

Tema 10

Wrappers de Tipos Primitivos:

- Java proporciona clases "wrapper" en el paquete java.lang para cada tipo primitivo.
- Estos wrappers permiten tratar los tipos primitivos como objetos.
- Los wrappers incluyen: Boolean, Character, Byte, Short, Integer, Long, Float, Double.
- Son útiles cuando se necesitan objetos en lugar de tipos primitivos, por ejemplo, al usar colecciones.

Métodos Comunes de los Wrappers:

- Constructores que aceptan un tipo primitivo o un String.
- Métodos como valueOf() para convertir un String al wrapper.
- Métodos como parseInt() para convertir un String al tipo primitivo.
- Métodos como toString() para convertir el tipo primitivo a String.
- Métodos para obtener el valor primitivo (por ejemplo, intValue()).
- Constantes para valores máximo y mínimo (por ejemplo, Integer.MAX_VALUE).

Autoboxing y Autounboxing:

- **Autoboxing:** La conversión automática de un tipo primitivo a su correspondiente objeto wrapper.
 - Ejemplo: Integer a = 5; (convierte el int 5 al Integer objeto).
- **Autounboxing:** La conversión automática de un objeto wrapper a su correspondiente tipo primitivo.
 - Ejemplo: int b = a; (convierte el Integer objeto a al int primitivo).
- Estas características facilitan el uso de wrappers y permiten operar con ellos de forma más natural.

¿Cuál es el propósito principal de las clases wrapper en Java?

- a) Aumentar la velocidad de los cálculos matemáticos.
- b) Permitir el uso de tipos primitivos como objetos.
- c) Definir nuevas estructuras de datos.
- d) Controlar el acceso a la memoria.

¿En qué paquete de Java se encuentran las clases wrapper para los tipos primitivos?

- a) java.io
- b) java.util
- c) java.lang
- d) java.awt

¿Cuál de las siguientes NO es una clase wrapper para un tipo primitivo en Java?

- a) String
- b) Integer

- c) Double
- d) Boolean

¿Qué tipo de dato primitivo representa la clase wrapper Character?

- a) byte
- b) int
- c) char
- d) boolean

¿Cuál de los siguientes métodos se utiliza para convertir un String a un objeto Integer?

- a) parseInt()
- b) toString()
- c) valueOf()
- d) intValue()

¿Cuál de los siguientes métodos se utiliza para convertir un String a un valor int primitivo?

- a) parseInt()
- b) toString()
- c) valueOf()
- d) intValue()

¿Qué hace el método toString() en una clase wrapper?

- a) Convierte el objeto wrapper a su tipo primitivo.
- b) Convierte el objeto wrapper a un String.
- c) Compara dos objetos wrapper.
- d) Devuelve el valor máximo del tipo primitivo.

¿Qué tipo de valor devuelven los métodos como intValue(), doubleValue(), etc., de las clases wrapper?

- a) Un objeto wrapper.
- b) Un String.
- c) El valor primitivo correspondiente.
- d) Un valor booleano.

¿Qué representa la constante Integer.MAX_VALUE?

- a) El valor mínimo que puede tener un int.
- b) El valor máximo que puede tener un int.
- c) El tamaño en bytes de un Integer objeto.
- d) El valor cero.

¿Qué es "autoboxing" en Java?

- a) La conversión automática de un objeto wrapper a un tipo primitivo.
- b) La conversión automática de un tipo primitivo a su objeto wrapper correspondiente.
- c) La creación automática de arrays de tipos primitivos.
- d) La declaración automática de variables..

¿Cuál de las siguientes líneas de código ejemplifica autoboxing?

- a) `int x = new Integer(10);`
- b) `Integer y = 20;`
- c) `String s = Integer.toString(30);`
- d) `double d = new Double("40.5");`

¿Cuál de las siguientes líneas de código ejemplifica autounboxing?

