Apuntes Rainvare

POO

Qué es Java?

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems.
Decimos que el lenguaje de programación Java es un lenguaje fuertemente tipado. Pero esto, ¿qué quiere decir? Un lenguaje tipado es el que me exige una declaración explícita de la variable antes de comenzar a usarla.
Entonces, para declarar una variable, es necesario indicar el tipo de dato y nombre que se le asigna. Recordar que Java es un lenguaje case sensitive, los tipos de datos siempre se escriben en minúscula (excepto los string).

```
int valor;
double coeficiente;
String nombre;
```

DON'T FORGET

- El comienzo del programa en Java se indica con public

```
public static void Main(String args[])
```

- Si queremos mostrar algo por consola usamos PrintIn

```
System.out.println("Mi primer ejemplo ");
```

Una vez declarada la variable, sólo podrá utilizarse con datos del tipo indicado, es decir, una variable de tipo int no podrá almacenar un valor de tipo double, una variable de tipo String no podrá almacenar un valor numérico que se utilice para hacer operaciones aritméticas

Operaciones

Respecto a las operaciones aritméticas, debemos tomar en cuenta que si se opera entre dos variables de tipo entero, el resultado es siempre un valor de tipo entero. Esto pasa con todos los tipos de datos, es decir, una operación solo puede realizarse con variables del mismo tipo y el resultado mantiene el tipo de dato. Pero hay una operación en la que podríamos querer cambiar el tipo de dato y que el resultado se diera en otro.

Sintaxis de Java

Operadores lógicos	
Y	&& (devuelve verdadero si las dos evaluaciones son verdaderas)
O	2palitos (devuelve verdadero si una de las dos evaluaciones son verdaderas)
No	! (devuelve lo opuesto al resultado de la evaluación)

Matematicos	
Suma	+
Resta	-
División	/
Multiplicación	*
Módulo	% (devuelve el resto de una división entera)
Operador unario sumar 1	++
Operador unario restar 1	--

Operadores relacionales	
Mayor	>
Menor	<
Igual	== o .equals()
Mayor o igual	>=
Menor o igual	<=
No igual	!=

Tipos de datos primitivos	
byte	Números enteros entre -128 y 127
short	Números enteros entre -32768, 32767
int	Números enteros entre -2147483648 y 2147483647
long	Enteros muy grandes, entre -9223372036854775808 y 9223372036854775807
float	Número con coma -3.402823e38 a 3.402823e38
double	Número con coma, mayor capacidad -1.79769313486232e308 a -1.79769313486232e308
string	Cadena de caracteres
char	Un carácter (Ej: 'a') Unicode
boolean	Verdadero o falso (true /false)

Las estructuras de control en Java tienen la misma sintaxis que en JavaScript.

Contamos con:

✓ Decisión if - switch

✓ Repetición while - for

Clases

El núcleo de Java son las clases, más adelante veremos que son y cómo construirlas. Pero para comenzar a trabajar debemos comenzar a utilizar las clases propias de Java.

En este caso tendremos un elemento que:

