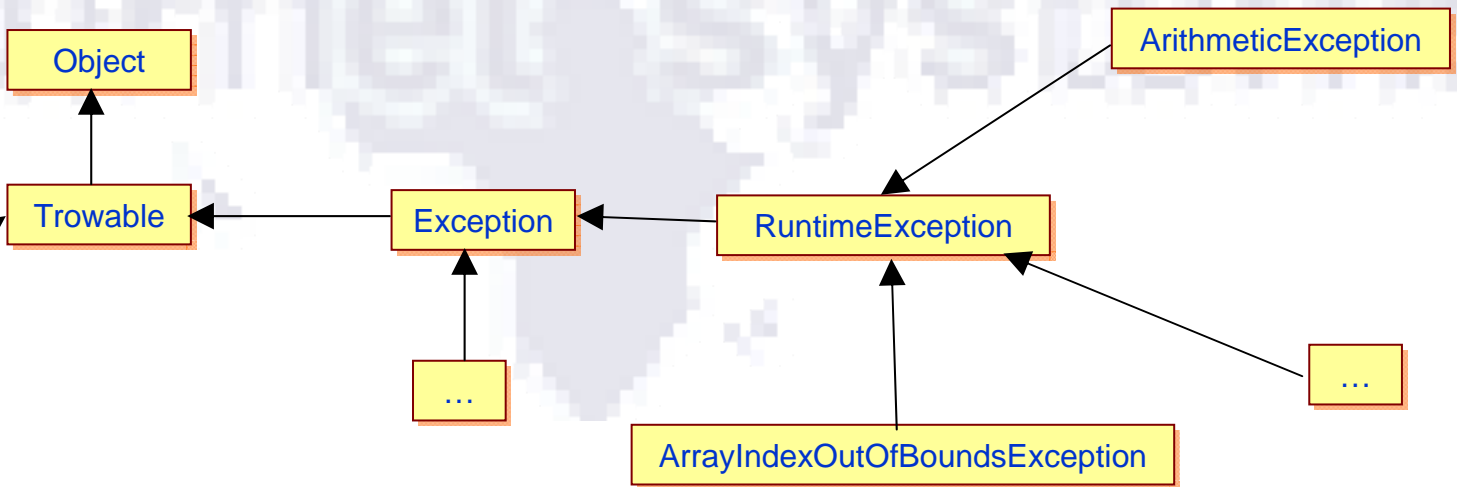




Excepciones

Una excepción es un error que se produce en tiempo de ejecución.

Una excepción es un objeto. Todos estos objetos son instancias de las clases de la clase **Throwable**, y se ajustan a una estructura jerárquica.





Excepciones

El control de de código de error se realiza con los bloques **try – catch**

Las zonas de código susceptibles de generar errores deben estar acotadas en un bloque **try**

Cuando se produce un error en el bloque **try** se genera un objeto del tipo de la excepción que se ha producido

Esta excepción se puede *recoger* en un bloque **catch**

El ámbito del bloque **catch** está restringido a las instancias del bloque **try** que le precede inmediatamente.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Excepciones

```

Excepcion {
    static void main(String args[]) {
        int d;
        int a;
        a = 42 / d;
        System.out.println("Mensaje");
    }
}

```

¡¡ERROR!!

```

class Excepcion {
    public static void main(String args[]) {
        int d, a;

        try {
            d = 0;
            a = 42 / d;
            System.out.println("Mensaje.");
        } catch (ArithmeticException e) {
            System.out.println("Division por cero.");
        }
        System.out.println("Despues de catch.");
    }
}

```

EXCEPCION

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



Excepciones

Si no se conoce el tipo de excepción que se puede producir, es aconsejable utilizar el objeto genérico `Exception`.

Existen métodos para obtener información sobre el objeto `Exception` (`e`). Estos y otros métodos pertenecen a la clase `Throwable`.

