

Lenguaje JAVA

Recordando JAVA desde cero

Si alguien se quiere retirar, adelante, pero...
¿de verdad crees que ya lo sabes todo?

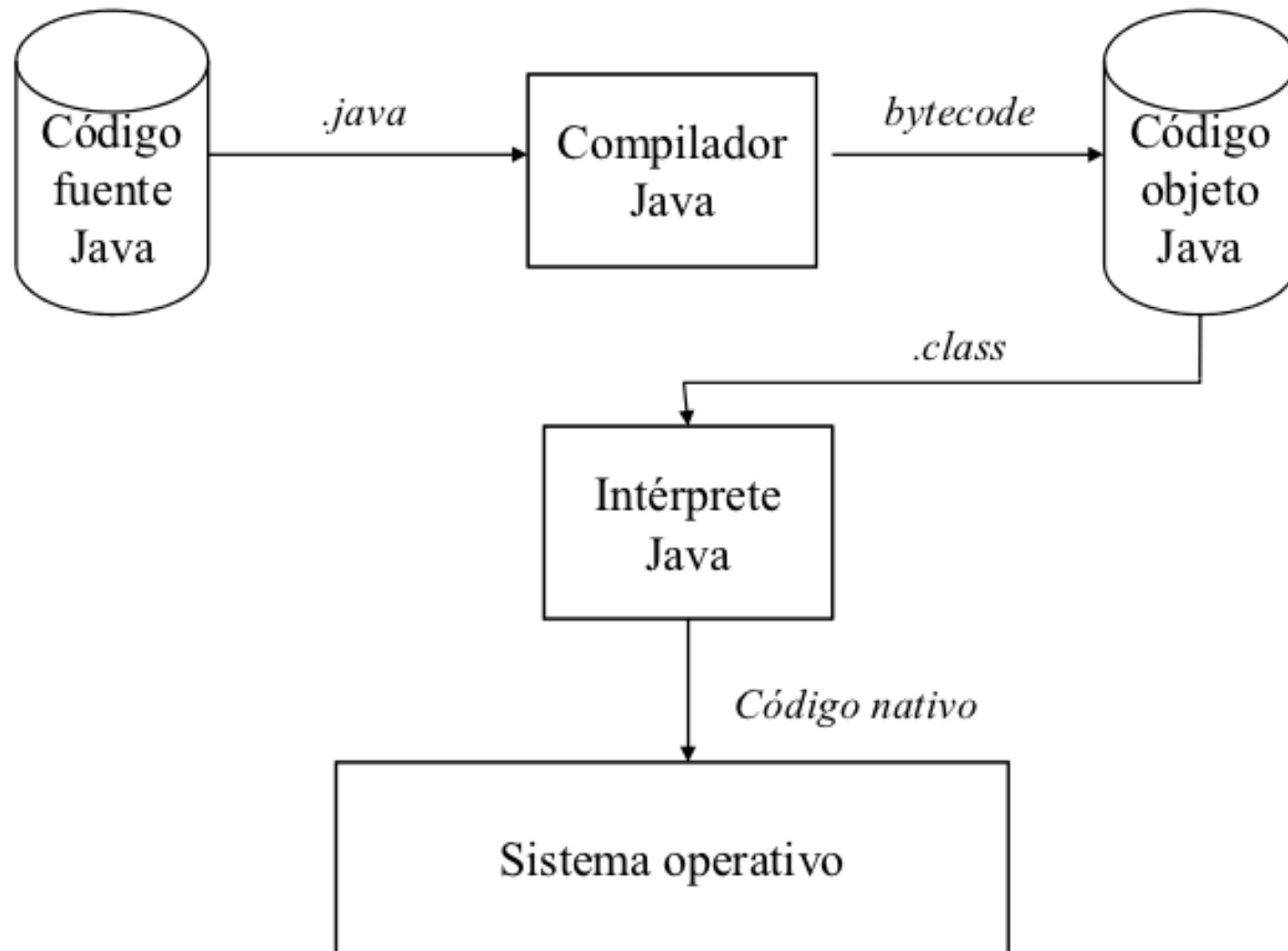


Ambiente de desarrollo

- A partir de un código fuente Java (*archivo .java*), se genera a través de un compilador Java (*javac*), código objeto Java (*bytecodes*) en un archivo *.class*.
- Una vez obtenido este código compilador a un nivel independiente de la plataforma, el intérprete de Java (máquina virtual de Java, **JVM**) lee los *bytecodes* generando código nativo específico de una plataforma.



