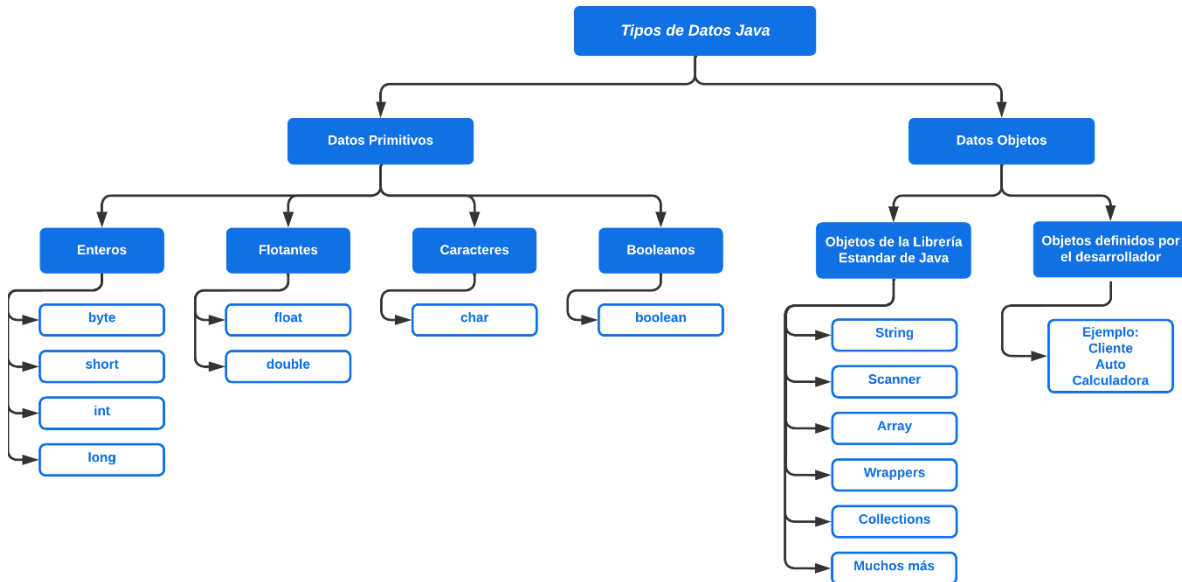


Clase 2: Tipos de datos

2.1. Tipos de datos en Java

2.1.1. ¿Cuáles son los tipos de datos en Java?

En Java tenemos variedad de datos, pero podemos dividirlos en dos grandes grupos: Datos de tipo primitivos y datos de tipo objetos.



2.1.2. Tipos de datos Primitivos

Son los tipos de datos más simples que podemos utilizar en Java. Son aquellos que nos proporciona el lenguaje y con los que podemos construir otros tipos de datos más complejos.

Los datos primitivos pueden ser clasificados en cuatro grupos: Enteros, Flotantes o Decimales, Caracteres y Booleanos.

Recordemos que Java es sensible a las mayúsculas por la que todos los datos de tipo primitivos se deben escribir en **minúsculas**.

2.1.2.1. Enteros

- Este grupo esta conformado por: byte, short, int y long.
- Todos son valores de tipo entero sin punto decimal.
- Lo que varía entre ellos son sus tamaños en bytes de memoria y su rango de alcance.
- Si bien tenemos 4 tipos de datos enteros en la gran mayoría de casos usaremos el tipo de dato int.

Nombre	Bytes	Rango
byte	1	-128 a 127
short	2	-32,768 a 32,767
int	4	-2,147,483,648 a 2,147,483,647
long	8	-9,223,372,036,854,775,808 a 9,223,372,036,854,775,807

```

1 byte enteroByte = 1;
2 short enteroShort = 32000;
3 int enteroInt = 1000000;
4 long enteroLong = 5000000L;

```

2.1.2.2. Flotantes/Decimales

- Este grupo está conformado por: float y double.
- Se utiliza para guardar números reales con números decimales.
- Se pueden usar números negativos y positivos.

