Escuela Java

Introduccion a Java

Objetivo

- Lograr que aprendamos a programar en Java, hasta el punto de construir aplicaciones reales en la tecnología.

Agenda de hoy

- ✓ Parte 1: Herramientas de base y J2EE
  - Módulo 1:
    - ✓El lenguaje Java
      - ✓Introducción a la tecnología Java
      - ✓Sintaxis del lenguaje

Introducción I

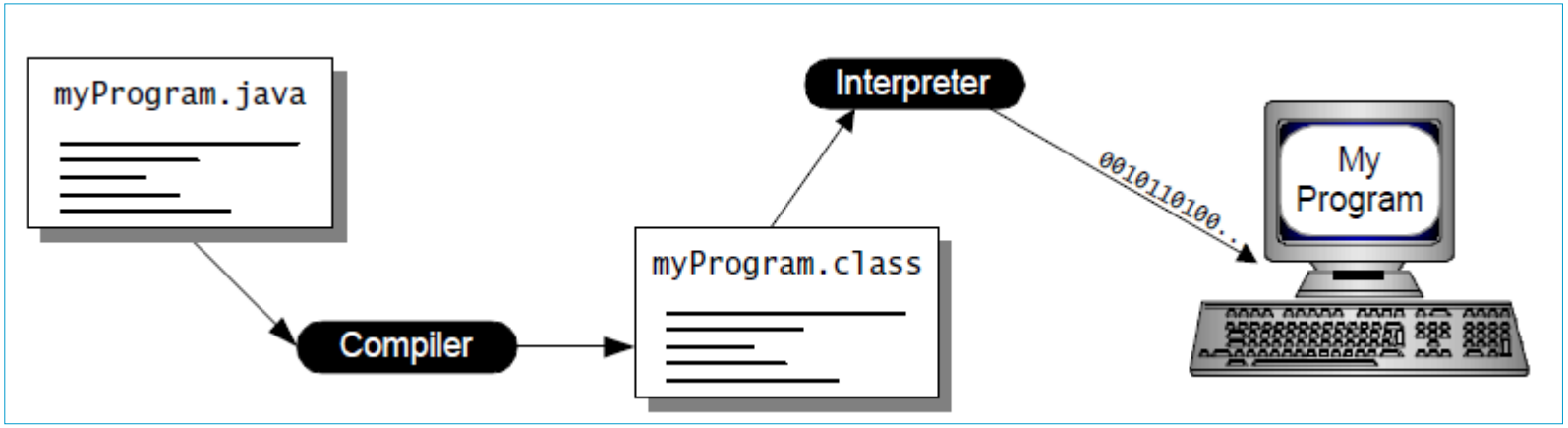
- Historia de Java

Introducción II

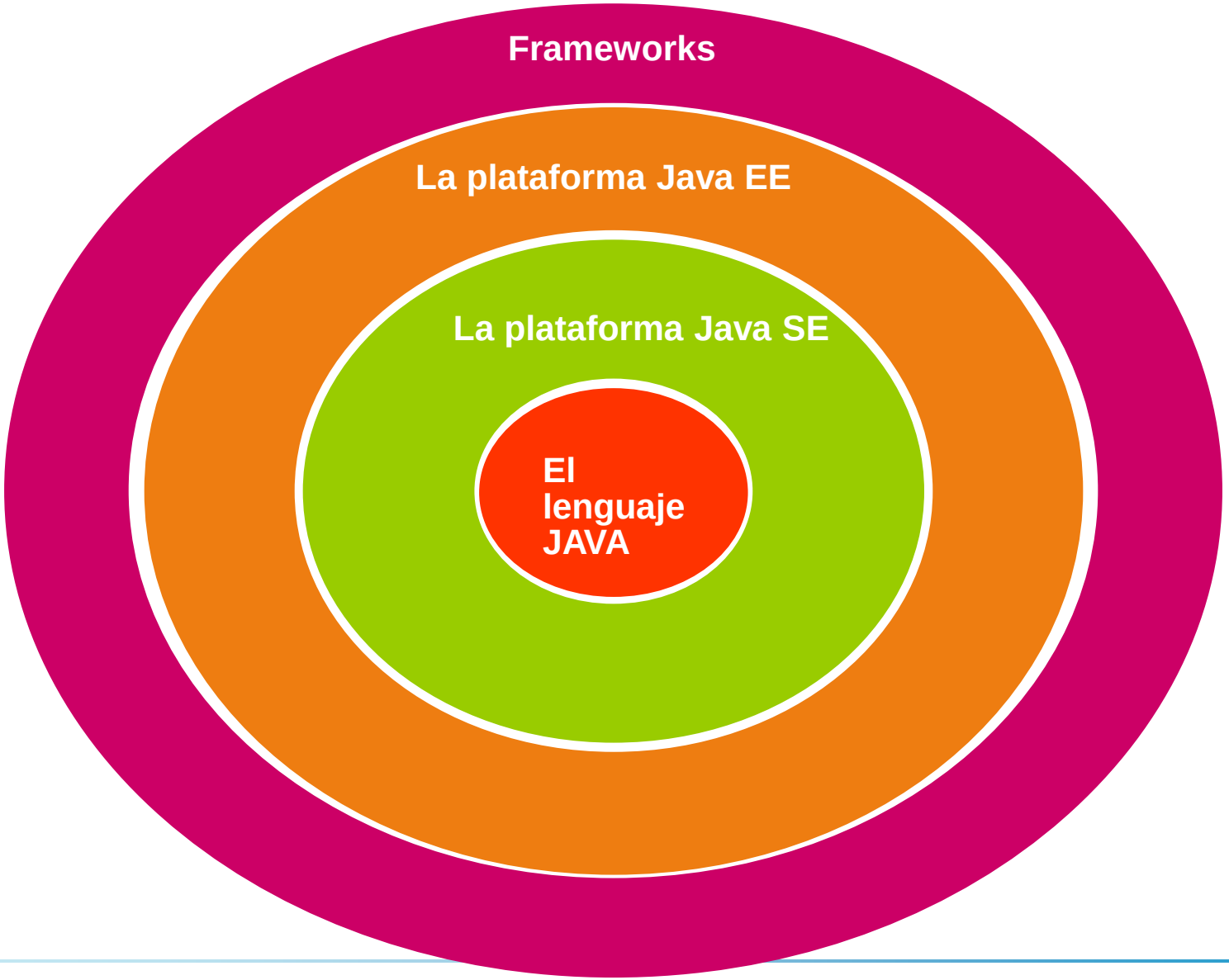
- Es una simplificación del C++
- Está concebido para desarrollo de aplicaciones en red.
- Permite multitarea.
- Es un lenguaje interpretado, independiente de la plataforma.
- Para poder ejecutar programas java, necesitamos una máquina virtual Java.
  - JVM.

Introducción III

- Java Virtual Machine (JVM)



Introducción IV



Introducción V

- El sitio oficial de SunMicrosystem
  - https://www.oracle.com/technetwork/es/java/javase/downloads/
- Al ejecutar el instalador se crea una estructura de carpetas en C:\Archivos de Programa\Java.

  - /bin . Herramientas y utilidades del JDK
  - /lib . Librerías del JDK
  - /demo . Archivos con códigos de ejemplo.
  - /jre . La JVM.
  - /db . Un DBMS gratuito
  - /src.zip . El código fuente de las librerías comprimido

Introducción VI

- Algunas de las herramientas incluidas son:
  - javac.exe -> Compilador de Java
  - java.exe -> Intérprete de Java (JVM)
  - jdb.exe -> Depurador
  - javadoc.exe -> Generador de documentación
  - javap.exe -> Desensamblador.

- El uso de este entorno de desarrollo es mediante editores de texto y la línea de comandos MS-DOS

Introducción VII

- Pero compilar por línea de comandos es la muerte en vida!

