

Introducción a Java

Conceptos básicos de la POO y elementos del lenguaje Java

Fundamentos de Programación

Departamento de Lenguajes y Sistemas Informáticos



2024/25

ÍNDICE

Introducción a Java

- Conceptos Básico de la P.O.O
- Elementos básicos del lenguaje Java
- Tipos de datos
- Variables y constantes
- Expresiones y Operadores



2024/25

Programación Orientada a Objetos

- ▶ La POO (**P**rogramación **O**rientada a **O**bjetos) es una forma de construir programas de ordenador donde la información y su manejo se basa en la forma que tenemos los humanos de concebir objetos.
- ▶ Unos objetos se distinguen de otros mediante sus propiedades y las funcionalidades o capacidades con que se manejan dichas propiedades. Los objetos que se modelen deberán considerar las propiedades que sean relevantes para el problema que se quiere resolver.



2024/25

Programación Orientada a Objetos

- ▶ Los elementos básicos en los que se organiza la POO son:
 - p**aquetes (permiten agrupar elementos del proyecto por algún criterio)
 - I**nterfaces (permiten definir el comportamiento de los objetos)
 - C**lases (permiten definir los objetos y manejarlos)
 - a**tributos (almacenan las propiedades)
 - m**étodos (definen las funcionalidades que consultan o actualizan las propiedades)
 - E**numerados (permiten manejar variables que toman un conjunto discreto de valores)



2024/25

Construcción de un proyecto Java

- ▶ Para **explicar y practicar** la Programación Orientada a Objetos utilizaremos **Java** como lenguaje de programación sobre un Entorno de Desarrollo Integrado (IDE) **Eclipse**.
- ▶ Implementaremos proyectos Java con la siguiente estructura:
(observe la **mayúscula/minúscula** de la primera letra de los elementos que componen un programa)
 - ▶ **N**ombreDelProyecto (aquí usamos la notación “*CamelCase*”)
 - ▶ **src** (*carpeta que por defecto crea eclipse para los programas fuentes*)
 - ▶ **p**aquetes
 - ▶ **E**numerados
 - ▶ **I**nterfaces
 - ▶ **C**lases
 - ▶ **a**tributos (almacenan las propiedades)
 - ▶ **m**étodos (consultan o actualizan las propiedades)
 - ▶ **v**ariables (permiten almacenar valores para las operaciones)
 - ▶ **C**ONSTANTES (todas en mayúsculas y con notación “*SNAKE_CASE*”)
- ▶ Elementos básicos del lenguaje Java



2024/25

¿Cómo se definen tipos objeto?

- ▶ Veremos que **Java dispone de numerosos tipos de objetos** ya implementados mediante sus respectivas clases. *Por ejemplo, para fechas, horas, duraciones, cadenas de caracteres,...*
- ▶ No obstante, **el programador puede definir sus propios tipos de objetos**, mediante la implementación de sus propias **clases**.
- ▶ Una **clase** es una descripción de:
 - ▶ Los *atributos* de los objetos (*qué información se guardará de un objeto.*).
 - ▶ Las funcionalidades de los objetos (**métodos** *que permite consultar/ modificar los valores que almacenan los atributos o derivados de estos*).
 - ▶ El proceso de creación de los objetos (**métodos constructores**)



2024/25

Ejercicio

Se quiere definir un tipo Automóvil de cara a un proyecto informático para un portal de compra-venta de vehículos.

*¿ **Qué propiedades** se nos ocurre que debemos guardar para cada objeto que se cree del tipo Automóvil?*

A ver, ¿quién se atreve a aportar alguna propiedad relevante sobre un coche para un portal de compraventas?



2024/25

Elementos básicos del lenguaje

- Identificadores.
- Palabras reservadas de Java
- Tipos de datos predefinidos
- Literales
- Comentarios



2024/25

Identificadores

- Sirven para nombrar a los diversos elementos que constituyen el código
- Nombran a las: **C**lases, **I**nterfaces, **m**étodos, **a**tributos, **v**ariables, **C**ONSTANTES, **p**arámetros y **p**aquetes.
- Deben cumplir *ciertas condiciones* para ser válidos:
 - Están formados por secuencia de **letras, dígitos y los símbolos '_' y '\$'**.
 - No pueden coincidir con **palabras reservadas** de Java.
 - No pueden comenzar por un dígito.
- Pueden tener cualquier longitud.
- No pueden tener espacios en blanco en medio.
- Son sensibles a **mayúsculas** y **minúsculas**.