`toString()` Devuelve una descripción de la excepción

`printStackTrace()` Devuelve una traza

```
e.printStackTrace();
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



Excepciones. Try anidados

```

idados {
static void main(String args[]) {
{
a = args.length;
b = 42 / a;
System.out.println("a = " + a);
try {
if(a==1) a = a/(a-a);
if(a==2) {
int c[] = { 1 };
c[42] = 99;
}
} catch(ArrayIndexOutOfBoundsException e) {
System.out.println("Desbordamiento: " + e);
}
} catch(ArithmeticException e) {
System.out.println("Division por 0: " + e);
}
}
}

```

- Si **a=0**, se genera una división por 0 y lo captura el catch externo
- Si **a=1**, genera división por 0. Como no puede capturarlo el catch interno, lo captura el externo.
- Si **a=2**, se produce un desbordamiento que lo captura el catch interno

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Excepciones

```
demo {
    void demoproc() {
        {
            throw new NullPointerException("demo");
        }
    }

    catch(NullPointerException e) {
        System.out.println("captura dentro de demo.");
        throw e;
    }
}

    static void main(String args[]) {
        {
            demoproc();
        }
    }

    catch(NullPointerException e) {
        System.out.println("Nueva captura: " + e);
    }
}
```

Cartagena99

--

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



Cartagenag9

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



PROGRAMACIÓN MULTITHILO



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



MULTIHILO. INTRODUCCIÓN

gramación multihilo es un estilo de ejecución en el que se muta entre distintas partes del código de un mismo programa durante la ejecución.

Una de estas partes individuales se llama Thread (hilos, hilos o contextos de ejecución)

Una VM los ejecuta de forma simultanea en una maquina multiprocesador y de forma conmutada en maquina de un solo procesador ,esto es, la ejecución de los threads se realizará conmutando un determinado quantum o espacio de tiempo a favor de uno de los threads.

Los hilos permiten ejecutar dos tareas al mismo tiempo por el mismo procesador. Los hilos son procesos ligeros

La conmutación entre hilos es mucho mas rápida que la conmutación entre procesos

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



MULTIHILO. INTRODUCCIÓN

En la actualidad casi todos los sistemas operativos proporcionan servicios para la programación multihilo.

La JVM de todas las plataformas proporcionan hilos de ejecución aunque el sistema operativo subyacente no los proporcione, en este caso la JVM simula ese comportamiento.

A partir de la versión 1.3 de Java, la JVM para entornos Windows proporciona servicios de hilos nativos (esto es proporcionados por la API Win32 y no simulados por el sistema). A parte de estos hilos nativos la JVM en todo momento sigue proporcionando hilos simulados por ella misma (no nativos) conocidos como hilos verdes (**green threads**).

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



MULTIHILO. INTRODUCCIÓN

Existen dos formas de crear hilos en Java. Un hilo es una clase que descende de la clase `java.lang.Thread` o bien que extiende a la interfaz `Runnable` (util en los casos de la clase ya forme parte de una jerarquía de clases).

La forma de construcción del hilo (derivación o implementación) es independiente de su utilización.

En Java un hilo puede encontrarse en cualquiera de los siguientes estados:

inactivo: El hilo se ha declarado pero todavía no se ha dado el orden de puesta en ejecución. (`método start()`).

listo: El hilo está preparado para entrar en ejecución pero el planificador aún no ha decidido su puesta en marcha.

ejecutándose: El hilo está ejecutándose en la CPU.

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



MULTIHILO. INTRODUCCIÓN

dormido: El hilo se ha detenido durante un instante de tiempo definido mediante la utilización del método `sleep()`.

bloqueado: El hilo está pendiente de una operación de I/O y no volverá al estado listo hasta que esta termine.

suspendido: El hilo ha detenido temporalmente su ejecución mediante la utilización del método `suspend()` y no la reanuda hasta que se llame a `resume()`. *Antiguo, 1.2.No debe usarse.*

esperando: El hilo ha detenido su ejecución debido a una condición externa o interna mediante la llamada a `wait()` y no reanuda esta hasta la llamada al método `notify()` o `notifyAll()`.

muerto: El hilo ha terminado su ejecución bien porque terminó de realizar su trabajo o bien porque se llamó al método `stop()`. *Antiguo 1.2.No debe usarse.*

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



CREACIÓN DE HILOS

lo puede crearse extendiendo a la clase **Thread** o implementando a la interfaz **Runnable**. En ambos casos la clase debe proporcionar una definición del método `run()` que contiene el código a ejecutarse una vez que el hilo entre en estado de ejecución