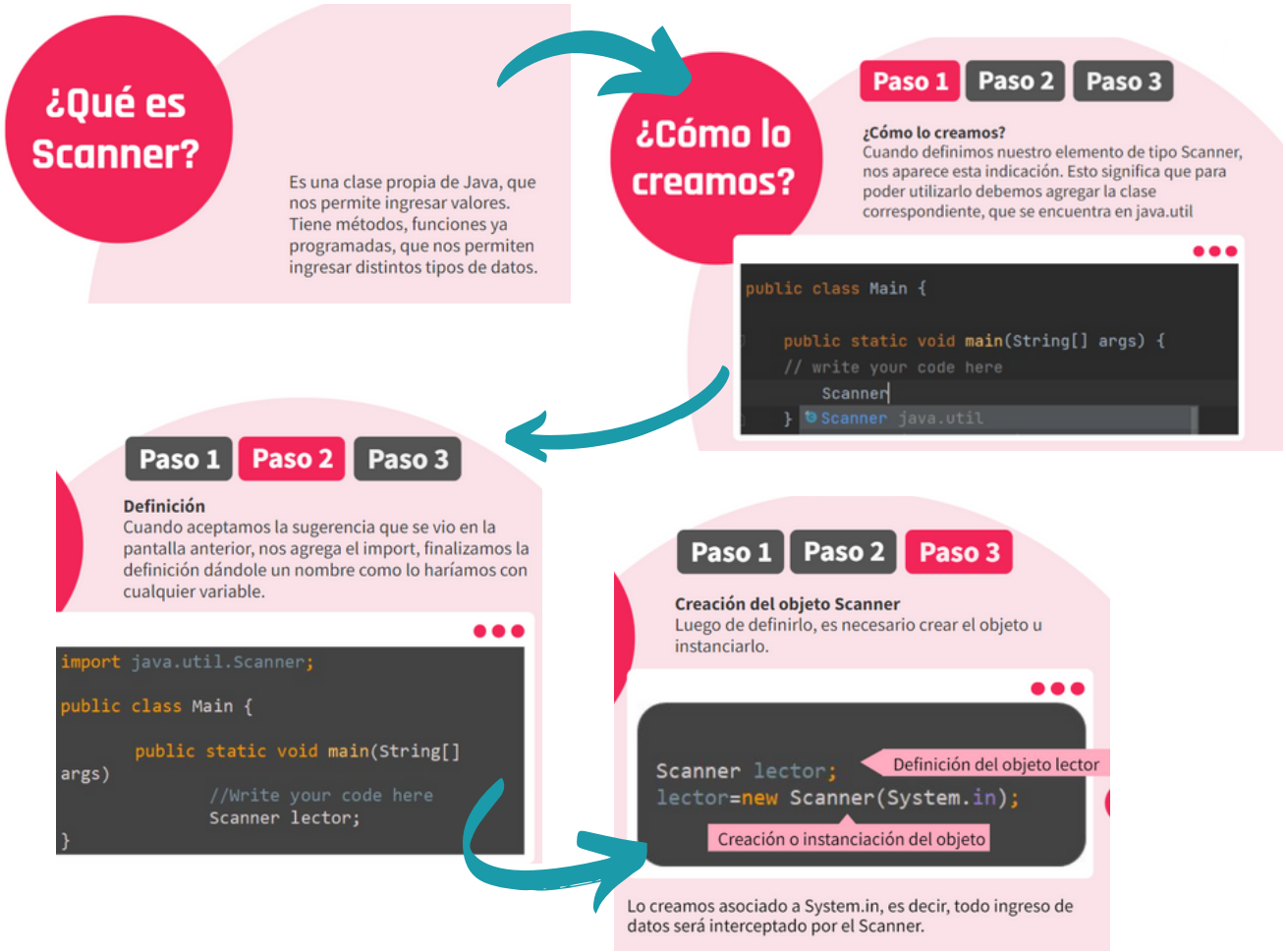
✅ Almacena un valor,

✅ Nos permite realizar ciertas operaciones que ya vienen programadas, a estas operaciones las llamamos métodos.

Por ejemplo:
String es una clase, por eso, se la inicializa en mayúscula; ya que todas las clases las nombramos con la inicial en mayúscula

Ingreso de datos, Scanner

Muchos elementos de Java son clases, vimos anteriormente String, Integer y Float. Para realizar la entrada y salida de datos también utilizamos clases propias de Java.



Métodos

Método .equals()

Algo a tener en cuenta cuando usamos estas clases es que no podemos usar operadores como "==", para efectuar una comparación por igual usamos .equals()

Método .compareTo()

Si queremos comparar si un valor es mayor o menor que otro debemos usar .compareTo()
Otra cosa a destacar es que una String a la cual no le asignamos nada tiene el valor null

Métodos para Strings

.length() //calcula longitud del string
.toUpperCase() //convierte a mayúscula
.equals() //comprueba
.toChar() //obtiene caracteres, en el () indicamos la ubicación

Clase System

Una clase muy importante es System, en ella encontramos System.in y System.out, que nos permitirán interactuar con las entradas y salidas del programa.

Ya vimos que Sistem.out.println nos permite mostrar un dato o mensaje. Para ingresar valores vamos a utilizar System.in.

Las entradas se realizan mediante esta clase, es decir, la información ingresa a través de System.in, pero para gestionarla y asignarla a las variables utilizaremos los métodos que nos provee Scanner

Funciones

Las funciones en Java son similares a las vistas en JavaScript, pero hay algunas cosas a tener en cuenta por ser un lenguaje tipado, vamos a tener que definir más cosas.

Definir una funcion
Para definirla vamos a considerar 3 cosas:
✓ Qué devuelve la función
✓ Qué nombre tiene
✓ Los parámetros que necesitamos

Cuando decimos qué devuelve nos referimos al tipo de dato que devuelve la función.

Parámetros

No hay muchas diferencias en cuanto a los parámetros, solo que es necesario indicar el tipo de cada uno.

Tipo devuelto

Las funciones pueden devolver un valor de retorno de algún tipo determinado, por ejemplo int, double, Integer, String, etc. En realidad pueden devolver cualquier cosa no solo valores, también estructuras enteras.

hay otro tipo de funciones, las que no devuelven nada en ese caso en donde indicamos el tipo devuelto colocaremos la palabra reservada void.

Veamos un ejemplo.

```
void mostrarMensaje(String mensaje) //Cuando no queremos que devuelva nada
```

Usamos las funciones de tipo void, cuando queremos que nuestra función sólo realice una serie de pasos o acciones y no nos devuelva nada.

Implementación de la función

Hasta ahora vimos cómo definir una función, ahora veamos que varía en la implementación, vamos a tener dos situaciones.

1. Que la función tenga valor de retorno
En el primer caso, debemos incluir un return con el valor devuelto, el tipo de este valor tiene que coincidir con el tipo de dato indicado como tipo devuelto.
2. Que no devuelve nada.

Paquetes

Para organizar las clases, existen los paquetes, estos son contenedores donde se pueden agrupar las clases. Más adelante los utilizaremos para nuestras clases, pero por ahora debemos saber que también las clases de Java se encuentran agrupadas en paquetes, o como su nombre en inglés: package.