2024/25

Palabras reservadas

- Palabras que tienen una semántica predefinida (significan algo concreto para Java). Las más **usuales**: **Tipos de datos**, aunque usaremos *envolturas*

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

- Elementos básicos del lenguaje Java



Literales

2024/25

- ▶ Elementos del lenguaje que permiten referenciar directamente los valores constantes de los distintos tipos del lenguaje.

Ejemplo	Tipo
2	int
3.1415926	double
'a'	char
1000L	long
300.5f	float
false, true	boolean
"Antonio"	String (<i>no es un tipo básico sino un objeto</i>)

- ▶ Elementos básicos del lenguaje Java



2024/25

Comentarios

- Permiten incluir texto libre dentro del código
- Java ignora los comentarios: no tienen ningún efecto en la ejecución
- Sirven para hacer el código más legible
- Dos tipos de comentarios:
 - De una línea:
`// Este texto es un comentario de una sola línea`
 - De varias líneas:
`/* Este texto es un comentario de
varias líneas.
concretamente de tres líneas */`



2024/25

Visualización de texto en consola

- Se utiliza el método “*System.out.println(...)*”, al que se le pasa como parámetro un String o “*cualquier objeto*”.

Veremos que cuando se indica un objeto realmente se invoca automáticamente al método `toString()` de la clase que define al objeto.

- La consola (también llamada “terminal”) es el método más simple de interacción con el usuario
 - La información se muestra mediante texto en la consola
 - Los datos son solicitados al usuario mediante el teclado



2024/25

Tipos de datos predefinidos

- Un tipo de datos está formado por un conjunto de valores posibles o dominio, y ello puede posibilitar un conjunto de operaciones que se pueden realizar sobre los mismos.
- Tipos de datos básicos, nativos o primitivos de Java son:
 - *boolean* → (para datos de tipo lógico: *true* o *false*)
 - *short, int, long* → (para datos numéricos enteros –no tienen decimales-)
 - *char* → (para datos que sean un carácter)
 - *float, double* → (para datos numéricos reales –tienen decimales-)
 - *void* → (para indicar que carece de valor –vacío-)



2024/25

Tipos de datos predefinidos

- Otros tipos (no son básicos) incluidos en Java (son **tipos objetos**):
 - *String* → (para datos cadena o texto)
 - *LocalDate* → (para trabajar con fechas)
 - *LocalTime* → (para trabajar con horas)
 - *Duration* → (para trabajar con duraciones: días, horas, minutos...)
 - *Period* → (para trabajar con periodos de tiempo: días, semanas, mese, años...)



2024/25

Java es un lenguaje “Tipado”

- Es necesario declarar cada variable antes de utilizarla. Para ello se indica el tipo delante de la variable separado por, al menos, un espacio.

Por ejemplo: `int edad;`

- En ocasiones, Java convierte **automáticamente** un tipo primitivo en otro:

Regla general: cuando el tipo de la variable receptora es afín y tiene mayor capacidad expresiva que el tipo de la expresión.

Se puede inducir en un literal numérico un tipo:

- *long* si la final se añade una **l** o una **L**
- *double* si al final se añade una **d** o una **D**

Ejercicio: (decida si son correctos o no):

`long n=10;`

`int n=10L;`

`double d = 20.54;`

`float f = 33.2d;`

`int i=5.3f;`

`float f = 10.0 + 3;`

`double d = 20.54f;`



2024/25

Conversiones de tipos

Cuando la conversión no es automática, se puede forzar mediante el operador de conversión (*casting*) de tipos.

```
double d = 33.2;  
int i=(int) d*10;
```

```
double d = 33.25;  
int i=(int) d*10;
```

Operador de conversión de tipo:

se escribe entre paréntesis el tipo al que se quiere convertir el resultado de la expresión de la derecha.

- Si *d* tiene 33.2, *d*10* evalúa a 332.0; con el operador de *casting* *i* almacenará 332.
- Si *d* tiene 33.25, *d*10* evalúa a 332.5; con el operador de *casting* *i* almacenará 332.