```
class HiloThread extends Thread{
    public void run(){
        while(true){
            System.out.println("Hola mundo, soy el hilo HiloTread");
        }
    }
}
```

```
class HiloRunnable implements Runnable{
    public void run(){
        while(true){
            System.out.println("Hola mundo, soy el hilo HiloRunnable");
        }
    }
}
```

¿Como los pongo en marcha y como interactúo con ellos?.



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 --
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Ejecución de hilos

```
class Test{
    public static void main(String[] args){
        //Creamos un hilo del primer tipo
        //y lo ponemos en marcha

        HiloThread ht = new HiloTread();
        ht.start();

        //Creamos un hilo del segundo tipo
        //y lo ponemos en marcha

        Thread hr = new Thread(new HiloRunnable());
        hr.start();
    }
}
```

Ambos hilos se construyen de manera distinta: mientras que el que desciende de Thread se crea como un objeto más, el que implementa Runnable, se crea y se pasa al constructor de la clase genérica Thread.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70





Constructores de Thread

La clase Thread tiene los siguientes constructores:

Thread()

Thread(Runnable target)

Thread(Runnable target, String name)

Thread(String name)

Thread(ThreadGroup group, Runnable target)

Thread(ThreadGroup group, Runnable target, String name)

Thread(ThreadGroup group, String name)

String name se usa para dar un nombre al thread de manera que luego pueda referenciarse de forma externa mediante los métodos String **getName()** y void **setName(String)**



String name se usa para dar un nombre al thread de manera que luego pueda referenciarse de forma externa mediante los métodos String **getName()** y void **setName(String)**

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Algunos métodos de Thread

live() determina si un hilo está vivo o si se ha terminado de ejecutar.

final boolean isAlive() throws InterruptedException{ }

- Devuelve true si el hilo que llama al método está vivo.

wait() espera hasta que termine de ejecutarse el hilo que llama al método.

final void join() throws InterruptedException{ }

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Algunos métodos de Thread

`suspend()` suspende la ejecución de un hilo hasta que se vuelva a reanudar (con `resume`)

```
final void suspend();
```

`resume()` reanuda un hilo que previamente ha sido suspendido.

```
final void resume();
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 --
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Ejemplo

```
class NuevoHilo implements Runnable {
    String name;
    Thread t;
    public NuevoHilo(String nombre) {
        name = nombre;
        t = new Thread(this, nombre);
        System.out.println("Nuevo hilo: " + t);
        t.start();
    }
    public void run() {
        try {
            for (int i = 5; i > 0; i--) {
                System.out.println(name + ":" + i);
                Thread.sleep(1000);
            }
        } catch (InterruptedException e) {
            System.out.println("interrupción en el hilo: " + name);
        }
    }
}
```



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
--
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



Ejemplo

```

class Principal {
public static void main(String[] args) {
    NuevoHilo ob1=new NuevoHilo("Uno");
    NuevoHilo ob2=new NuevoHilo("Dos");
    NuevoHilo ob3=new NuevoHilo("Tres");
    System.out.println("¿El hilo Uno esta vivo?: "+ob1.t.isAlive());
    System.out.println("¿El hilo Dos esta vivo?: "+ob2.t.isAlive());
    System.out.println("¿El hilo Tres esta vivo?: "+ob3.t.isAlive());

    try {
        System.out.println("esperando a que terminen los otros");
        ob1.t.join();
        ob2.t.join();
        ob3.t.join();
    } catch (Exception e) {
        System.out.println("Interrupcion hilo principal");}
    System.out.println("¿El hilo Uno esta vivo?: "+ob1.t.isAlive());
    System.out.println("¿El hilo Dos esta vivo?: "+ob2.t.isAlive());
    System.out.println("¿El hilo Tres esta vivo?: "+ob3.t.isAlive());
    System.out.println("Sale del principal");
}
}

```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Prioridades de los hilos

Cada hilo tiene asociada una prioridad representada por un número entero.

