

CURSOS
INTERSEMESTRALES



PROTECO

Manipulación de Datos

Java Básico 2019-1

Recordatorio

Animal
int patas
Animal perro
perro.caminar()
perro.nombre



Abstracción
Encapsulamiento
Herencia
Polimorfismo

Nombre de Variable

- El primer carácter tiene que ser una letra (a, b, c...), el carácter guión bajo (_) o el dólar (\$).
- Los siguientes caracteres pueden ser letras, el guión bajo, el dólar o dígitos (0, 1, 2...).

_123

123-123

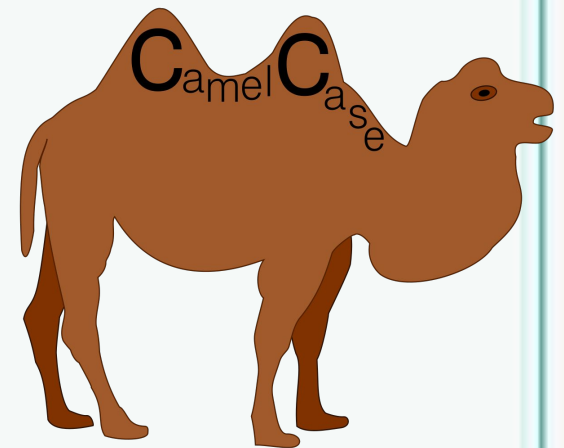
123\$123

\$123

abc

CamelCase

- UpperCamelCase, cuando la primera letra de cada una de las palabras es mayúscula. Ejemplo: *EjemploDeUpperCamelCase*.
- lowerCamelCase, igual que la anterior con la excepción de que la primera letra es minúscula. Ejemplo: *ejemploDeLowerCamelCase*.



Paquete

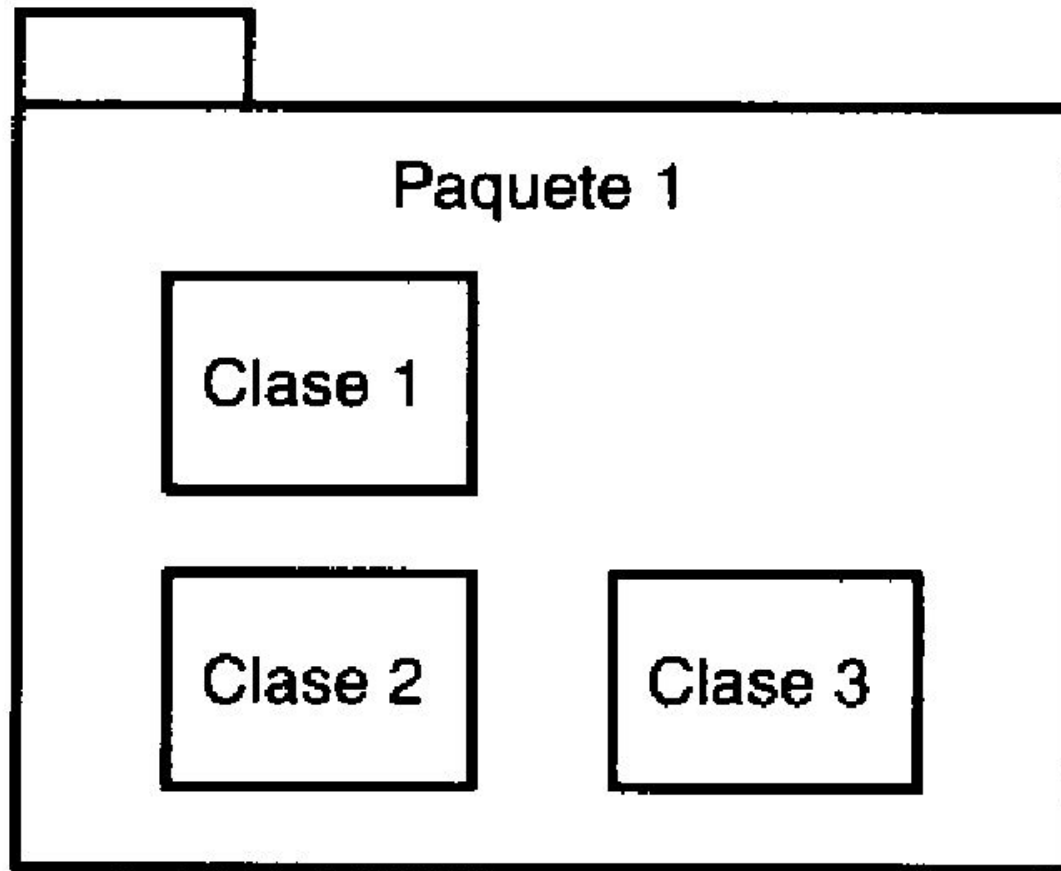
Un paquete es un conjunto de clases, interfaces, enumeraciones, excepciones, errores, anotaciones y otros paquetes que cumplen un determinado objetivo.