2024/25

Ejercicio

- Enumere qué tipos de datos predefinidos conoce para almacenar:
 - La *edad* de una persona.
 - La *estatura* de una persona.
 - La *inicial del nombre* de una persona.
 - El *nombre* de una persona.
 - Si una persona *es apta o no apta* para un estudio.
 - La *fecha* de nacimiento de una persona.



2024/25

Ejercicio

- Señale el tipo de los siguientes literales:
 - 50
 - true
 - 'x'
 - 50.5
 - 2L
 - 0.35f
 - “FP, mi asignatura favorita”



2024/25

Practicamos un poco

1. Crear un proyecto Java: *T01_PrimerosEjercicios*.
2. Dentro del proyecto un paquete *primerosEjercicios*
3. Una clase *Ejercicios_01* que incorpore el método *main()*.

Este proyecto lo usaremos para ir practicando las siguientes dispositivas.



2024/25

Ejercicio

- **Declare** variables para almacenar cada uno de los valores anteriores, e **inicialícelas** con los mismos.
 - a) 50
 - b) true
 - c) 'x'
 - d) 50.5
 - e) 2L
 - f) 0.35f
 - g) “FP, mi asignatura favorita”



2024/25

Ejercicio

- Pruebe si son correctos los siguientes identificadores:
 - a) `IdentificadorMásLargoQueUnDíaSinPan`
 - b) `num`
 - c) `num80`
 - d) `80num`
 - e) `num_2`
 - f) `_num`
 - g) `$pred$`
 - h) `num-num80`
 - i) `Var`



2024/25

Tipos de datos envoltura (wrappers)

En Java determinados métodos necesitan “trabajar” con objetos, por lo que para cada tipo básico existe un objeto que lo representa

Tipos envolturas:

- Boolean → boolean
- Short, Integer, Long → short, int y long
- Character → char
- Float, Double → float y double
- Void → void

Empiezan por mayúsculas porque son tipos de datos (nombres de Clases)



2024/25

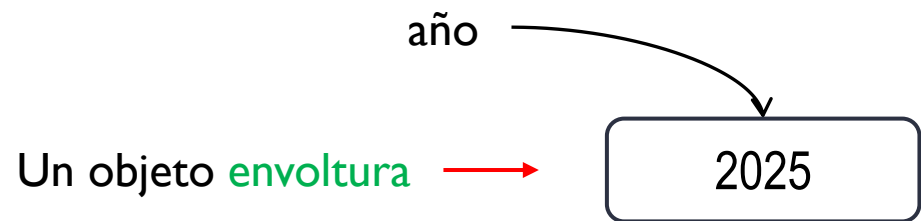
Tipos de Datos

- **Tipos de datos predefinidos** (denominados como *básicos*, nativos o *primitivos*)
- **Tipos envolturas** (*wrappers*: son objetos que como propiedad contiene un tipo primitivos y permiten trabajar los dichos tipos predefinidos)