La prioridad determina cuando se debe ejecutar un hilo antes que a otro, y no la velocidad de ejecución.

Cuando un hilo deja de ejecutarse para ejecutarse otro, se produce un **cambio de contexto** => al detenerse un hilo, se analizan los siguientes hilos. El que tiene una prioridad más alta, se ejecuta.

Las prioridades de un hilo pueden oscilar entre 1 y 10. La prioridad por defecto es 5.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

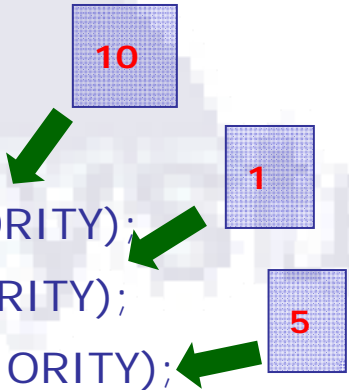


Prioridades de los hilos

Priority() asigna prioridad a un hilo

```
final void setPriority(int nivel){ }
```

- Hilo.setPriority(2);
- Hilo.setPriority(Thread.MAX_PRIORITY);
- Hilo.setPriority(Thread.MIN_PRIORITY);
- Hilo.setPriority(Thread.NORM_PRIORITY);



Priority() devuelve la prioridad actual de un hilo

```
final int getPriority(){ }
```

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



¡¡OJO CON LAS PRIORIDADES!!

```
public class Contador implements Runnable{
    int click=0;
    Thread t;
    private boolean running=true;
    public Contador(int p){
        t=new Thread(this);
        t.setPriority(p);
    }
    public void run(){
        while (running){
            click++;
        }
    }
    public void stop(){
        running=false;
    }
    public void start(){
        t.start();
    }
}
```



El siguiente ejemplo lanzan dos hilos que cuentan el número de clics que realiza un usuario durante un tiempo. Los dos hilos ejecutan el código en distintas prioridades. Al final se muestra en pantalla el resultado. ¿Cuál será? El resultado va a ser diferente dependiendo de la plataforma en que se ejecute.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



¡¡OJO CON LAS PRIORIDADES!!

```

class Principal {
    public static void main(String[] args) {
        Thread.currentThread().setPriority(Thread.MAX_PRIORITY);
        Contador hi=new Contador(Thread.NORM_PRIORITY+2);
        Contador lo=new Contador(Thread.NORM_PRIORITY-2);
        lo.start();hi.start();
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            System.out.println("interrupcion");}
        lo.stop();hi.stop();
        try {
            hi.t.join();
            lo.t.join();
        } catch (InterruptedException e1) {
            System.out.println("captura");}
        System.out.println("hilo con prioridad baja: "+lo.click);
        System.out.println("hilo con prioridad alta: "+hi.click);
    }

```

hilo con prioridad baja: 0
hilo con prioridad alta: 138989761

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



INTRODUCCIÓN A LAS APLICACIONES WEB



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

-- --

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



INTRODUCCIÓN A INTERNET

I: Una red es una agrupación de computadoras mediante la cual se posibilita el intercambio de información de un modo eficiente y transparente.

LAN: (Local area network) Red de área local. Se reduce a los límites de un edificio.

MAN: (Metropolitan area network) Metropolitana. Varios edificios dentro de una misma área urbana.

WAN: (Wide area network) Una centros dispersos en áreas geográficas muy amplias. Suele estar integrada por varias redes mas pequeñas. Internet se puede considerar la WAN mas grande del mundo.

W.W.: (World Wide Web) Denominación que se da a Internet – La gran telaraña mundial – Internet es una red de redes.