... Ambiente de desarrollo



Estructura de un programa

- Un programa en Java se estructura con base en las siguientes reglas:
 - Al menos define una sola clase, pero pueden haber varias
 - Para mejor organización, cada clase debería estar en su propio archivo.
 - El nombre del archivo debe ser el mismo que el nombre de la clase definida.
 - Tal archivo debe tener la extensión .java



... Estructura de un programa

- Típicamente, un programa Java se estructura en tres partes:
 - Declaración del paquete,
 - Importación de clases (o paquetes)
 - Declaración de la clase



... Estructura de un programa

```
//Declaración de paquete  
package udg.cucei.tps.sesion01;  
  
//Declaración de importaciones  
//Importa una clase específica  
import javax.swing.JButton;  
  
//Importa un paquete entero  
import java.util.*;  
  
//Definición de clase  
public class PrimeraClase {  
    //Implementación  
}
```



Variables

- Una variable es una porción de memoria referenciado con un identificador que sirven para almacenar un dato. Una variable solo puede almacenar un solo tipo de dato.
- Por lo tanto, cada dato tiene su representación en memoria y su posible rango de valores. Una variable se declara de la forma:

```
tipo identificador;
```



... Variables

- Un identificador es usado por el programador para referenciar una variable, un paquete, un método o una clase.
- Las reglas para los identificadores son:
 - Cadenas de texto Unicode de cualquier tamaño
 - El carácter inicial debe ser una letra, un guión bajo (_) o signo de peso (\$)
 - No puede ser igual a una palabra reservada, literales booleanas (true, false) o null
 - Después, puede contener cualquier combinación de caracteres excepto los espacios y caracteres reservados
 - Distingue mayúsculas y minúsculas



... Variables

- **Por convención**, los identificadores de variables comienzan con letra minúscula. Si fuera conformado por varias palabras, las demás palabras después de la inicial, se escriben con letra mayúscula.



Tipos de datos primitivos

- Java ofrece sentencias de tipos para representar datos primitivos como
 - Valores numéricos enteros y reales
 - Caracteres
 - Valores lógicos



... Tipos de datos primitivos

- Los tipos de datos numéricos enteros son:

Nombre	Tamaño	Valores
byte	8 bits	-128 a 127
short	16 bits	-32768 a 32767
int	32 bits	-2147483648 a 2147483647
long	64 bits	-9223372036854775808 a 9223372036854775807



... Tipos de datos primitivos

- Los tipos de datos numéricos decimales son:

Nombre	Tamaño	Valores
float	32 bits	-3.4E8 a 3.4E38
double	64 bits	-1.7E308 a 1.7E308



... Tipos de datos primitivos

- Los tipos de datos lógicos son:

Nombre	Tamaño	Valores
boolean	1 bit	true, false



... Tipos de datos primitivos

- Los tipos de datos de caracteres son:

Nombre	Tamaño	Valores
char	16 bits	Tabla Unicode



Impresión en consola

- La forma más elemental de presentación de información en consola, es utilizando la función

```
System.out.println(...);
```

- Donde ... Es una cadena de texto o una concatenación de cadenas.

- Ejemplo:

```
"esta es una cadena" + "concatenada"
```

```
"el valor de i es " + i
```



... Impresión en consola

- Otra forma de impresión en consola es con el clásico método **printf** de la siguiente forma:

```
int i=2;
```

```
System.out.printf("El valor de %s  
es %d\n", "i", i);
```



Constantes

- Una constante es una variable cuyo contenido no puede ser modificado una vez inicializado su valor.
- En Java se les llama variables finales. Una constante se declara usando el modificador final.

```
final tipo identificador = valor;
```

