Programación

Unidad 10: Wrappers de tipos primitivos

Unidad 10

- Wrappers de tipos primitivos
- Boolean
- Character
- Byte
- Short
- Integer
- Long
- Float
- Double
- Autoboxing y Autounboxing

Wrappers de tipos primitivos

Unidad 10: Wrappers de tipos primitivos

Wrappers de tipos primitivos

Hay ocasiones en las que necesitaríamos usar un tipo primitivo como un objeto (tipo complejo). Por ejemplo, cuando queremos guardar números en una colección (las veremos más adelante) que solo admite **java.lang.Object**

En el paquete **java.lang.*** existe un wrapper para cada tipo primitivo (no siempre coincide el nombre):

- Boolean
- Character
- Byte, Short, Integer, Long
- Float, Double

Unidad 10: Wrappers de tipos primitivos

Wrappers de tipos primitivos

Casi siempre, suelen tener los siguientes métodos:

Constructores que reciben un String o el tipo primitivo que representan:

- `Integer a = new Integer(3);`

Convertidores de tipo String a su tipo complejo (wrapper):

- `Integer b = Integer.valueOf("3");`

Convertidores de tipo String al tipo primitivo que representan:

- `int c = Integer.parseInt("3");`

Convertidores de tipo primitivo a String:

- `String d = Integer.toString(c);`

Extractores del tipo primitivo que representan:

```
int e = b.intValue();
```

Unidad 10: Wrappers de tipos primitivos

Wrappers de tipos primitivos

Todas estas conversiones son susceptibles de producir errores.

¿Qué pasaría si se intenta crear un **Integer** utilizando **“Hola”** como parámetro?

Siempre que la conversión no sea posible, la JVM lanzará una **excepción** del tipo:

`java.lang.NumberFormatException`

Ya veremos como capturar y manejar este tipo de errores en el apartado dedicado a las excepciones.

Unidad 10: Wrappers de tipos primitivos

Wrappers de tipos primitivos

Todos los wrappers **sobreescriben** el método **equals** de la clase **java.lang.Object**.

De esta forma, podemos saber si dos objetos distintos de un mismo tipo de wrapper, representan el mismo valor primitivo o no.

```
public static void main(String[] args) {
    Integer i1 = new Integer(3);
    Integer i2 = 3;
    System.out.println(i1==i2);
    System.out.println(i1.equals(i2));
}
```

```
<terminated> Te
false
true
```

Boolean

Unidad 10: Wrappers de tipos primitivos

Wrappers de tipos primitivos - Boolean

Clase java.lang.Boolean.

Es el wrapper del tipo primitivo boolean.

`public Boolean (boolean value);`

Constructor de un Boolean con el boolean value.

`public Boolean (String s);`

Constructor de un Boolean con el String s. Si s no vale ”true” entonces siempre tomará el valor false.

`public static Boolean valueOf(String s);`

Convierte el String s en un Boolean. Si s no vale “true” entonces siempre devuelve un Boolean con false.

Unidad 10: Wrappers de tipos primitivos

Wrappers de tipos primitivos - Boolean

`public static boolean parseBoolean(String s);`

Convierte el String s en un boolean. Si s no vale “true” entonces siempre devuelve false.

`public static String toString(boolean b);`

Convierte el boolean b en un String.

`public String toString();`

Devuelve su representación String (sobreescribe el método toString() de java.lang.Object).

`public boolean booleanValue();`

Extrae el boolean que representa.

Más información y métodos:

https://docs.oracle.com/en/java/javase/23/docs/api/java.base/java/lang/Boolean.html

Character

Unidad 10: Wrappers de tipos primitivos

Wrappers de tipos primitivos - Character

Es el wrapper del tipo primitivo char.

`public Character(char value);`

Constructor de un Character con el char value.

`public static String toString(char s);`

Convierte el char c en un String.

`public String toString();`

Devuelve su representación String (sobreescribe el método toString() de java.lang.Object).

`public char charValue();`

Extrae el char que representa.