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 --
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70





INTRODUCCIÓN A INTERNET

TCP/IP

Para que los ordenadores que componen una red puedan comunicarse es necesario que todos hablen en el mismo "idioma". A esto se denomina un **protocolo**.

Un **protocolo** es un conjunto reglas e instrucciones comunes a todos los ordenadores de la misma red.

Internet utiliza el protocolo **TCP/IP**.

TCP (Transport Control Protocol)

IP (Internet protocol)

Es un sistema de comunicaciones basado en ensamblado y desensamblado de paquetes.



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

--



INTRODUCCIÓN A INTERNET

servicios

Sobre la base de comunicación del protocolo TCP/IP existen otros protocolos más específicos que permiten realizar tareas particulares:

SMTP: correo electrónico.

TELNET: conexiones y ejecución de comandos con máquinas remotas.

FTP: transmisión de ficheros

NNTP: acceso a foros de discusión y news

HTTP: conexión a servidores web

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70





INTRODUCCIÓN A INTERNET

P

Permite solicitar datos a un servidor remoto para visualizarlos localmente (en un navegador) utilizando TCP/IP.

El protocolo **HTTP** no es permanente y es necesario mantener la conexión desde el servidor mediante el estado de la sesión.

Normalmente, en una aplicación cliente-servidor, el cliente va a hacer peticiones a páginas web. Estas peticiones se suelen hacer por medio de una **URL**.



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



INTRODUCCIÓN A INTERNET

Una URL es el nombre completo de un recurso.

Todo ordenador que use internet tiene su propia dirección electrónica (**IP**).

Una IP está compuesta por una secuencia de cuatro números de un máximo de tres cifras cada uno y separados por puntos:

194.224.71.2

Para que resulte más fácil recordar los identificadores no se usa la IP, sino un sistema de nomenclatura paralelo denominado **DNS**.

Una URL hace referencia a un recurso (archivo, directorio de correo, ...) ubicado en una máquina remota identificada mediante su DNS.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 --
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70





Cartagenag9

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



INTRODUCCIÓN A INTERNET

...s para HTTP

La sintaxis completa de una URL para HTTP es:

://servidor:puerto/ruta del fichero

Ejemplo:

http://localhost:8100/euro/conversor.html

En este caso estamos haciendo una petición a una aplicación ubicada en un servidor web cuyo puerto de escucha es el 8100.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



ARQUITECTURA CLIENTE-SERVIDOR

tendencias actuales en aplicaciones C-S

Añadir inteligencia al cliente

- JavaScript
- Applets

Añadir inteligencia al servidor

- Servlets
- Jsp



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
-- --
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



APPLETS



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



APPLETS

Un applet es un pequeño programa que se transmite al cliente junto con una aplicación web y que se instalan y ejecutan en el cliente.

Un applet es una clase compilada que se envía al cliente junto con el documento HTML solicitado en una petición.

Un applet se ejecuta (interpreta) en el cliente (navegador) por medio de su JVM.

Los applets permiten dotar al cliente de funcionalidad para interactuar con la aplicación.

Las aplicaciones web que contienen applets son lentas.



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



APPLETS

Un applet debe heredar de la clase Applet.

La clase Applet está contenida en el paquete

java.applet

La clase java.applet define tres interfaces

AppletContext

AppletStub

AudioClip

Una clase que define un applet debe importar el paquete

```
import java.applet.*;
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



APPLETS

Para permitir que una applet tenga una interface gráfica de visualizarse debe importar el paquete.