- Por ello: Entornos visuales!

  - **Eclipse**
  - **Netbeans**
  - **WebSphere Studio**
  - **Borland JBuilder**
  - **Oracle JDeveloper**

Sintaxis I

- Nombres de clases, métodos, propiedades y variables:
  - Deben empezar por una letra, _ o $.
  - Después del primer carácter se pueden usar números.
  - Java distingue entre mayúsculas y minúsculas.
  - No se pueden utilizar palabras reservadas:

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

Sintaxis II

- Existen tres formas de escribir los comentarios:

```
// Comentario en una única línea

/* Comentario de una o
más líneas */

/**
Comentario en formato JavaDoc
*/
```

Sintaxis III

- Una sentencia es una línea simple de código terminada en punto y coma:

```
System.out.println(“Hola”);
```

- Un bloque es un conjunto de sentencias agrupadas entre llaves. Pueden estar anidados.

```
while (true) {

    x=y+1;

    if(x<0) {
        x=x+1;
    }
}
```

Sintaxis IV

Variables y tipos de datos

- Una declaración de variable consiste en una sentencia en la que indicamos el nombre de la variable y el tipo de dato que va a almacenar:

```
int contador;
```

- En la misma sentencia podemos asignar un valor a la variable:

```
int contador = 10;
```

Sintaxis V

Variables y tipos de datos

- En Java existen dos tipos de datos genéricos:
  - Tipos primitivos
  - Tipos complejos -> Clases.
- Tipos Primitivos
  - Lógico: boolean.
  - Caracter: char.

```
boolean acierto = true;
```

```
char unaLetra = ‘a’;
```

Sintaxis VI

Variables y tipos de datos

- Tipos Primitivos
  - Números enteros: byte, short, int y long

```
byte unByte = 12;
short unShort;
int unEntero = -199;
```

  - Números reales: double y float.

```
float unFloat = 0.17F;
double unDouble;
double otroDouble = -12.01E30;
```

Sintaxis VII

Variables y tipos de datos

- Tipos Complejos (Clases)

- Se identifican con el nombre de la clase a la que pertenecen.
- Su valor por defecto es null.

```
Producto unProducto;
String unString = new String("HOLA");
```