Nombre	Bytes	Rango
float	4	-1.4E-45 a 3.4E-38
double	8	-4.9E-324 a 1.8E-308

```

1 float decimalFloat = 30.45;
2 double decimalDouble = 48925.12;

```

2.1.2.3. Caracteres

- Conformado por el tipo de dato: char.
- Nos permite trabajar con: caracteres alfanuméricos, caracteres especiales y caracteres de escape.
- Sus valores siempre deben estar encerrados entre **comillas simples** (' ').
- Solo puede contener (1) carácter.
- Puede contener mas de (1) carácter dentro de las comillas simples cuando se le asigna una secuencia de escape UNICODE.
- Usa el código UNICODE.

```

1 char caracterLetra = 'a';
2 char caracterEspecial = '%';
3 char caracterNumererico = '1';
4 char caracterUnicode = '\u00E1';

```

2.1.2.4. Booleanos

- Conformado por el tipo de dato: boolean.
- Nos permite trabajar con valores lógicos: verdadero o falso.
- Para especificar un literal boolean, se usan las palabras clave true (verdadero) o false (falso).
- true y false amabas deben ser escritas en minúsculas.
- Ocupan solo 1 bit de memoria.
- Son el tipo de dato más simple.

```

1 boolean a = true;
2 boolean b = false;

```

2.1.3. Tipo de Datos Objetos

Java es un lenguaje orientado a objetos. Para poder aplicar esa orientación a objetos en nuestros programas debemos de utilizar tipos de datos objetos.

Podemos clasificar estos objetos en dos grandes grupos: Objetos de la librería estándar de Java (Java Standard Library) y Objetos definidos por el desarrollador (nosotros mismos).

Los objetos los veremos a detalle en el Capítulo 2 por lo que puedes tomar la siguiente información como una introducción solamente.

2.1.3.1. Objetos de la librería estándar de Java

Dentro del JDK (Java Development Kit) de Java tenemos muchísimas clases predefinidas que podemos utilizar en nuestros programas.

Dependiendo de nuestras necesidades de programación podremos hacer uso de las clases predefinidas en los paquetes del JDK creando objetos de esas clases.

La ventaja de los objetos de la librería estándar de Java sobre los datos primitivos es que éstos objetos vienen con funciones predefinidas en el JDK que facilitan la construcción de programas en Java.

Si ya tienes instalado un JDK en tu computador podrás ver fácilmente todas las clases disponibles para utilizar en tus programas que vienen incluidas en las librerías estándar de Java.

Solo debes ir a donde tengas guardada la carpeta JDK -> Abrir la carpeta src.zip y encontraras todas las librerías (carpetas) disponibles.

En mi caso la dirección de mi librería src del JDK es la siguiente:

C:\Program Files\Java\jdk1.8.0_341\src.zip\java

Nombre	Fecha de modifica...	Tipo	Tamaño
applet		Carpeta de archivos	
awt		Carpeta de archivos	
beans		Carpeta de archivos	
io		Carpeta de archivos	
lang		Carpeta de archivos	
math		Carpeta de archivos	
net		Carpeta de archivos	
nio		Carpeta de archivos	
rmi		Carpeta de archivos	
security		Carpeta de archivos	
sql		Carpeta de archivos	
text		Carpeta de archivos	
time		Carpeta de archivos	
util		Carpeta de archivos	

Cada una de las carpetas listadas se conocen como librerías estándar y contienen clases disponibles para utilizar por los desarrolladores Java.

Si bien son muchísimas clases podemos nombrar algunas de las más utilizadas:

Clase String

- Definido dentro de la librería java.lang.
- Uno de los tipos de datos más utilizados.
- Esta compuesto por un conjunto de caracteres.

- Sus valores siempre deben estar encerrados dentro de comillas dobles (" ").
- Un objeto de tipo cadena se declara utilizando la palabra String (en mayúscula).
- Los objetos String en principio son datos sencillos y fáciles de utilizar, pero tienen muchísimos métodos que facilitan su uso y aplicación. Para poder entender a fondo como los objetos String funciona debemos entender como funcionan las clases, métodos y objetos por lo que veremos a fondo como funcionan en el capítulo 2.

```

1 //Declaración de un objeto String
2 String nombre = "Omar";
3 String apellido = "Montoya";

```

Clase Array

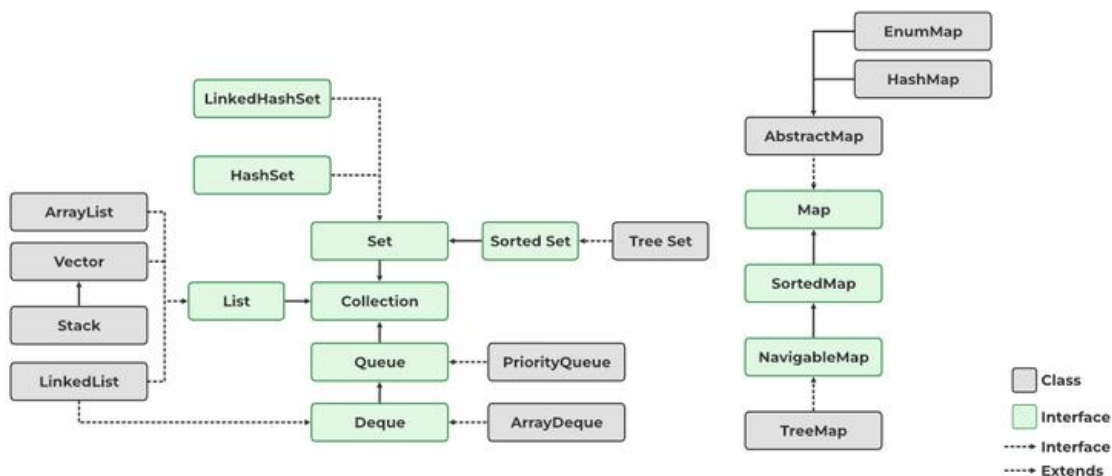
- Un array es una estructura de datos que nos permite almacenar una lista de datos de un mismo tipo.
- Son estáticos: Se declara su tamaño desde un principio y no puede cambiarse cuando el programa se esté ejecutando.
- Es la estructura de dato mas fundamental en los lenguajes de programación.
- Pueden tener 1 o más dimensiones.
- Veremos a detalles los arrays en otro capítulo.