Así como las clases siguen la notación UpperCamelCase, los paquetes siguen la siguiente notación:

Todo el nombre va con minúsculas
Para separar palabras se usa puntos.

Por ejemplo: `este.es.un.ejemplo`

Representación



Paquetes Importantes

java.lang: Posee todas las clases necesarias para el funcionamiento de Java, no se requiere importar ninguna clase de este paquete puesto que todas las clases del paquete se importan por defecto. Contiene clases como String, Clases Envolveres, Clase Math, Object

java.util: Posee clases con funciones útiles (aunque no indispensables) para el funcionamiento de Java. Como Scanner, las ArrayList.

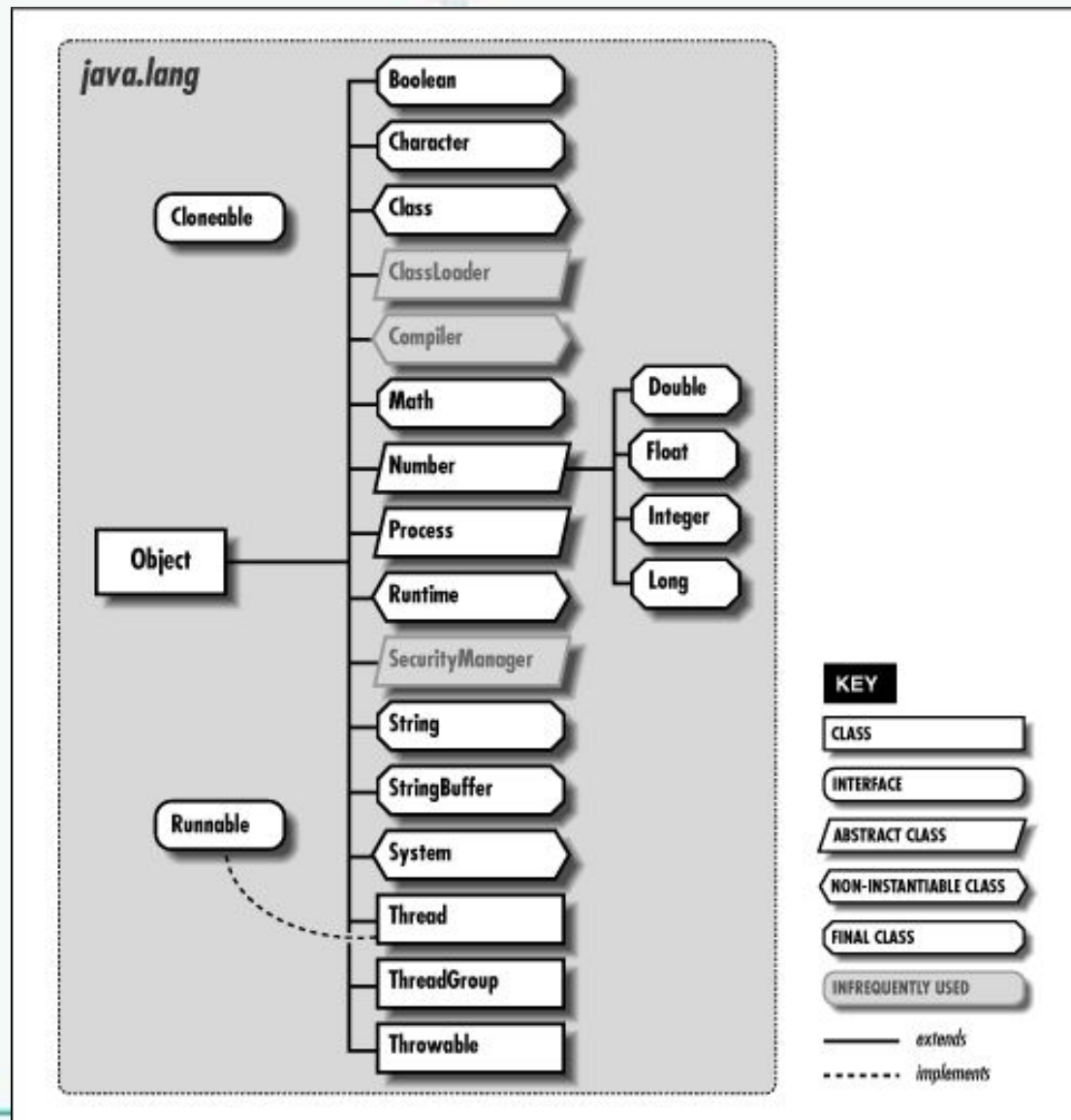
Paquete Importantes

javax.swing: Posee clases para realizar interfaces gráficas.

java.math: Posee clases que permiten trabajar con cualquier entero y con cualquier decimal.

java.io: Posee clases que continen funciones de entrada y salida.