Sintaxis VIII

Variables y tipos de datos

- Ejemplo

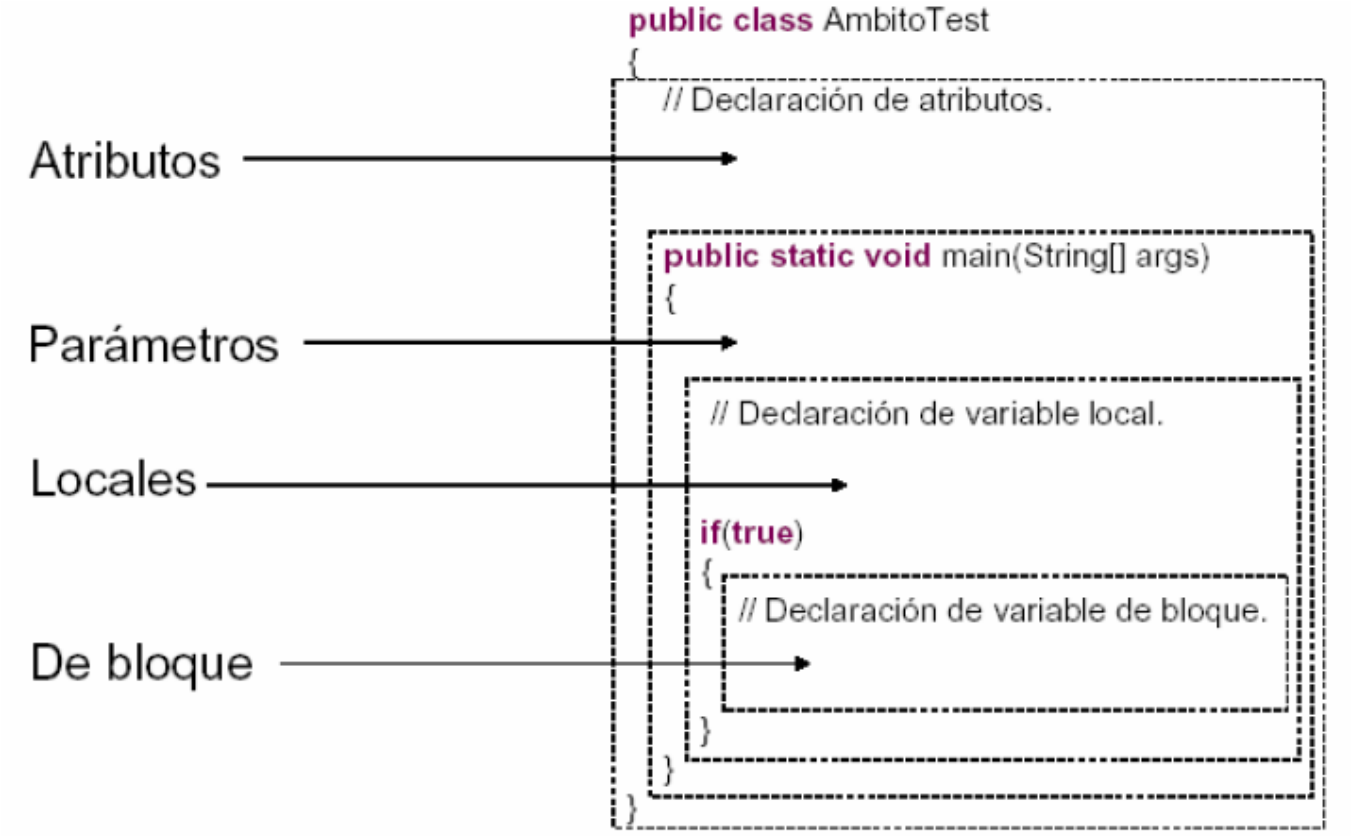
```
public class VariablesTest2
{
   public static void main(String[] args)
   {
      boolean unBoolean = true;
      byte    unByte    = 10;
      short   unShort   = 10;
      int     unInt     = 10;
      long    unLong    = 10;
      float   unFloat   = 3.14F;
      double  unDouble  = 3.14;
      char    unChar    = 'A';
      String  unString  = new String("Hola");

      System.out.println("El boolean vale: " + unBoolean);
      System.out.println("El byte vale: " + unByte);
      System.out.println("El short vale: " + unShort);
      System.out.println("El int vale: " + unInt);
      System.out.println("El long vale: " + unLong);
      System.out.println("El float vale: " + unFloat);
      System.out.println("El double vale: " + unDouble);
      System.out.println("El char vale: " + unChar);
      System.out.println("El String vale: " + unString);
   }
}
```

Sintaxis IX

Variables y tipos de datos

- Variable Scope



Sintaxis X

Operadores

- Operadores aritméticos
  - Los operadores básicos de Java son **+ , - , * , /** para suma, resta, producto y división. El operador **%** calcula el resto de una división.
  - Además existen los operadores decremento e incremento: **--** y **++** respectivamente.
  - También es posible utilizar los operadores **+=, -=**,…etc.

