Ciencias de la Computación
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

Curso de **Programación** en **Java**

Juan Francisco Maldonado León
Arquitecto de Software

Ciencias de la Computación
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

Fundamentos de Programación

Juan Francisco Maldonado León
Arquitecto de Software

Introducción a Java

Fundamentos de **Programación**

Java

Java es un lenguaje de programación originalmente desarrollado por **Sun Microsystems** (James Gosling) publicado en 1995

Ciencias de la Computación
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

Introducción a Java

Fundamentos de **Programación**

Java

Write Once, Run Anywhere

lenguaje independiente de la plataforma y un entorno de ejecución

Introducción a Java

Fundamentos de **Programación**

Java

Java paso a ser parte de **Oracle** hacia 2010 cuando esta adquirió las acciones de Sun mycrosystems por mas de 5.710 millones de dólares

Introducción a Java

Fundamentos de **Programación**

Java

- Lenguaje Orientado a Objetos.
- Compilado e Interpretado.
- Portable.
- Multihilo.

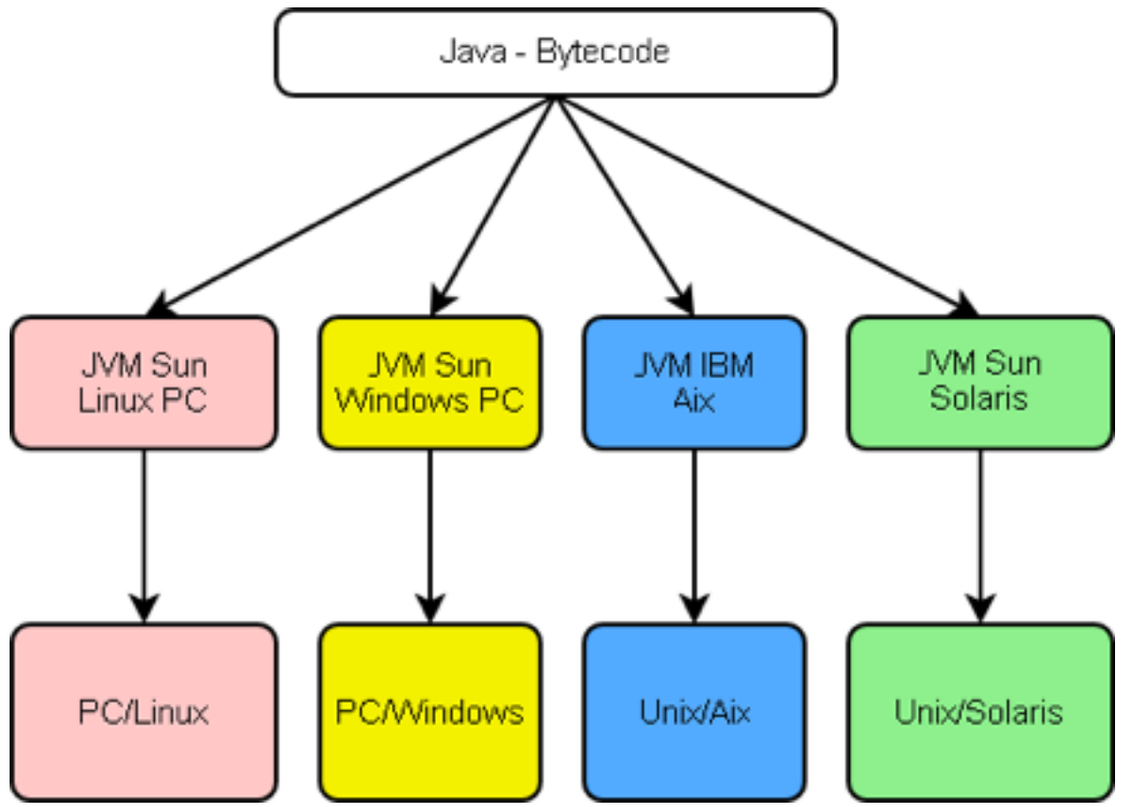
Maquina Virtual

Para poder ejecutar una aplicación en una Máquina Virtual de Java, el programa código (Archivo extension java) debe compilarse de acuerdo a un formato binario portable estandarizado, normalmente en forma de ficheros con extensión .class