`int` año=2025



`Integer` año=2025



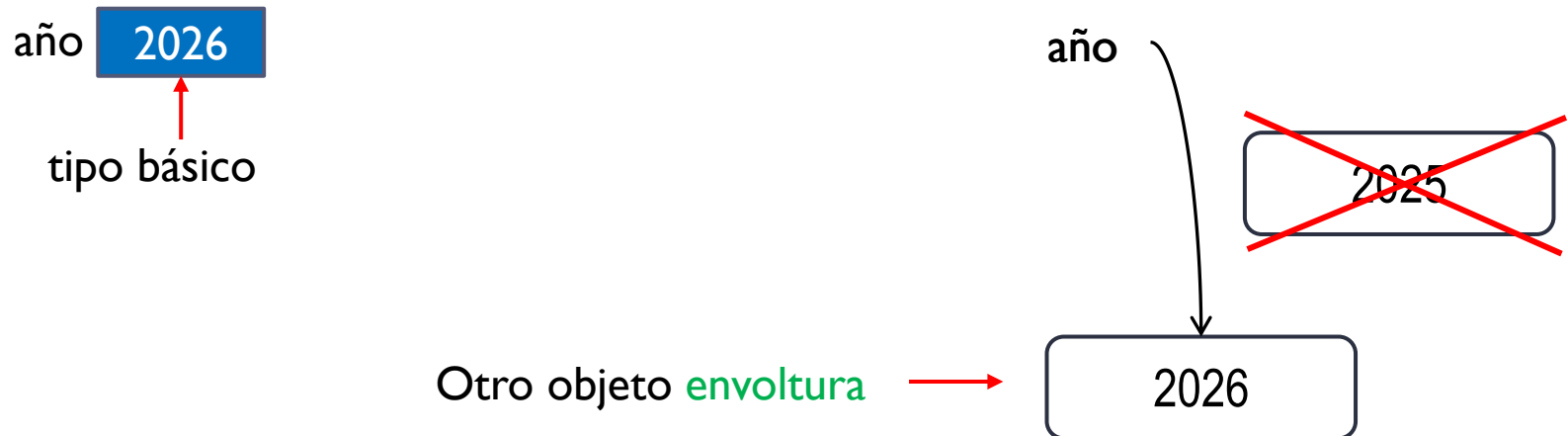


2024/25

Tipos de Datos

- **Tipos de datos predefinidos** (denominados como *básicos*, nativos o *primitivos*)
- **Tipos envolturas** (*wrappers*: son objetos que como propiedad contiene un tipo primitivos y permiten trabajar los dichos tipos predefinidos)

Cambiamos el valor de año. Por ejemplo: $\text{año} += 1$





2024/25

Tipos primitivos vs. Tipos objeto

Tipos **envoltura** (*wrappers*): Los tipos envoltura son compatibles con los tipos primitivos

```
int i = 10;
```

```
Integer i2 = i; // Java convierte int en Integer (autoboxing)
```

```
int i3 = i2; // Java convierte Integer en int (unboxing)
```

¿Qué tipo de datos usar?

- *Sobre todo*, como atributo de objetos que creemos en nuestros programas
- Para un índice de un for, un contador, un “sumador” y, en general, las variables que constantemente están actualizándose en un cálculo matemático.



2024/25

Tipos enumerados

- Un tipo enumerado es el tipo más sencillo que podemos definir utilizando enum.
- Las variables declaradas de un tipo enumerado únicamente pueden tomar dichos valores y NO se escribe entre comillas.

```
public enum ColorNatural{  
    ROJO, AZUL, VERDE, AMARILLO, BLANCO, NEGRO  
}  
public enum ColorSemaforo{  
    ROJO, AMARILLO, VERDE  
}  
ColorNatural miColor=ColorNatural.ROJO;  
ColorSemaforo colorDeParar= ColorSemaforo.ROJO;
```



2024/25

Variables y Constantes

- **Variables**
- **Constantes**



2024/25

Variables y Constantes

- **Variables**

- Las **variables** son elementos del lenguaje que permiten *guardar y acceder* a los datos que se manejan
- En Java es necesario declararlas antes de usarlas en cualquier parte del código y, por convenio, se escriben en **minúsculas**

```
int valor;
```

```
Double a1 = 2.25, a2 = 7.0;
```

```
char c = 'T';
```

```
String cadena= "Curso de Java";
```

```
Color relleno = Color.AZUL;
```



2024/25

Variables y Constantes

- **Constantes**

- Las constantes son elementos del lenguaje que permiten guardar y referenciar datos que van a permanecer invariables durante la ejecución del código.
- La declaración de una constante comienza por la palabra reservada *final*.
- Se escriben enteras en mayúsculas

```
final int DÍAS_SEMANA = 7;
```

```
final Double PI = 3.1415926;
```

```
final String TÍTULO = "E.T.S. de Ingeniería Informática";
```

En la declaración se puede omitir una asignación, pero una vez asignado un valor ya no se puede cambiar, por ejemplo:

```
final int N;
```

...

```
N = 2*6;    → a partir de aquí la constante N valdrá siempre 12.
```



2024/25

Operadores, Expresiones y Asignaciones

- Operadores
- Expresiones
- Asignaciones



2024/25

Operadores

Elementos que, a partir de uno o varios operandos, obtienen un resultado o modifican el valor de alguno de los operandos.

