
1

¿Qué es Java?



Características

- Lenguaje de programación basado en C y C++
- Filosofía *Write Once, Run Anywhere*
- Se ejecuta en una máquina virtual (JVM)
- Orientado a Objetos
- Tipado estático

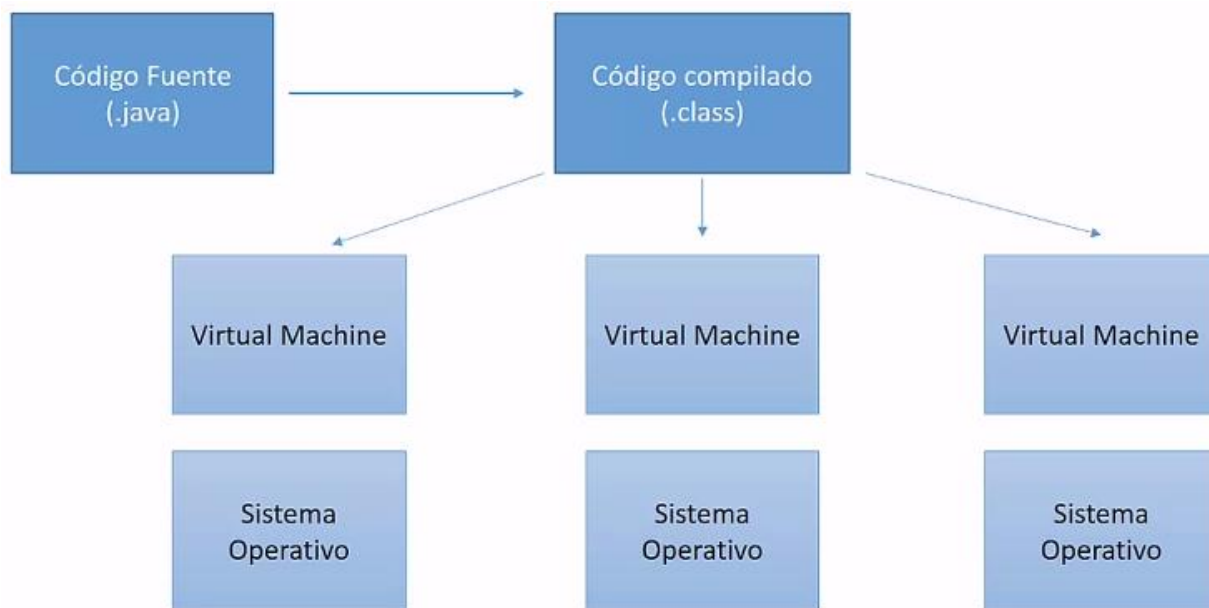


Breve historia

- Creado por James Gosling en Sun Microsystems
- Lanzado originalmente en 1995
- Diversos releases: JDK 1 (1996) – Java SE 8 (2014)
- Adquirido por Oracle en 2010



Java Virtual Machine





Java Virtual Machine

- Añade una capa de abstracción entre el código y el sistema operativo
- Ejecuta Java Bytecode
 - Código ensamblador para JVM



Visión General de la plataforma Java

- El lenguaje Java
- El compilador Java
- La JVM
- El Recolector de Basura
- El kit de desarrollo de Java (JDK)
- El Java Runtime Environment (JRE)



Tipo de Aplicaciones

- Aplicaciones de escritorio (Swing, Java Fx)
- Aplicaciones Web (Java EE)
- Aplicaciones Móviles (Android)



Práctica

Hola Mundo !

2

Instrucciones, bloques y comentarios



Instrucciones

- Son las acciones que podemos pedir que haga el programa
- En Java se separan utilizando un “;”

```
int a = 0;  
int b = 5;  
int c = a+b;
```



Bloques

- Agrupan instrucciones
- En Java se utilizan las llaves “{” y “}” para delimitarlos



Comentarios

- De una línea “//”
- De múltiples líneas “/* */”

3

Variables



Variables

- Identificadores válidos
- Declaración y definición
- Tipos de datos Java



Tipos de Datos

Enteros	byte, short, int, long
Reales	float, double
Lógicos	boolean
Caracteres	char
Texto	String



Asignación

- Operador “=”
- Sentido de derecha a izquierda

4

Entrada y Salida de datos



Salida e ingreso de datos

- Salida en Consola

```
System.out.println ("Hola Mundo!")
```

- Lectura desde teclado

```
Scanner entrada = new Scanner(System.in);  
System.out.println("Ingresa tu nombre");  
String nombre = entrada.nextLine();  
System.out.printf("Tu nombre es %s \n",nombre);
```