- Ejemplos:
 - `final double PI = 3.1416;`
 - `final int MAX = 6;`



Operaciones aritméticas

- Java permite el cálculo de operaciones aritméticas básicas tales como:
 - Suma (+)
 - Resta (—)
 - Multiplicación (*)
 - División (/)
 - Módulo (%)



Operaciones aritméticas

- Ejemplos:

```
int a = 2;
```

```
int b = (a*50) / (4-a);
```

```
long c = 8390L;
```

```
a = c/b;
```

```
//¿Cuánto vale a?
```

```
double d = 5.5;
```

```
float f = d*2;
```

```
//¿Cuánto vale f?
```



Incremento / Decremento

- Al igual que C++, Java ofrece operadores para incrementar una variable en una cantidad dada.

- Suponga la línea siguiente:

```
int i = 0;
```

- Incremento en 1

```
i++; o ++i; //equivalente a i=i+1;
```

- Decremento en 1

```
i--; o --i; //equivalente a i=i-1;
```



... Incremento / Decremento

- Incremento en n

`i += n; //equivalente a i=i+n;`

- Decremento en n

`i -= n; //equivalente a i=i-n;`

- Ejemplos:

```
int a = 20;
```

```
int b = 30;
```

```
int c = a++ + --b; //¿Cuánto vale c?
```



Comentarios

- Son textos dentro del código fuente de un programa que son ignorados por el compilador y no generan instrucciones máquina.
- Suelen acompañar una línea de código para explicar el propósito de esa línea en el programa.
- Pueden ser de una o varias líneas.



... Comentarios

- Ejemplo

```
/*
```

```
* Este es un
```

```
* comentario multilinea
```

```
*/
```

```
//Este comentario es de una linea
```



Promoción de datos (casting)

- Hay situaciones donde se requiere ejecutar operaciones con diferentes tipos de datos (ej. Sumar un double con un int). Tales casos pueden generar ambigüedad. Considere la siguiente operación:

```
double res = 1.5 + 3/2;
```

- ¿Cuánto vale res?



... Promoción de datos (casting)

- La respuesta es 2.5. ¿Por qué?
- Debido a que 3 y 2 son enteros, entonces el resultado es interpretado como un entero, por lo tanto, el resultado parcial sería $1 + 1.5 = 2.5$.
- ¿Cómo hacer para que sea interpretado como un dato double?



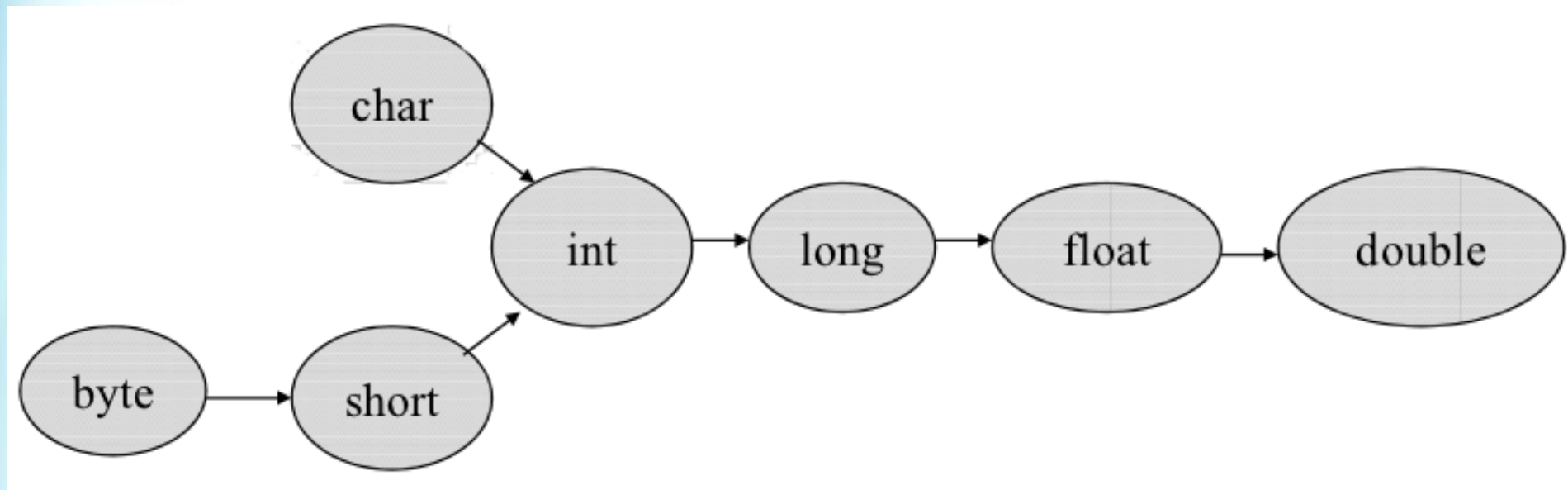
... Promoción de datos (casting)

