

### Conceptos Basicos

**Clase** Archivo escrito en lenguaje Java que modela atributos y metodos de un Objeto.

**Objeto** Representación de una clase en tiempo de ejecución.

**Estado** Valor de cada uno de los atributos en un momento dado.

**Comportamiento** Funcionalidad que provee una clase, se conforma por los metodos de la misma

**Interfaces** Contratos de comportamiento en POO

### Modificadores de no acceso

Nombre	Metodo	Clase
--------	--------	-------

Strictp	Modifica la precision de la aritmetica flotante.	Modifica la precision de la aritmetica flotante de todos los métodos.
---------	--	---

Final	Garantiza que no sea sobrescrito.	Garantiza que la clase no sea extensible.
-------	-----------------------------------	---

Abstract	Permite omitir la implementación del método.	Permite a la clase tener 1-N métodos abstractos
----------	--	---

Native	Implementacion dependiente de plataforma	
--------	--	--

Synchronized	Garantiza que solo un hilo accedera al metodo en todo momento	
--------------	---	--

### Inclusión estatica.

**Función** Provee de un atajo para no escribir los nombres completamente cualificados para acceder a MIEMBROS estaticos

**Syntaxis** static import \*.\*.MiembroEstatico

**Ej.** import static java.lang.System.out;

### Asignación de literales enteras

**Binarias** Comienzan con 0b ó 0B

**Octales** Comienzan con 0

**Hexadecimal** Comienzan con 0x ó 0X

Toda literal no fraccionaria es entera por defecto.

Todo calculo entre literales ES ENTERO por defecto si al menos un operando es de tipo entero.

Se aplica cast implicito a tipos de dato con menor tamaño de palabra

### Asignación de literales Flotantes

DEBEN terminar con f ya que toda literal fraccionaria es doble por defecto Ej.10.05f

### Identificadores legales

Identificador de variable legal

Un identificador legal tiene las siguientes características:

Debe comenzar por una letra o un simbolo(\$,\_) , a partir del segundo caracter se permitira el uso de números.

### Compilador de java

**Función** Transformar archivos escritos en lenguaje Java(.java) en bytecode que la JVM puede entender y ejecutar (.class)

**Uso** javac [opciones] [archivos fuente]

### Modificadores de acceso

Nombre	Funcion	Clase	Metodo
--------	---------	-------	--------

public	Provee acceso a todas las clases.	Si	Si
--------	-----------------------------------	----	----

protected	Provee acceso dentro del mismo paquete y a subclases en caso de existir	No	Si
-----------	---	----	----

default	Provee acceso solo dentro del mismo paquete	Si	Si
---------	---	----	----

private	Garantiza el acceso solo dentro de la misma clase	No	Si
---------	---	----	----

Se encuentran ordenados de mas permisivo a menos permisivo

### Asignación de literales caracter

Unicode	"\uxxx"
---------	---------

Caracter	'caracter'
----------	------------

Numerico	Numero representable en 2 bytes
----------	---------------------------------

### Declaracion de Clases

Debe cumplir con las siguientes reglas:

a) De tener una clase pública el nombre del archivo DEBE coincidir con la misma.

b) Si la clase pertenece a un paquete este debe estar especificado en la primera linea.

### JVM

Funcion Ejecutar programas una vez compilados.

Uso java [opciones] [Nombre C.principal] [argumentos]

### Reglas de escritura de Interfaces

a) La syntaxis sera la siguiente:

<modificador> <abstract> interface nombre <extends  
int1,int2....>

b) Solo puede contener miembros constantes

<public> <static> <final> nombre

c) Todos sus métodos son implícitamente "public" "abstract" y  
NO permiten  
modificadores de no acceso.

<> Engloban parametros opcionales.

### Metodos de argumentos variables

Ejemplo de Syntaxis void test(int ... t){}

Solo se permite uno por metodo y DEBE ser el último  
argumento.

### Declaracion de arreglos

Ej. arreglo con nombre int a[]=new int[5]; / int a[]={  
1,2,3}

Ej. arreglo con nombre y bloque int a[]=new int[]{1,2}  
inicializador

Ej. arreglo anonimo new int[]{1,2}

### Reglas de oro de la sobrecarga de metodos

Una sobrecarga DEBE modificar los argumentos del método

Una sobrecarga PUEDE modificar tipo de retorno o  
modificador de acceso