```
import java.awt.*;
```

La ejecución de un applet **no** comienza con main().

La salida de información de un applet **no** se realiza con el método System.out.print().

Un applet es una clase que hay que compilar. Una vez generado el archivo XXX.class, hay que hacer una referencia a este desde un documento HTML con la etiqueta

```
<APPLET>
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



APPLETS

Referencia a un applet:

```

<HTML>
<HEAD>
</HEAD>
<BODY>
  Esto es una referencia a un applet
  <APPLET CODE="ejemploapplet.Miapplet" WIDTH=200
  HEIGHT=60>
</APPLET>
</BODY>
</HTML>

```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 --
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



ALGUNOS METODOS GRAFICOS

void drawString(String mensaje, int x, int y)

Permite mostrar cadenas de caracteres en un applet

Pertenece a Graphics

mensaje: La cadena que se va a mostrar

x,y : coordenadas de la posición donde se va a mostrar

void setBackground(Color nuevocolor)

Asigna color de fondo a la ventana del applet

nuevocolor: Objeto de tipo Color que define los colores mediante constantes de la clase

- Color.black, Color.blue, Color.orange, ...

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



ALGUNOS METODOS GRAFICOS

void setForeground(Color nuevocolor)

Color de primer plano.

Color getBackground()

Devuelve el color de fondo actual.

Color getForeground()

Devuelve el color de primer plano actual.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



EJEMPLO

```
<HTML>
<HEAD>
</HEAD>
<BODY>
  Esto es una referencia a un applet <br>
  <APPLET CODE="ejemploapplet.Miapplet" WIDTH=500 HEIGHT=300>
</APPLET>
</BODY>
</HTML>
```

```
ejemploapplet;
java.awt.*;
java.applet.*;
class Miapplet extends Applet{
  String msg;

  public void init() {
    setBackground(Color.orange);
    setForeground(Color.blue);
    msg="dentro de init() -- ";
  }

  public void paint(Graphics g) {
    msg+="Dentro de paint() -- ";
    g.drawString(msg,10,30);
  }

  public void start() {
    msg+="dentro de start() -- ";
  }
}
```

- crear la clase Miapplet
- compilar y exportarla a una carpeta llamada **practicaapplet**

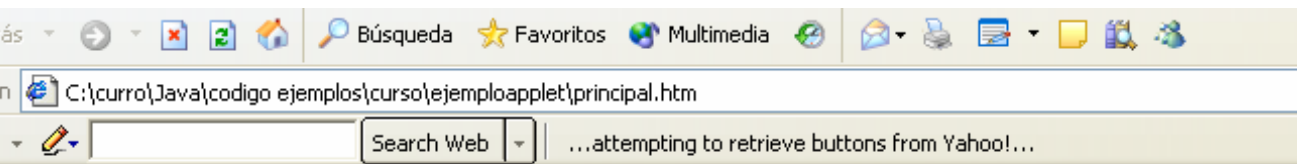
- Crear el documento HTML y guardarlo con el nombre **principal.htm** en la carpeta **practicaapplet**.
- Ejecutar el documento **principal.htm**

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



RESULTADO



es una referencia a un applet

ro de init() -- dentro de start() -- Dentro de paint() --

--

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 --
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

Cartagena99





METODOS

método repaint

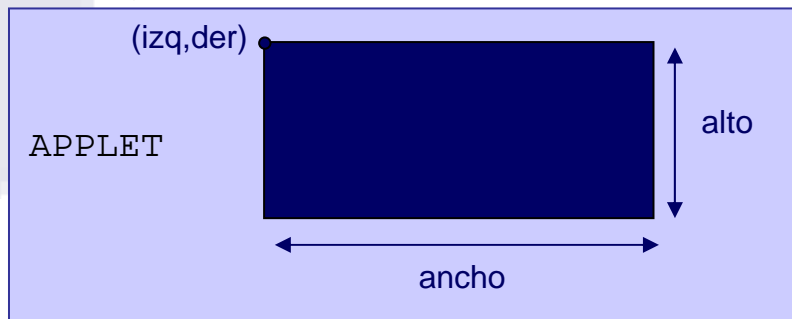
public void repaint()

public void repaint(int izq, int sup, int ancho, int alto)

Realiza una llamada al método paint() y así poder actualizar el contenido del applet.

izq, sup: coordenadas de origen de la zona que se quiere actualizar

ancho, alto: dimensiones de esa zona.