- Existen dos tipos de casting:
- Implícito
 - Si uno de los operandos es double, entonces el otro operando es interpretado como double
- Explícito
 - Se antecede el tipo de dato a la cual se quiere ajustar el resultado de una operación

```
res = 1.5 + ((double) 3) / 2;
```



... Promoción de datos (casting)



Operaciones a nivel de bits

- A diferencia de otros lenguajes, Java ofrece operadores para ejecutar acciones a nivel de bits. Esto es, para modificar el contenido de una variable bit por bit.
- Tales operadores son:
 - `&`, AND lógico
 - `|`, OR lógico
 - `^`, XOR lógico
 - `~`, negación o complemento
 - `>>`, corrimiento derecho con propagación de signo
 - `<<`, corrimiento izquierda con 0s a la derecha
 - `>>>`, corrimiento derecho con 0s a la izquierda



... Operaciones a nivel de bits

- Ejemplos:

```
int a = 0x00000000;
```

```
a = ~a | 0x100;
```

```
a >>>= 7;
```

```
//¿Cuánto vale a?
```



Operaciones booleanas

- Son expresiones que generan un valor booleano (true o false). Para conjuntar una expresión booleana es necesario utilizar operadores booleanos. En Java son:
 - &&, AND lógica
 - ||, OR lógica
 - !, negación o diferencia
 - ==, igualdad
 - !=, desigualdad
 - >, mayor que
 - <, menor que
 - >=, mayor o igual
 - <=, menor o igual



... Operaciones booleanas

- Ejemplos:

```
boolean b1 = (5 == 4);
```

```
boolean b2 = !b1;
```

```
b1 = b1 || b2;
```

- ¿A qué es igual la expresión $!(b1 \ || \ b2)?$
 $!b1 \ \&\& \ !b2$



Condicionales

- Son las que permiten redirigir el flujo del programa con base en el resultado de una evaluación de una expresión booleana (condición).

```
if (expresión-booleana) {  
    bloque de código1;  
} else {  
    bloque de código2;  
}
```

- Si la expresión booleana se cumple entonces se ejecuta el bloque 1, en otro caso, el bloque 2.



... Condicionales

- Si se requiere efectuar una selección de entre varias opciones, y la opción es entera, entonces es conveniente usar la estructura switch.

```
switch (variable_numérica) {  
    case valor1:  
        sentencias;  
        break;  
    case valor2:  
        sentencias;  
        break;  
    default:  
        sentencias;  
        break;  
}
```



... Condicionales

- Ejemplo:

```
switch (mes) {  
    case 12: case 1: case 2: estacion = "Invierno";  
    break;  
    case 3: case 4: case 5: estacion = "Primavera";  
    break;
```



Iterativas

- Las sentencias de iteración son mejor conocidas como bucles. En éstos se ejecuta un bloque de sentencias mientras una expresión booleana sea verdadera.

```
while (expresión-booleana) {  
    sentencias;  
}
```



... Iterativas

- Una variante de esta forma es definir hasta el final la expresión booleana.

```
do {  
    sentencias;  
} while (expresión-booleana);
```

- El ciclo por lo menos se ejecuta UNA vez.