Paquete Lang



Paquetes

Para importar una clase de un paquete se realiza de la siguiente forma (por ejem: importar la clase ArrayList de java.util):

```
import java.util.ArrayList;
```

Para importar todas las clases de un paquete se realiza de la siguiente forma (por ejem: importar todas las clases de java.util):

```
import java.util.*;
```

No es recomendable importar todas las clases de un paquete, porque hace al compilador un poco más lento y genera errores si tenemos dos o más clases con el mismo nombre.



Creación de Paquetes

Podemos crear paquetes usando la palabra reservada ``package nombre.del.paquete;``

De esta forma indicamos que estamos en un paquete llamado "nombre.del.paquete"

Todo lo que queramos usar (clases, interfaces, excepciones, etc.) que no se encuentre en nuestro paquete hay que importarlo.

Si no definimos un paquete se crea un paquete llamado ``default``.



Encapsulamiento

Encapsulamiento consiste en la ocultación de los atributos (campos) para que las demás clases (o determinadas) no puedan acceder directamente a ellos. En Java el encapsulamiento se representa con modificadores de acceso.



Modificadores de Acceso

Visibilidad	Public	Protected	Default	Private
Desde la misma Clase	SI	SI	SI	SI
Desde cualquier Clase del mismo Paquete	SI	SI	SI	NO
Desde una SubClase del mismo Paquete	SI	SI	SI	NO
Desde una SubClase fuera del mismo Paquete	SI	SI, a través de la herencia	NO	NO
Desde cualquier Clase fuera del Paquete	SI	NO	NO	NO

Modificadores de Acceso

Visibilidad	Public	Protected	Default	Private
Desde la misma Clase	SI	SI	SI	SI
Desde cualquier Clase del mismo Paquete	SI	SI	SI	NO
Desde cualquier Clase fuera del Paquete	SI	NO	NO	NO



- Subclase hace referencia a conceptos de herencia (se ve posteriormente).
- Los modificadores de acceso se pueden aplicar a métodos, atributos, clases, interfaces, enumeraciones y excepciones.



Modificadores

Un modificador permite realizar un cambio en una variable, método o clase.

final está relacionado con la incapacidad para modificar una variable, método o clase.

static está relacionado con la no instancia de las cosas, mejor dicho, en lugar de la pertenencia al objeto pertenece a la clase.

Modificador: final

Final aplicado a un atributo, permite establecer dicho atributo como constante. Una constante debe poseer su nombre en mayúscula, y de preferencia poseer el modificador de acceso public.

Final aplicado a un método, establece que dicho método no puede ser sobrescrito en una clase hija

Final aplicado a una clase, establece que ninguna clase puede heredar de ella.

Modificador: static

Static aplicado a un atributo, permite que esta sea igual para cualquier instancia de la clase y puede ser accedida (si los modificadores de acceso lo permiten) por otra clase sin la necesidad de crear una instancia.

Static aplicado a un método hace que ese método se pueda usar sin instanciar la clase.

static se puede aplicar a una clase, pero tiene que ser una clase anidada.

Mutabilidad

Mutabilidad hace referencia a el alteramiento de un objeto. Existen dos tipos de objetos en cuanto a su mutabilidad:

* **Objetos Mtables:** Objetos que si se pasan como parámetro de una función ó si se utiliza alguno de sus métodos afectan su contenido.

* **Objetos no Mtables:** Objetos que si se pasan como parámetro de una función ó si se utiliza alguno de sus métodos no pueden afectar su contenido.

