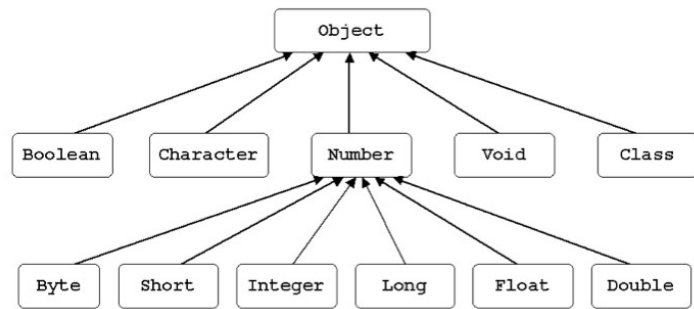


Clases de envoltura en Java

La **clase wrapper en Java** proporciona el mecanismo *para convertir primitivo en objeto y objeto en primitivo*.

Desde J2SE 5.0, la función de **boxing** y **unboxing** convierte primitivos en objetos y objetos en primitivos automáticamente. La conversión automática de primitivo en un objeto se conoce como autoboxing y viceversa unboxing.



Uso de clases Wrapper en Java

Debemos tratar con objetos muchas veces, con Colecciones, Serialización, Sincronización, etc. Veamos diferentes escenarios, donde necesitamos usar las clases de contenedor.

- **Cambiar el valor en un método:** Java solo admite llamadas por valor para tipos primitivos. Entonces, si pasamos un valor primitivo, no cambiará el valor original. Pero, si convertimos el valor primitivo en un objeto, cambiará el valor original.
- **Serialización:** Necesitamos convertir los objetos en flujos para realizar la serialización. Si tenemos un valor primitivo, podemos convertirlo en objetos a través de las clases de contenedor.
- **Sincronización:** la sincronización Java funciona con objetos en Multithreading.
- **Paquete java.util:** el paquete java.util proporciona las clases de utilidad para tratar con objetos.
- **Framework de colecciones:** el marco de colecciones funciona solo con objetos. Todas las clases del marco **colección** (ArrayList, LinkedList, Vector, HashSet, LinkedHashSet, TreeSet, PriorityQueue, ArrayDeque, etc.) solo tratan con objetos.

Las ocho clases del paquete *java.lang* se conocen como clases wrapper en Java. La lista de ocho clases de envoltura se da a continuación:

Tipo primitivo	Clase de envoltura
Boolean	Boolean
char	Char
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double