... Iterativas

- Otra manera de declarar bucles es mediante la declaración de un rango de veces en el cual se efectúa un bloque de código.

```
for (inicialización; terminación;  
incremento) {  
sentencias;  
}
```



... Iterativas

- Ejemplo

```
int a,b;  
for (a=1,b=4; a<b; a++,b--) {  
    System.out.println("a=" + a);  
    System.out.println("b=" + b);  
}
```



Sentencias de paro

- Un ciclo puede terminarse abruptamente o condicionarse su ejecución.
- El uso de la sentencia **break** nos permite romper con la ejecución de un bucle.

```
ciclo {  
    ...  
    break;  
    ...  
}
```



... Sentencias de paro

- Otra sentencia de control de ejecución dentro del bloque de un ciclo es la sentencia **continue**. Esta sentencia rompe con la ejecución del ciclo sin salirse del ciclo.

```
ciclo {  
    ...  
    continue;  
    ...  
}
```



Ciclo for mejorado

- A partir de la la versión 1.5, Java ofrece una versión alterna del ciclo for. Esta consiste en reemplazar los ciclos incrementales en uno, por una expresión más sencilla, para el recorrido de arreglos.

```
for ( parámetro: arreglo )  
    sentencia
```



Alcance de una variable

- Una variable tiene un período de vida que está determinado por el ámbito donde se declaró. Por lo tanto, el alcance o cobertura indica su visibilidad y usabilidad.
 - Si la variable se declara en el ámbito de una clase (variable global), su cobertura es de toda la clase
 - Si la variable se declara en el ámbito de una función (variable local), su cobertura es solo la función
 - Si la variable se declara dentro de una sentencia (if, for, while, etc.), su cobertura es solo la sentencia



...Alcance de una variable

- Ejemplo

```
int x;
```

```
...
```

```
if (x > 0) {
```

```
    int y = x + 2;
```

```
    ...
```

```
}
```

```
x = y + 3;
```

```
//¿Cuánto vale la variable x?
```



Arreglos

- Un arreglo de datos primitivos es una colección ordenada de datos primitivos. Los arreglos son homogéneos, todos los datos del arreglo deben ser del mismo tipo.
- Para crear un arreglo se siguen tres pasos:
 - Declaración
 - Construcción
 - Inicialización



... Arreglos

- La declaración especifica al compilador cuál es el nombre del arreglo y de qué tipo es.
- Ejemplos:

```
int[] enteros;
```

```
double[] dobles;
```

```
int[][] matriz;
```

```
char cadena[];
```

- Como se ve, la declaración no define el tamaño del arreglo.



... Arreglos

- La construcción del arreglo se hace en tiempo de ejecución y lo que hace es reservar espacio en memoria para ese conjunto de datos.
- Ejemplos:

```
//Arreglo de 50 enteros
```

```
int[] enteros = new int[50];
```

```
int[][] matriz = new int[2][2];
```



... Arreglos

- La inicialización es la asignación de valores a cada elemento del arreglo.
- Ejemplos:

```
int[] enteros = {5, 6, 7};
```

```
int[] a = new int[2];
```

```
a[0] = 8;
```

```
a[1] = -2;
```

```
int[][] matriz = {{1, 0}, {0, 1}};
```



... Arreglos

- Cuando no se inicializa el arreglo, los valores por defectos son:
 - int, short, byte en **0**
 - long en **0L**
 - float en **0.0f**
 - double en **0.0d**
 - char en **'\u0000'**
 - boolean en **false**



Tamaño de un arreglo

- Una propiedad muy útil de los arreglos es su tamaño.
- Tal propiedad se llama length
- Ejemplo:

```
int[] arreglo = new int[5];  
System.out.println("Tamaño: " +  
arreglo.length);
```



Parámetros de programa

- Se dice que una clase es ejecutable (o es un programa en Java) si tiene el método estático main.
- Este método define un parámetro que es un arreglo de cadenas (`String[] args`).
- Cada cadena corresponde a un parámetro enviado desde la línea de comando, de tal forma que `args[0]` es el primer parámetro, `args[1]` el segundo, y así sucesivamente en orden de aparición



... Parámetros de programa

- Ejemplo:

```
C:\> java MiClase param1 param2  
param3 .... paramN
```

- De tal forma que el arreglo args sería de tamaño N, donde args[0] contiene “param1”, args[1] contiene “param2”, y así sucesivamente



Tarea

- Investigar los métodos de lectura de datos en consola para Java y sus diferencias.

