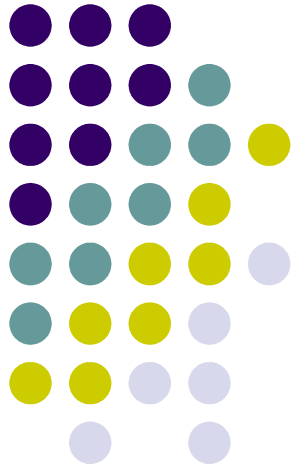
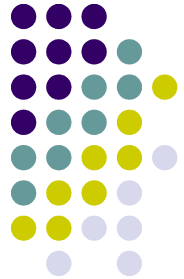
Introducción a Java

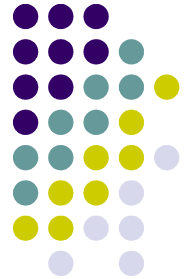


Fundamentos del lenguaje



- Archivos fuente
- Palabras reservadas e identificadores
- Literales
- Arreglos

Fundamentos del lenguaje



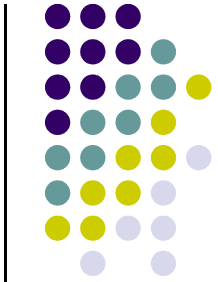
Los archivos tienen extensión .java

Mismo nombre que la clase

Código fuente contiene 3 elementos no requeridos pero con precedencia:

5. Declaración de paquete
6. Imports
7. Definiciones de clase

Fundamentos del lenguaje



Un paquete es una colección de clases con nombre.

Los paquetes sirven para agrupar un conjunto de clases relacionadas y definen un *namespace* para para las clases contenidas.

Para definir el paquete de una clase se utiliza la palabra *package*. Si aparece en el código fuente, debe ser la primera palabra del archivo.

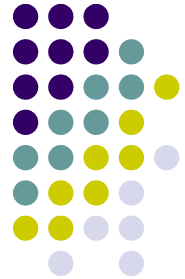
Formato de declaración:

package uno.dos.tres;

Refleja estructura de directorios.

Comúnmente se usa el dominio de la empresa en modo inverso: **com.softtek.curso**

Fundamentos del lenguaje



Una clase en el paquete p puede hacer referencia a cualquier otra clase dentro del mismo paquete simplemente con su nombre.

Si la clase se encuentra en otro paquete, se usa el nombre completo del paquete y la clase: **com.softtek.Clase.**

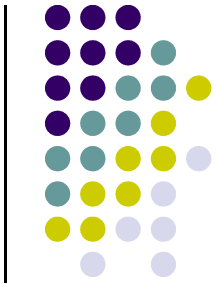
Para simplificar el código se utiliza la palabra *import*.

La palabra import puede hacer referencia a una clase específica o a un conjunto de clases:

import com.softtek.Clase;

import com.softtek.*;

Fundamentos del lenguaje



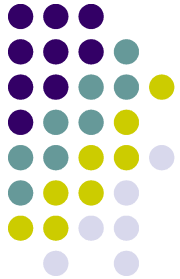
Las clases se definen con la palabra reservada *class.*

```
public class NombreClase { … }
```

Los nombres de variables deben empezar con una letra o los caracteres $ y _.

```
foobar      // bien
ABCdef      // bien
$abcd       // bien
3_abc       // mal
!algo       // mal
```

Fundamentos del lenguaje

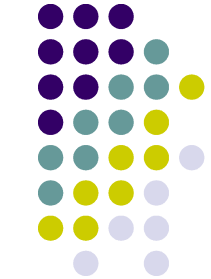


Hay dos tipos de comentarios:

```
int var;      // comentarios de línea

/**
 * comentarios varias líneas
 */
```

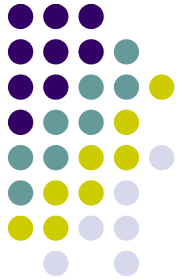
Fundamentos del lenguaje



Tipos de datos o primitivos

boolean
char
byte
long
short
int
long
float
double

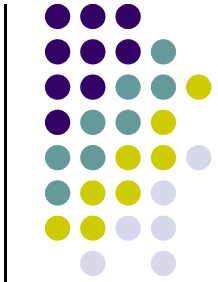
Fundamentos del lenguaje



Tamaño en bits

Tipo	bits	Tipo	bits
boolean	1	char	16
Byte	8	short	16
Int	32	long	64
Float	32	double	64