Introducción a Java

Fundamentos de **Programación**

Maquina Virtual



Maquina Virtual

Para compilar un archivo

javac Archivo.java

Si la sintaxis esta correcta se generara un archivo con la extension .class

Introducción a Java

Fundamentos de **Programación**

Tipo de dato	Representación	Tamaño (Bytes)	Rango de Valores	Valor por defecto	Clase Asociada
byte	Numérico Entero con signo	1	-128 a 127	0	Byte
short	Numérico Entero con signo	2	-32768 a 32767	0	Short
int	Numérico Entero con signo	4	-2147483648 a 2147483647	0	Integer
long	Numérico Entero con signo	8	-9223372036854775808 a 9223372036854775807	0	Long
float	Numérico en Coma flotante de precisión simple Norma IEEE 754	4	$\pm 3.4x10^{-38}$ a $\pm 3.4x10^{38}$	0.0	Float
double	Numérico en Coma flotante de precisión doble Norma IEEE 754	8	$\pm 1.8x10^{-308}$ a $\pm 1.8x10^{308}$	0.0	Double
char	Carácter Unicode	2	\u0000 a \uFFFF	\u0000	Character
boolean	Dato lógico	-	true ó false	false	Boolean
void	-	-	-	-	Void

Introducción a Java

Fundamentos de **Programación**

```
String nombre = "juan";
char carcter = 'j';

int numero = 2102923232;
byte numByte = 127;
short numShort = 32123;
long numLong = 12321312312123123312L;

double numDouble = 12.3;
float numFloat = 123.3f;

boolean variableLogica = true;
```

Ciencias de la Computación
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

Introducción a Java

Fundamentos de **Programación**

Estructuras de Decisión Simple

```
if( numero > 10 )
{
    System.out.print("Numero mayor a 10");
}
```

Si condición **Entonces**
Instrucción 1
Instrucción 2
Instrucción 3
...
Fin Si

Introducción a Java

Fundamentos de **Programación**

Estructuras de Decisión Doble

```
if( numero > 10 )
{
    System.out.print("Numero mayor");
}
else
{
    System.out.println("Numero menor");
}
```

Si condición **Entonces**

Instrucción 1

...

Si No

Instrucción 1

...

Fin Si

Introducción a Java

Fundamentos de **Programación**

Estructuras de Decisión Multiple

```
switch( opcion)
{
    case 1 :
    {
        System.out.println("caso 1");
        break;
    }
    case 2 :
    {
        System.out.println("caso 3");
        break;
    }
    default :
    {
        System.out.println("De otro modo");
    }
}
```

Según expresión **hacer**
caso <opción 1>
instrucción 1
…
caso <opción 2>
instrucción 2
…
Fin caso

Introducción a Java

Fundamentos de **Programación**

Estructuras de Repetición Para

Para <inicializar> *Hasta* <condición> *Con Paso* <incremento> **Hacer**
    Instrucción 1
    Instrucción 2
    Instrucción 3
    . . .
FinPara

```
for( int cont=0; cont <10; cont++ )
{
    System.out.println( cont );
}
```

Introducción a Java

Fundamentos de **Programación**

Estructuras de Repetición Mientras

```
int cont = 0;
while ( cont > 10 )
{
    System.out.println(cont);
    cont++;
}
```

Mientras <condición>
Instrucción 1
Instrucción 2
...
Fin Mientras

Introducción a Java

Fundamentos de **Programación**

Estructuras de Repetición Repetir Hasta

```
do
{
     System.out.println(cont);
     cont++;
}
while( cont < 10 );
```

Repetir

Instrucción 1

Instrucción 2

...

Hasta Que <condición>

Introducción a Java

Fundamentos de **Programación**

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

Introducción a Java

Fundamentos de **Programación**

Packages

Los packages tiene la finalidad de estructurar nuestro código de forma jerárquica.

Es la primera sentencia dentro de una clase Java.

Si no se indica , quiere decir que se está utilizando el package por defecto

“principle of encapsulation and modularity".

Introducción a Java

Fundamentos de **Programación**

Packages

- Son opcionales
- Los nombres son equivalentes a una estructura de directorio ej cl.curso.java es igual a cl/curso/java
- La convención de nombres de package está dado por la organización de cada desarrollo.
- Los nombres de package que comienzan con **java**.* y **javax**.* son reservados
- Los nombres de packages deben ser en **minúsculas** y la separación entre palabras debe ser utilizando un guión bajo(_).

Introducción a Java

Fundamentos de **Programación**

Packages

La versión estándar de Java tiene una serie de packages los cuales se puede revisar en la siguiente dirección

http://docs.oracle.com/javase/7/docs/api/