- a) `Integer a = new Integer(5);`
- b) `int b = a;`
- c) `Integer c = Integer.valueOf("15");`
- d) `float f = 25.0f;`

¿Qué facilita el autoboxing y autounboxing en Java?

- a) El uso de tipos primitivos en estructuras de datos orientadas a objetos.
- b) La creación de nuevas clases.
- c) La definición de interfaces.
- d) La gestión de la memoria.

¿Cuál es la principal ventaja de usar autoboxing y autounboxing?

- a) Aumenta la velocidad de ejecución del programa.

- b) Hace el código más legible y conciso.
- c) Permite la herencia múltiple.
- d) Simplifica la gestión de excepciones.

¿Cuál de las siguientes operaciones es válida gracias al autounboxing?

- a) `Integer x = 10 + 5;`
- b) `int y = new Integer(10) + 5;`
- c) `String s = new Integer(10).toString();`
- d) `boolean b = new Boolean(true) && false;`

¿Qué ocurre al intentar asignar un valor null a un tipo primitivo mediante autounboxing?

- a) Se asigna el valor 0 al tipo primitivo.
- b) Se lanza una `NullPointerException`.
- c) Se asigna el valor `false` al tipo primitivo (si es `boolean`).
- d) Se produce un error de compilación.

¿Cuál de las siguientes comparaciones es válida gracias al autounboxing?

- a) `Integer a = 5; String b = "5"; if (a == b) ...`
- b) `Integer a = 5; int b = 6; if (a == b) ...`
- c) `int a = 5; Integer b = null; if (a == b) ...`
- d) `Integer a = null; Integer b = 5; if (a == b) ...`

¿Cuál de las siguientes afirmaciones es CIERTA sobre los constructores de las clases wrapper?

- a) Solo pueden recibir un `String` como argumento.
- b) Solo pueden recibir el tipo primitivo que representan como argumento.
- c) Pueden recibir tanto un `String` como el tipo primitivo que representan como argumento.
- d) No tienen constructores.

¿Qué tipo de dato se devuelve al llamar al método `intValue()` en un objeto `Integer`?

- a) `Integer`
- b) `String`
- c) `int`

d) boolean

¿Cuál es la función del método valueOf(String s) en las clases wrapper?

- a) Convertir un tipo primitivo a String.
- b) Convertir un String al tipo primitivo correspondiente.
- c) Convertir un String al objeto wrapper correspondiente.
- d) Comparar dos Strings.

¿Cuál de las siguientes opciones describe mejor la relación entre un tipo primitivo y su clase wrapper?

- a) La clase wrapper "contiene" el tipo primitivo como un objeto.
- b) El tipo primitivo "hereda" de la clase wrapper.
- c) Son intercambiables sin ninguna diferencia.
- d) No tienen ninguna relación.

¿Qué ventaja proporciona el uso de wrappers al trabajar con colecciones en Java?

- a) Las colecciones solo pueden almacenar objetos, no tipos primitivos.
- b) Los wrappers aumentan la eficiencia de las colecciones.
- c) Los wrappers permiten ordenar las colecciones más rápidamente.
- d) No hay ninguna ventaja.

¿Cuál de las siguientes afirmaciones es FALSA sobre el autoboxing y autounboxing?

- a) Pueden hacer el código más corto y legible.
- b) Pueden simplificar las operaciones con wrappers.
- c) Pueden eliminar por completo la necesidad de usar tipos primitivos.
- d) Pueden ocurrir de forma automática.

¿Qué precaución se debe tener al usar autounboxing?

- a) Asegurarse de que el objeto wrapper no sea null para evitar NullPointerException.
- b) Evitar usarlo dentro de bucles para no afectar el rendimiento.
- c) No usarlo con tipos float y double por posibles pérdidas de precisión.
- d) No hay que tener ninguna precaución especial.

1. b
2. c
3. a
4. c
5. c
6. a
7. b
8. c
9. b
10. b
11. b
12. b
13. b
14. a
15. b
16. b
17. b
18. b
19. c
20. c
21. c
22. a
23. a
24. c
25. a