Compilación en java

En java para compilar usamos el mando javac, este posee opciones adicionales:

`javac [opciones adicionales] [nombre de archivo .java]`

las opciones adicionales más importantes son:

- d [ubicación] esta opción (bandera) nos permite establecer donde queremos guardar los .class
- cp [ubicacion] esta bandera nos permite establecer dónde se encuentra algunos .class que necesitamos

Compilación en Java

Por ejemplo, si deseamos compilar la class Ejemplo.java y queremos guardar la clase en una carpeta superior:

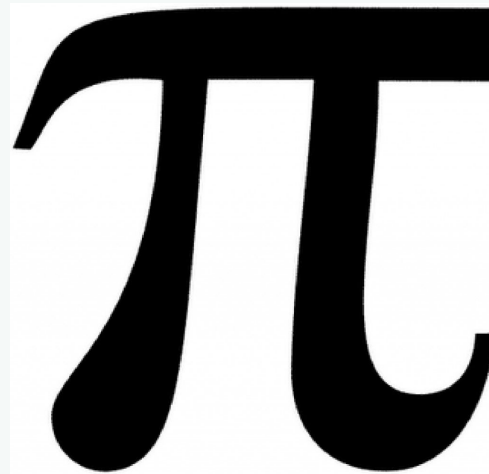
```
javac -d .. Ejemplo.java
```

Si queremos compilar todos los .java de una carpeta podemos usar el asterisco:

```
javac *.java
```

El comando java que sirve para ejecutar un programa posee tambien las banderas -d y -cp.

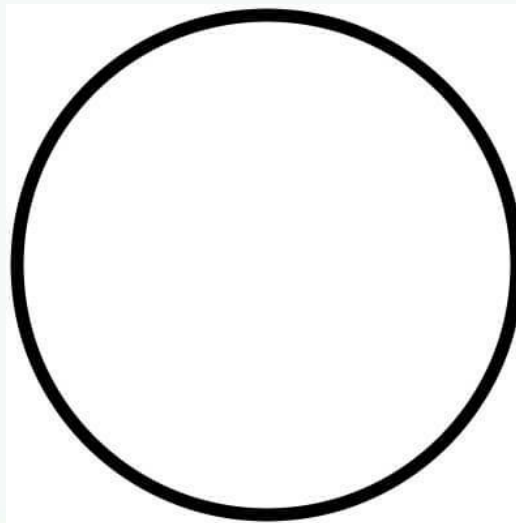
La clase Math es una clase de utilidad (que posee métodos y atributos relacionados con matemáticas. Esta clase se encuentra en `java.lang`, así que no hay que importarla.



Constantes

Posee dos constantes:

- * **e**: La base del logaritmo natural
- * **pi**: La relación entre la longitud de una circunferencia y su diámetro.



La clase Math incluye métodos relacionados con matemáticas, los cuales abarcan:

- Funciones trigonométricas
- Funciones exponencial y logarítmica
- Potencia y Raíz Cuadrada
- Redondeos de decimales
- Números Aleatorios
- Maximos y Minimos
- Valor Absoluto

Clase String

La clase String pertenece a `java.lang` por lo que no hay que importarla, la clase String permite representar texto en un programa en Java, aunque no es la única clase que permite esto es la más intuitiva y más usada.

Cabe destacar que los objetos pertenecientes a String son inmutables, es decir, una vez establecidos su valor estos no cambian.

Operadores Relacionales

Operadores Relacionales	
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que
!=	Distinto
==	Igual

Clase String

Cualquier función que se le aplique a un objeto de tipo String crea un nuevo objeto como resultado de la acción.

La clase String es un poco compleja en el uso de constructores, dado el siguiente código:

```
String cadena = "Texto";  
String cadena2 = "Texto";  
String cadena3 = new String("Texto");
```

Clase String

Si recordamos lo que el operador `==` hace, nos regresa true si `a == b`, a y b se encuentran en la misma dirección de memoria. El resultado de ese operador en las variables es:

- * `cadena == cadena2` -> True
- * `cadena2 == cadena3` -> False
- * `cadena == cadena3` -> False

Clase String

Normalmente la máquina virtual de java hace que los objetos que tengan la misma cadena en su interior apunten a la misma dirección de memoria, pero al momento de usar el constructor se fuerza a apuntar a una nueva dirección de memoria a pesar que internamente posean la misma información, es una mala práctica.

En general, para comparar objetos hay que utilizar el método `.equals()` en vez del operador `==` porque este genera fallas en algunas ocasiones.

Caracteres Especiales

Secuencia de escape	Valor
<code>\b</code>	Retroceso o <i>backspace</i> (equivalente a <code>\u0008</code>)
<code>\t</code>	Tabulador (equivalente a <code>\u0009</code>)
<code>\n</code>	Nueva línea (equivalente a <code>\u000A</code>)
<code>\f</code>	Salto de página (equivalente a <code>\u000C</code>)
<code>\r</code>	Retorno de carro (equivalente a <code>\u000D</code>)
<code>\"</code>	Doble comilla (equivalente a <code>\u0022</code>)
<code>\'</code>	Comilla simple (equivalente a <code>\u0027</code>)