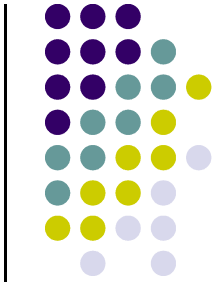
Fundamentos del lenguaje



Enteros con signo

byte
short
int
long

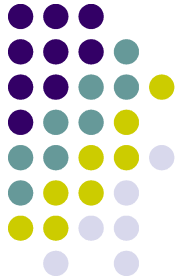
Fundamentos del lenguaje



Rangos de valores

Tipo	Tamaño	Mínimo	Máximo
byte	8	-2.00E+007	2.00E+007 - 1
short	16	-2.00E+015	2.00E+015 - 1
int	32	-2.00E+031	2.00E+031 - 1
long	64	-2.00E+063	2.00E+063 - 1

Fundamentos del lenguaje



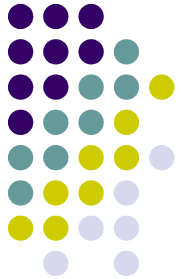
Existen dos tipos de primitivos de punto flotante: *float* y *double*.

Las clases Float y Double definen valores especiales.

Todos los tipos numéricos tienen signo, con excepción de boolean y char, que va de 0 a 2e16 – 1.

Float.NaN
Float.NEGATIVE_INFINITY
Float.POSITIVE_INFINITY
Double.NaN
Double.NEGATIVE_INFINITY
Double.POSITIVE_INFINITY

Literales

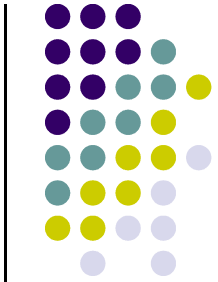


Una literal es un valor especificado en el codigo fuente.

Las literales representan valores primitivos o cadenas de caracteres.

A continuación se describen los siguientes tipos de literales.

Literales



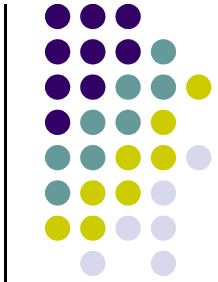
Las literales de tipo boolean pueden tener el valor *true* o *false*.

```
boolean a = true;
boolean b = false;
```

Litearles de caracteres

```
char = 'c';
```

Literales



Para especificar un caracter en UNICODE se usa la siguiente notacion:

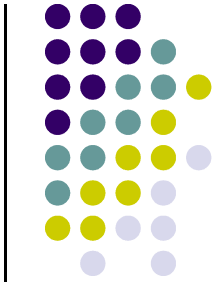
```
char c1 = '\u4567';
```

Existen algunos caracteres especiales que requieren una notación distinta.

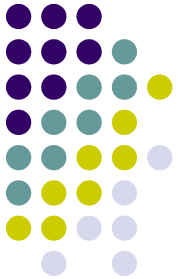
Literales

Caracteres especiales:

'\n' – nueva linea.
'\r' – return.
'\t' – tab.
'\'' - comillas simples.
'\\' - backslash
'\”' - comillas.



Literales



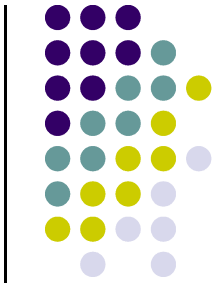
Los valores enteros se expresan en forma decimal, octal y hexadecimal.

Las literales octal llevan el prefijo 0 (cero).

Hexadecimal lleva el prefijo 0x ó 0X.

Los valores pueden ser mayúscula o minúscula.

Literales



Ejemplo. El 28 se puede escribir de las siguientes formas:

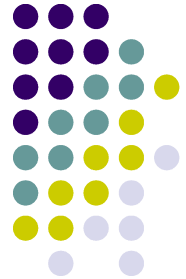
28
034
0x1c
0x1C
0X1c
0X1C

Literales de punto flotante

Una literal de punto flotante expresa un número de punto flotante. Para que una expresión sea interpretada como una literal de punto flotante debe cumplir con alguna de las siguientes condiciones:

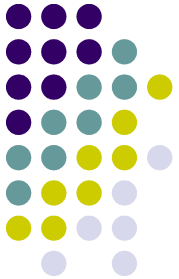
2. Punto decimal: 1.23
3. Tener la letra *E* ó *e* para indicar notacion cientifica: 4.23E+21
4. El sufijo F o f para indicar float: 1.23f
5. El sufijo D o d para indicar double: 1.23d

Literales de punto flotante



Una literal de punto flotante sin el sufijo *F* o *D* es double por defecto.