5

Conversión de tipos



Conversiones

- Implícitas
- Casting
- Métodos en las clases asociadas



Conversiones

```
int a = 5;
```

```
int b = 7;
```

```
double c = 0.5;
```

```
c = a; // Implícita
```

```
a = (int)c; // Explícita o "casting"
```

```
String s = "7";
```

```
b = Integer.parseInt(s); // Métodos en las clases asociadas
```

6

Operadores aritméticos



Operadores aritméticos

Operador	Operación
+	Suma
-	Resta
*	Multiplicación
/	División
%	Resto

7

Estructuras de alternativas



Alternativa simple

```
if (expresión)
{
    sentencia1;
    sentencia2;
}
```



Alternativa simple

```
if (expresión){  
    sentencia1;  
    sentencia2;  
}  
else {  
    sentencia3;  
}
```



Alternativa múltiple

```
switch (expresión) {  
    case constante1:  
        sentencia1;  
    break;  
    case constante2:  
        sentencia2;  
    break;  
    default:  
        sentencia3;  
    break;  
}
```

8

Operadores relacionales



Operadores relacionales

Operador	Operación
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que
==	Igual que
!=	Distinto de



Operadores relacionales

Operador	Operación
&&	AND
	OR
!	NOT



Práctica

Crea un proyecto llamado EjercicioHoraDia que pida al usuario la hora actual, en horas y minutos. Después en función de los valores que el usuario escriba, mostrará si es de madrugada, mañana, tarde o noche.

En caso de que sean las doce (mediodía o noche) mostrará un mensaje especial al usuario.

9

Estructuras repetitivas



Repetitivas

$0 \rightarrow n$

$1 \rightarrow n$

$n \rightarrow m$

while

do-while

for



while

```
while (condicion_fin)
{
    sentencia1;
    sentencia2;
    ...
}
```



do while

```
do
{
    sentencia1;
    sentencia2;
    ...
}
while (condición_fin)
```



for

```
for (valor_inicial; condicion_fin;  
variacion)  
{  
    sentencia1;  
    sentencia2;  
    ...  
}
```

10

Funciones



Función main()

```
public static void main (String[] args)
{
    ...
}
```



Componentes de una función

- Nombre
- Parámetros
- Valor de retorno
- Instrucciones



Declaración

```
int multiplicaPorTres (int a)
{
    int r= a * 3;
    return r;
}
```




Llamada

```
int multiplicaPorTres (int a){  
    int r= a * 3;  
    return r;  
}  
  
public static void main(String[] args) {  
    int a = 5;  
    int b= multiplicaPorTres(a);  
}
```

11

Objetos y clases



Objeto y clase



Bombilla
-estado : boolean
+encender() +apagar()

Un objeto tiene estado, comportamiento e identidad.



Clase Bombilla

```
public class Bombilla{  
    private boolean estado = false;  
    public void encender () {  
        estado = true;  
    }  
    public void apagar () {  
        estado = false;  
    }  
}
```



Clase Bombilla

```
public class Bombilla{  
    private boolean estado = false;  
    public void encender () {  
        estado = true;  
    }  
    public void apagar () {  
        estado = false;  
    }  
}
```



Instancia de una clase

```
public class MiAplicacion {  
    public static void main (String[] args) {  
        Bombilla b = new Bombilla ();  
    }  
}
```



Mensaje

```
public class MiAplicacion {  
    public static void main (String[] args) {  
        Bombilla b = new Bombilla ();  
        b.encender ();  
    }  
}
```



Constructor

```
public class Bombilla {  
    private boolean estado = false;  
    Bombilla (boolean estado) {  
        this.estado = estado;  
    }  
}
```

12

Encapsulación



Clase Bombilla

```
public class Bombilla{  
    private boolean estado = false;  
    public void encender () {  
        estado = true;  
    }  
    public void apagar () {  
        estado = false;  
    }  
}
```



Clase Bombilla

```
public class MiAplicacion {  
    public static void main (String[] args) {  
        Bombilla b = new Bombilla ();  
        b.encender ();  
        b.estado = false;  
    }  
}
```



Visibilidad

Tipo	Modificador
Privada	private
Pública	public
Protegida	protected
Por defecto o de paquete	