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



EJERCICIO

er un applet que sea capaz de mostrar un texto
plazándose horizontalmente (banner)

esitas saber:

String.charAt(int n) → Devuelve el carácter que esta en la
posición n de un String

String.substring(int n, int n2) → Devuelve un String con
n2 caracteres cogidos a partir de la posición n del String
que llama al método.

Para producir un retardo en un bucle:

- **Thread.sleep(250)**
- Para poder utilizar la clase Thread la clase debe implementar la interface **Runnable**.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70
 --
 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



SOLUCIÓN

```

let.*;
.*;
anner extends Applet
ments Runnable {
    - un cartel en movimiento - ";
l;
destroy() {
    !=null){
stop();
null;

init() {
ckground(Color.orange);
reground(Color.blue);
    Thread(this);
rt();
pend();

```

```

public void paint (Graphics g){
    g.drawString(msg,50,30);
}
public void run() {
    char ch;
    for(int i=1;i<200;i++ ){
        try{
            repaint();
            t.sleep(250);
            ch=msg.charAt(0);
            msg=msg.substring(1,msg.length());
            msg+=ch;
        }
        catch(InterruptedException e){}
    }
}
public void start() {
    t.resume();
}
public void stop() {
    t.suspend();
}
}

```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



VENTANA DE ESTADO

`showStatus(String cadena)`

Se utiliza para mostrar un mensaje en la barra de estado

```

AppletEstado;
import java.awt.*;
import java.applet.*;

public class VentanaEstado extends Applet{

    void init() {
        setBackground(Color.cyan);
    }

    void paint(Graphics g) {
        g.drawString("Esto es un applet.", 10, 20);
        showStatus("Esto es un mensaje en la barra de estado.");
    }
}

```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



GESTION DE EVENTOS

Los eventos que se generan en un applet están encapsulados dentro de un objeto **Event**.

Event pertenece a AWT.

Event define varias variables que describen el evento:

`x`, `y` : posición del ratón cuando se produce el evento de ratón.

`key`: almacena la tecla pulsada.

Event contiene métodos que procesan los eventos más usuales.

Al sobrescribir en un applet algunos de estos métodos, se indica indicando que hará el applet cuando se produzca dicho evento.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70



EJEMPLO

C:\curro\Java\codigo ejemplos\curso\naves\naves

```

naves;
import java.awt.*;
import java.applet.*;

class Naves extends Applet {
    private Image normalUfo,ufo,bonusUfo;
    int rx=0,ry=0;
    public void init(){
        normalUfo = getImage(getCodeBase(), "Images/Ufo1.gif");
        bonusUfo = getImage(getCodeBase(), "Images/Ufo3.gif");
        setBackground(Color.orange);
        setForeground(Color.blue);}

    public boolean mouseDown (Event evento, int x, int y){
        rx=x;
        ry=y;
        ufo=bonusUfo;
        repaint();
        return true;}

    public boolean mouseDrag (Event evento, int x, int y){
        showStatus("Arrastrando. Posición del ratón: "+x+", "+y);
        rx=x;
        ry=y;
        ufo=bonusUfo;
        repaint();
        return true;}
}

```





EJEMPLO continuación

```

public boolean mouseEnter (Event evento, int x, int y){
    showStatus("Entrando. Posición del ratón: "+x+", "+y);
    rx=x;ry=y;ufo=normalUfo;
    repaint();
    return true;}

public boolean mouseExit (Event evento, int x, int y){
    showStatus("Saliendo. Posición del ratón: "+x+", "+y);
    rx=x;ry=y;ufo=normalUfo;
    repaint();
    return true;}

public boolean mouseMove (Event evento, int x, int y){
    showStatus("Posición del ratón: "+x+", "+y);
    rx=x;ry=y;ufo=normalUfo;
    repaint();
    return true;}

public boolean mouseUp (Event evento, int x, int y){
    rx=x;ry=y;ufo=normalUfo;
    repaint();
    return true;}

public void paint (Graphics g){
    g.drawImage( ufo, rx-15, ry-30, this);}

```