- Más información:

  https://docs.oracle.com/en/java/javase/23/docs/api/java.base/java/lang/Character.html

Byte

Unidad 10: Wrappers de tipos primitivos

Wrappers de tipos primitivos - Byte

Es el wrapper del tipo primitivo short.

`public Byte(byte value);`

Constructor de un short con el short value.

`public Byte(String s);`

Constructor de un short con el String s.

`public static Byte valueOf(String s);`

Convierte el String s en un Byte.

`public static byte parseByte(String s);`

Convierte el String s en un byte.

Unidad 10: Wrappers de tipos primitivos

Wrappers de tipos primitivos - Byte

`public static String toString(byte b);`

Convierte el byte b en un String.

`public String toString();`

Devuelve su representación String (sobreescribe el método toString() de java.lang.Object).

`public byte byteValue();`

Extrae el byte que representa.

Más información y métodos:

https://docs.oracle.com/en/java/javase/23/docs/api/java.base/java/lang/short.html

Short

Unidad 10: Wrappers de tipos primitivos

Wrappers de tipos primitivos - Short

Es el wrapper del tipo primitivo short.

`public Short(short value);`

Constructor de un Short con el short value.

`public Short(String s);`

Constructor de un Short con el String s.

`public static Short valueOf(String s);`

Convierte el String s en un Short.

`public static short parseShort(String s);`

Convierte el String s en un short.

Unidad 10: Wrappers de tipos primitivos

Wrappers de tipos primitivos - Short

`public static String toString(short b);`

Convierte el short b en un String.

`public String toString();`

Devuelve su representación String (sobreescribe el método toString() de java.lang.Object).

`public short shortValue();`

Extrae el short que representa.

Más información y métodos:

https://docs.oracle.com/en/java/javase/23/docs/api/java.base/java/lang/short.html

Integer

Unidad 10: Wrappers de tipos primitivos

Wrappers de tipos primitivos - Integer

Es el wrapper del tipo primitivo int.

`public Integer(int value);`

Constructor de un Integer con el int value.

`public Integer(String s);`

Constructor de un Integer con el String s.

`public static Integer valueOf(String s);`

Convierte el String s en un Integer.

`public static int parseInteger(String s);`

Convierte el String s en un int.

Unidad 10: Wrappers de tipos primitivos

Wrappers de tipos primitivos - Integer

- **`public static String toString(int b);`**

  Convierte el int b en un String.

- **`public String toString();`**

  Devuelve su representación String (sobreescribe el método toString() de java.lang.Object).

- **`public int intValue();`**

  Extrae el int que representa.

Más información y métodos:

https://docs.oracle.com/en/java/javase/23/docs/api/java.base/java/lang/Integer.html

Long

Wrappers de tipos primitivos - Long

Es el wrapper del tipo primitivo float.

`public Long(long value);`

Constructor de un Long con el long value.

`public Long(String s);`

Constructor de un Long con el String s.

`public static Long valueOf(String s);`

Convierte el String s en un Long.

`public static long parseLong(String s);`

Convierte el String s en un long.

Unidad 10: Wrappers de tipos primitivos

Wrappers de tipos primitivos - Long

`public static String toString(long b);`

Convierte el long b en un String.

`public String toString();`

Devuelve su representación String (sobreescribe el método toString() de java.lang.Object).

`public long longValue();`

Extrae el long que representa.

Más información y métodos:

https://docs.oracle.com/en/java/javase/23/docs/api/java.base/java/lang/Long.html

Float

Unidad 10: Wrappers de tipos primitivos

Wrappers de tipos primitivos - Float

Es el wrapper del tipo primitivo float.

`public Float(float value);`

Constructor de un Float con el float value.

`public Float(String s);`

Constructor de un Float con el String s.

`public static Float valueOf(String s);`

Convierte el String s en un Float.

`public static float parseFloat(String s);`

Convierte el String s en un float.

Unidad 10: Wrappers de tipos primitivos

Wrappers de tipos primitivos - Float

`public static String toString(float b);`

Convierte el float b en un String.