Aritméticos	
+	Suma
-	Resta
*	Producto
/	División
%	Módulo

Lógicos	
&&	y (and)
	o (or)
!	no (not)

De relación	
>	mayor que
<	menor que
>=	mayor o igual que
<=	menor o igual que
==	igual que/idéntico a
!=	distinto de/no idéntico a

Otros	
.	Invocar método
(tipo)	Conversión de tipo
=	Asignación



2024/25

Operadores Aritméticos

Permiten operaciones aritméticas entre variable y/o literales numéricos.

No obstante, el *operador suma aplicado a dos String* devuelve la *concatenación* de las mismas.

Aritméticos	
+	Suma
-	Resta
*	Producto
/	División
%	Módulo

También para String (concatenación)



2024/25

Operadores de Relación

Permite evaluar la relación de orden entre dos expresiones debidamente construidas

De relación	
>	mayor que
<	menor que
>=	mayor o igual que
<=	menor o igual que
==	igual que/idéntico a
!=	distinto de/no idéntico a

Sólo para números o caracteres.

OJO:

No vale para las cadenas, ni para objetos.

Habrà que construir métodos específicos



2024/25

Operadores Lógicos

Permite evaluar el resultado de varios operadores de relación.

Unen dos operadores de relación

Lógicos	
&&	y (and)
 	o (or)
!	no (not)



2024/25

Otros Operadores

Permiten:

- `·` → Invocar a un método o acceder a un atributo.
- `(tipo)` → Cambiar el tipo de un resultado (*casting*)
- `=` → Asignar un valor a una variable. Dicha variable debe estar escrita a la izquierda del `=`

Otros	
<code>·</code>	Invocar método
<code>(tipo)</code>	Conversión de tipo
<code>=</code>	Asignación



2024/25

Prelación entre Operadores

Existe un orden de prelación entre los operadores:

Operador	Asociatividad
. ()	
++ -- (tipo) new	derecha a izquierda
* / %	izquierda a derecha
+ -	izquierda a derecha
< <= > >=	izquierda a derecha
== !=	izquierda a derecha
&&	izquierda a derecha
	izquierda a derecha
=	derecha a izquierda





2024/25

Operadores aritméticos abreviados

La operación de asignación junto a las operaciones aritméticas puede escribirse de manera abreviada:

Operadores de asignación	
Abreviado	No abreviado
a += b	a = a + b
a -= b	a = a – b
a *= b	a = a * b
a /= b	a = a / b
a %= b	a = a % b
a++ / a--	a = a + 1 ; a+=1 / a=a-1;a-=1



2024/25

Expresiones

- Una expresión es un conjunto *bien formado* de constantes, variables y operadores.
- Evaluar una expresión consiste en aplicar los operadores y obtener un valor de algún tipo.
- El **tipo de una expresión** es el tipo del valor obtenido al evaluar la expresión. Si en la operación intervienen tipos compatibles el resultado será el de mayor capacidad.

Ejemplos:

$3+8$ \Rightarrow Tipo de la expresión: int

$(3*a)==b$ \Rightarrow Tipo de la expresión: boolean.

$3 * 8.6$ \Rightarrow Tipo de la expresión: double



2024/25

Asignaciones

Una asignación es una expresión que cambia el valor de la variable que se escribe a la izquierda.

El nuevo valor será el obtenido al evaluar la expresión que se escribe a la derecha.

Ejemplo:

```
int a;  
a=3+9;
```

¡El tipo de la expresión **debe coincidir** con el tipo de la variable!

```
int a;  
a=3>10; // ¡incorrecto!
```




Escriba en Java las siguientes expresiones:

- a) Una expresión lógica que sea cierta si un número entero 'num' está comprendido entre 1 y 5 (ambos inclusive) o bien es igual a 9.
- b) Una expresión lógica que sea cierta si un número entero 'num' es múltiplo de 3 y está comprendido entre 20 y 30 (ambos inclusive).
- c) Una expresión lógica que devuelva *true* si el año 'a' es bisiesto, y *false* en caso contrario. Recuerde que un año es bisiesto si es divisible por 400, o bien si es divisible por 4 pero no por 100.



Ejercicios

2024/25

Una expresión que calcule el área de un trapecio de bases 'b1' y 'b2' y altura 'h'. Recuerde que el área de un trapecio se obtiene mediante la fórmula:

$$a = \frac{(b1 + b2) * h}{2}$$

Para probar asigne a b1, b2 y h los valores que desee.