Clases de tipo Colecciones (Collections)

las colecciones son estructuras de datos que se utilizan para almacenar y manipular grupos de elementos. Existen varias interfaces y clases en el paquete java.util que representan diferentes tipos de colecciones.

Los objetos de tipo collection en Java son: ArrayList, LinkedList, Vector, HashSet, LinkedHashSet, TreeSet, LinkedList, PriorityQueue, HashMap, LinkedHashMap, TreeMap y ArrayDeque.

Los objetos e interfaces collection son un tema avanzado que veremos en otro capítulo.

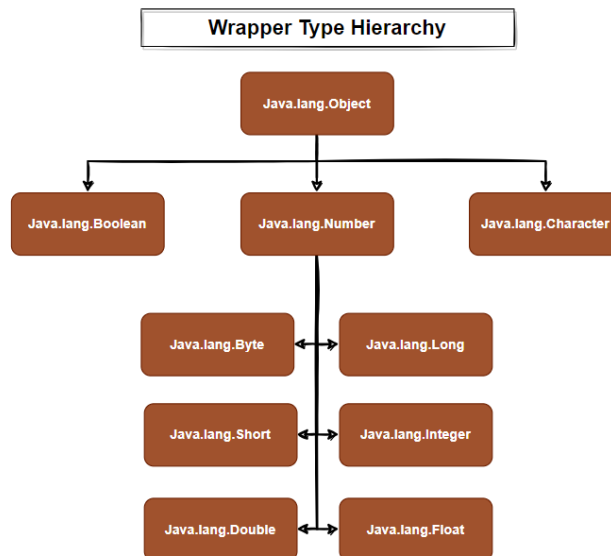


Clase Wrapper

Para todos los tipos de datos primitivos, existen una clase Wrapper (Envoltorio en español).

Proveen una serie de mecanismos que nos permiten envolver a un tipo de dato primitivo permitiéndonos utilizarlos como Objetos.

Dato Primitivo	Wrapper
byte	Byte
short	Short
int	Int
long	Long
boolean	Boolean
float	Float
double	Double
char	Char



La primera diferencia que se puede mencionar sobre ambos es que el tipo de dato primitivo se escribe con minúscula mientras que la clase Wrapper se escribe con **mayúscula**.

Al implementar una clase Wrapper tenemos acceso a métodos preddefinidos en el JDK para facilitar su manejo en los programas.

Uno de sus principales usos es debido a que cuando utilizamos colecciones en Java debemos utilizar siempre Objetos. No podemos proveerles datos primitivos a las colecciones. Entonces, ¿Cómo definimos una colección de números enteros, decimales, caracteres o booleanos? Fácil, en vez de utilizar datos primitivos utilizamos sus respectivas clases Wrappers.

2.1.3.1. Objetos definidos por el desarrollador

Las clases del JDK son sumamente útiles a la hora de desarrollar software, pero un programa complejo no se puede construir solamente con clases de la librería estándar de Java. Siempre necesitamos crear y definir nuestras propias clases.

En las clases definimos las variables (llamados atributos) y las funciones (llamados métodos) que necesitamos implementar en nuestro programa. Para poder utilizar esos atributos y métodos necesitamos definir objetos de esa clase que definimos.

Por ejemplo: Podemos definir una clase llamada Estudiante con los atributos nombre, apellido, edad, escuela y que tenga métodos para acceder a esos valores, modificarlos y crear objetos de esa clase.

Los objetos y clases son fácilmente el tema más importante para comprender en Java y a su vez es el tema que más les cuesta aprender a los desarrolladores cuando están empezando a aprender Java.

En el capítulo 2 veremos a fondo qué son las clases y como crear objetos.