`public String toString();`

Devuelve su representación String (sobreescribe el método toString() de java.lang.Object).

`public float floatValue();`

Extrae el float que representa.

Más información y métodos:

https://docs.oracle.com/en/java/javase/23/docs/api/java.base/java/lang/Float.html

Double

Unidad 10: Wrappers de tipos primitivos

Wrappers de tipos primitivos - Double

Es el wrapper del tipo primitivo double.

`public Double(double value);`

Constructor de un Double con el double value.

`public Double(String s);`

Constructor de un Double con el String s.

`public static Double valueOf(String s);`

Convierte el String s en un Double.

`public static double parseDouble(String s);`

Convierte el String s en un double.

Unidad 10: Wrappers de tipos primitivos

Wrappers de tipos primitivos - Double

`public static String toString(double b);`

Convierte el double b en un String.

`public String toString();`

Devuelve su representación String (sobreescribe el método toString() de java.lang.Object).

`public double doubleValue();`

Extrae el double que representa.

Más información y métodos:

https://docs.oracle.com/en/java/javase/23/docs/api/java.base/java/lang/Float.html

Unidad 10: Wrappers de tipos primitivos

Wrappers de tipos primitivos - Double

```java
public class TestWrapper {

    public static void main(String[] args) {
        String texto = new String("3");
        byte b = Byte.parseByte(texto);
        System.out.println(b);
        short s = Short.parseShort(texto);
        System.out.println(s);
        int i = Integer.parseInt(texto);
        System.out.println(i);
        long l = Long.parseLong(texto);
        System.out.println(l);
        float f = Float.parseFloat(texto);
        System.out.println(f);
        double d = Double.parseDouble(texto);
        System.out.println(d);
    }

}
```

```
<terminate
3
3
3
3
3.0
3.0
```

Autoboxing y Autounboxing

Unidad 10: Wrappers de tipos primitivos

Wrappers de tipos primitivos – Autoboxing y autounboxing

Desde la versión 5.0 de Java se añade la posibilidad de que las conversiones entre tipos primitivos y sus wrappers se hagan de forma automática. Antes de esta versión, para crear un wrapper a partir de un tipo primitivo se utilizaba su constructor:

```
Integer i = new Integer(1);
```

Sin embargo ahora se puede hacer directamente:

```
Integer i = 1;
```

El compilador se encarga de realizar la conversión de forma automática (***autoboxing***).

Unidad 10: Wrappers de tipos primitivos

Wrappers de tipos primitivos – Autoboxing y autounboxing

De igual forma, antes para extraer un tipo primitivo de su wrapper utilizábamos el siguiente método:

```
Integer a = new Integer(1);
int b = a.intValue();
```

Sin embargo ahora se puede hacer directamente:

```
int b = a;
```

El compilador se encarga de realizar la extracción de forma automática (***auto-unboxing***). Esto nos permite también operar con los wrappers.

```
Integer a = 10; Integer b = 3; int c = a + b;
```

Unidad 10: Wrappers de tipos primitivos

Wrappers de tipos primitivos – Autoboxing y autounboxing

También se permiten las comparaciones:

```
Integer a = 5; int b = 6;
if(a == b)
System.out.println(“Iguales”);
```

El wrapper Boolean también se ve favorecido por esta nueva funcionalidad. Antes no podía participar en condiciones, pero ahora si:

```
Boolean a = true; boolean b = false;
Boolean c = a && b;
```

Unidad 10: Wrappers de tipos primitivos

Wrappers de tipos primitivos – Autoboxing y autounboxing

¿Y qué pasa con la sobrecarga de métodos?

```
public void metodo(double param){};
public void metodo(Integer param){};
int a = 5;
this.metodo(a);
```

Para evitar diferencias en la funcionalidad de una aplicación al migrar de versiones anteriores, **primero** se busca el método a ejecutar sin tener en cuenta el autoboxing y auto-unboxing. **Si no se encuentra ninguno**, entonces se busca teniendo en cuenta el autoboxing y auto-unboxing.

Se llamaría a:

```
public void metodo(double param){};
```