Sintaxis XI

Operadores

- Operadores de comparación

!	Negación
==	Es igual
!=	Distinto
<	Menor
>	Mayor
<=	Menor o igual
>=	Mayor o igual
&&	Y
||	O

Sintaxis XII

Operadores

- Otros Operadores

?	Es una abreviatura de if-then-else. op1 ? op2 : op3 Si se cumple el op1, se evalúa op2 y si nó op3.
[ ]	Para declarar, crear y acceder a arrays
.	Para acceder a atributos y métodos de los objetos
(parámetros)	Para pasar parámetros a los métodos
(tipo)	Para hacer castings
new	Para crear objetos nuevos (instanciar)
instaceof	Comprueba si el primer operando es una instancia del segundo

Sintaxis XIII

Flow of Control

- Sentencia while

```
while (condicion){
<instrucciones>
}
```

- Sentencia do while

```
do {
<instrucciones>
} while(condicion);
```

```
public class Bucles
{
   public static void main(String[] args)
   {
      int cont1 = 0;
      while(cont1 < 3)
      {
         System.out.println(cont1);
         cont1++;
      }

      int cont2 = 0;
      do
      {
         System.out.println(cont2);
         cont2++;
      }
      while(cont2 < 3);
   }
}
```

Sintaxis XIV

Flow of Control

- Sentencia for

```
for (inicialización; condicion; incremento){
<instrucciones>
}
```

```
for(int cont3 = 0; cont3 < 3; cont3++)
{
   System.out.println(cont3);
}
```

Sintaxis XV

Flow of Control

- Sentencia if

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

```
if(expresión)
{
   sentencias;
}
else
{
   sentencias;
}
```

```
if(expresión)
{
   sentencias;
}
else if(expresión)
{
   sentencias;
}
else
{
   sentencias;
}
```

Sintaxis XVI

Flow of Control

- Sentencia switch

```
switch(selector) {
case valor1 : Grupo de sentencias1; break;
case valor2 : Grupo de sentencias2; break;
case valor3 : Grupo de sentencias3; break;
case valor4 : Grupo de sentencias4; break;
case valor5 : Grupo de sentencias5; break;
// ...
default: statement;
}
```

Sintaxis XVII

Flow of Control

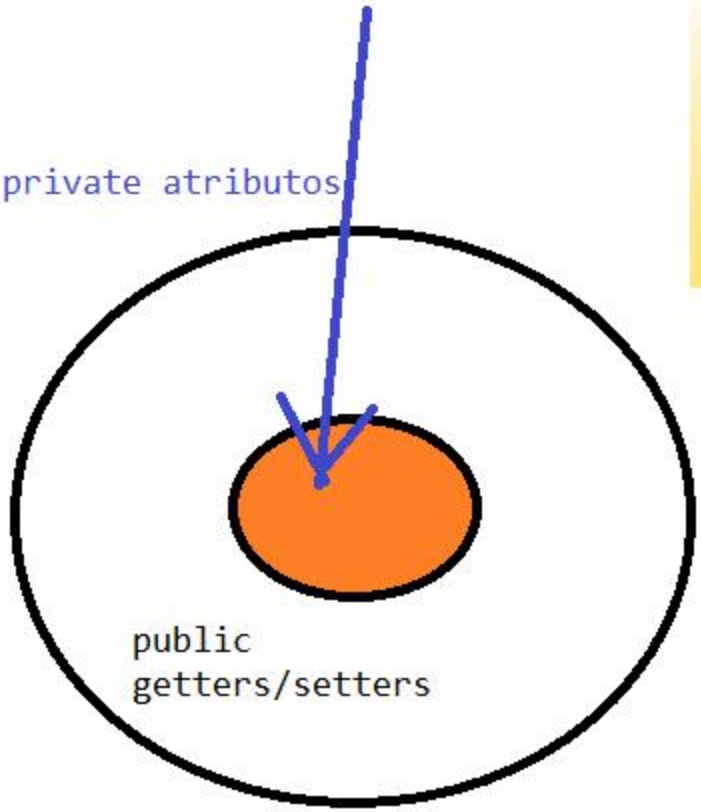
- Sentencias break, continue y return

```
while (true) {
     i++;
     int j = i * 27;
     if (j == 1269)
          break; // Salimos del lazo
     if (i % 10 != 0)
          continue;// Salto a la siguiente iteración
     System.out.println(i);
}
```

Ejercicio

Ejercicio II de la guía

Muchas gracias!



POJO = PLAIN OLD JAVA OBJECT