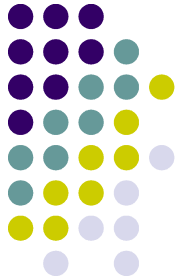
Literales de cadena (String)



Las cadenas de caracteres se representan entre comillas:

String s = “uno dos tres”;

Arreglos

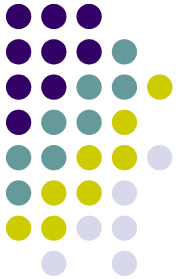


Un arreglo en Java es una colección ordenada de primitivos, referencias a objetos u otro arreglos.

Los arreglos son homogéneos, es decir, todos los elementos deben ser del mismo tipo.

Los elementos en un arreglo tienen un índice numérico.

Arreglos

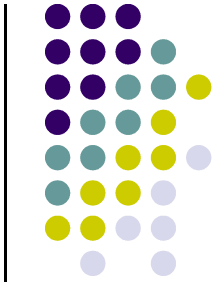


Los arreglos tienen un tamaño fijo.

Para crear un arreglo se deben seguir tres pasos:

4. Declaración
5. Construcción
6. Inicialización

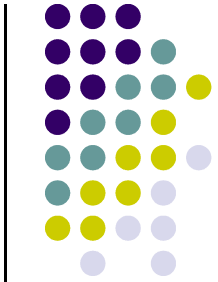
Arreglos



La declaracion le indica al compilador el nombre del arreglo y el tipo de elementos que almacena:

```
int[] abc;
double[] def;
Objeto[] ghi;
float[][] dosDimensiones;
```

Arreglos



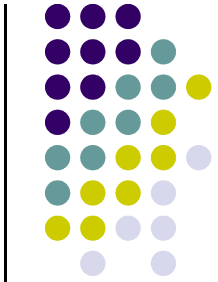
La declaración no especifica el tamaño del arreglo.

El tamaño se especifica al momento de ejecución con la palabra *new*.

```
int [] ints;
ints = new int[20];
```

Cuando un arreglo se construye sus elementos inicializados a sus valores por defecto.

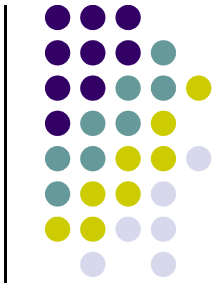
Fundamentos del lenguaje



Tamaño en bits

Tipo	Valor inicial	Tipo	Valor inicial
byte	0	short	0
int	0	long	0L
float	0.0f	double	0.0d
char	‘\u0000’	boolean	false
Referencia de objeto	null		

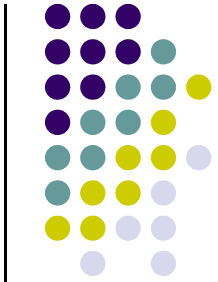
Arreglos



Los arreglos tienen un índice para cada elemento con el rango [0 .. n-1].

```
long[] abc;
abc = new long[100];
for(i=0 ; i<100 ; i++)
{
      abc[i] = i * 3;
}
```

Fundamentos de clases



El método main() es el punto de entrada para una aplicación Java.

Para crear una aplicación es necesario escribir alguna clase con el método main ().

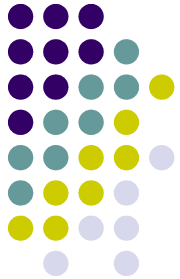
Para ejecutar una aplicación se puede teclear:

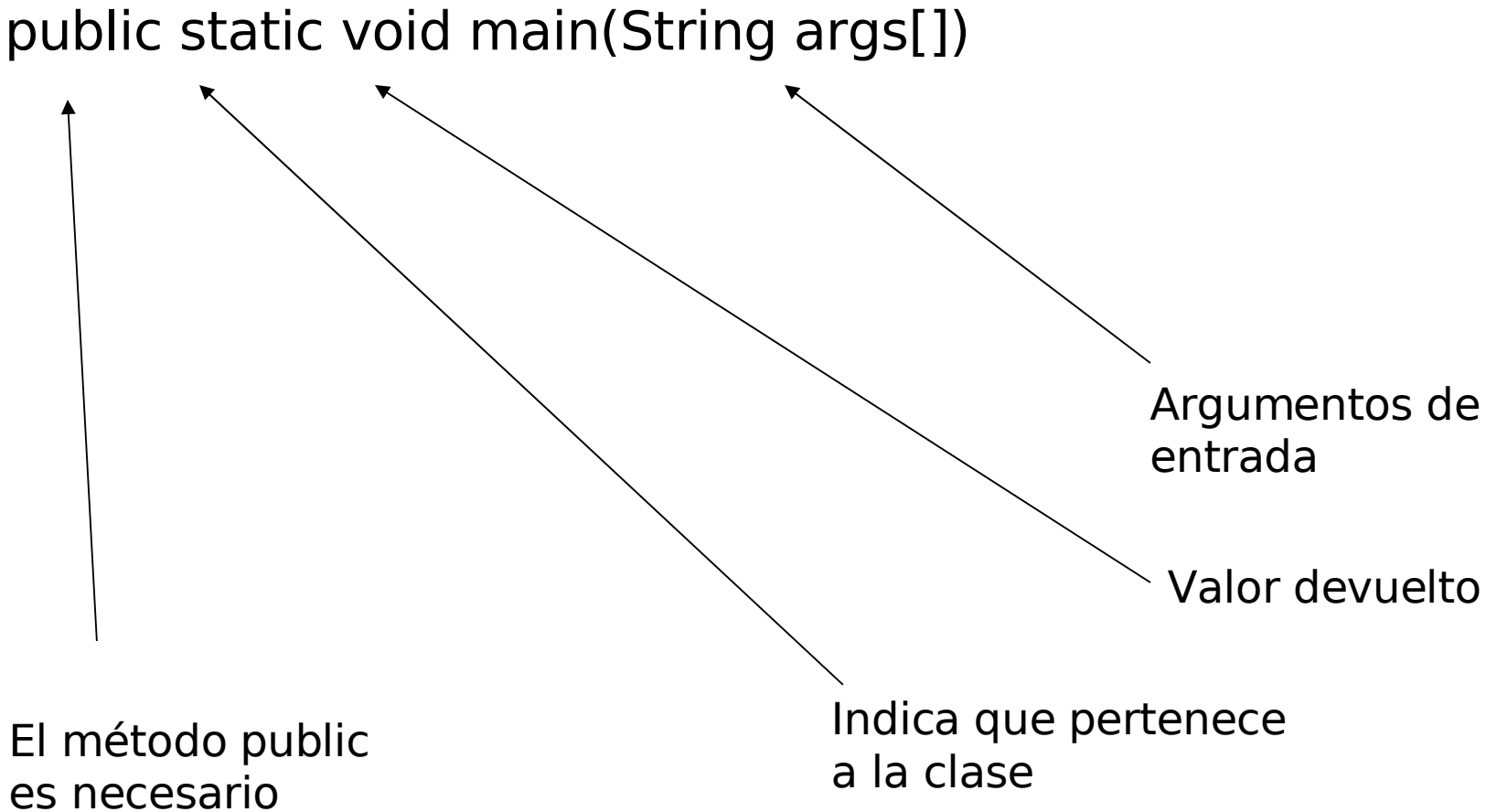
java ClaseConMain

La firma de main() es:

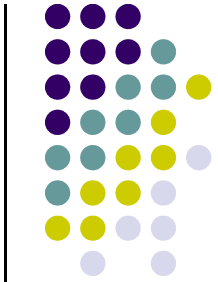
public static void main(String args[])

Fundamentos de clases





Argumentos



Los argumentos que reciben los métodos se pasan por valor (primitivos) o por referencia (objetos).

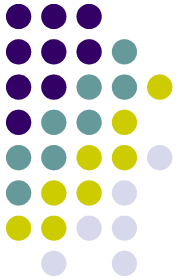
Cuando un argumento es un primitivo se hace una copia del valor.

Los cambios internos a un argumento no surten efecto fuera del método.

En el caso de objetos se pasa una referencia al objeto.

Esta referencia no puede ser modificada, pero sí el objeto.

Recolección de basura



Proceso de baja prioridad.

Parametrizable.

Distintos algoritmos de recolección.

System.gc()

Runtime.gc()