

Clases Envoltentes.

Java es un lenguaje de programación orientado a objetos puro. Un programa Java debe contener objetos y las operaciones entre ellos. La única excepción a esto son los tipos de datos primitivos (int, double, char, etc.)

Los tipos de datos primitivos no son objetos pero en ocasiones es necesario tratarlos como tales. Por ejemplo, hay determinadas clases que manipulan objetos (ArrayList, HashMap, ...). Para poder utilizar tipos primitivos con estas clases, Java provee las llamadas clases envoltentes también llamadas **Clases Contenedoras o Wrappers**.

Cada tipo primitivo tiene su correspondiente clase envoltente:

Tipo primitivo	Clase Envoltente
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
char	Character
boolean	Boolean

Estas clases proporcionan métodos que permiten manipular el tipo de dato primitivo como si fuese un objeto.

Las conversiones entre los tipos primitivos y sus clases envoltentes son automáticas. No es necesario hacer un casting. Para realizar las conversiones se utiliza el **Boxing/Unboxing**.

Boxing: Convertir un tipo primitivo en su clase Wrapper.

Unboxing: Convertir un objeto de una clase Wrapper en su tipo primitivo.

Ejemplo de Boxing:

```
double x = 29.95;
```

```
Double y;
```

```
y = x; // boxing
```

Ejemplo de Unboxing:

```
double x;
```

```
Double y = 29.95;
```

```
x = y; // unboxing
```

Clase Integer

En la siguiente tabla aparecen algunos métodos de la clase Integer. El resto de clases envolventes correspondientes a tipos primitivos numéricos tienen métodos similares.

Integer(int valor)	Constructor a partir de un int Integer n = new Integer(20);
Integer(String valor)	Constructor a partir de un String String s = "123456"; Integer a = new Integer(s);
int intValue() float floatValue() double doubleValue() ...	Devuelve el valor equivalente Integer n = new Integer(30); int x = n.intValue(); double y = n.doubleValue();
int parseInt(String s)	Método estático que devuelve un int a partir de un String. String s = "123456"; int z = Integer.parseInt(s);
String toBinaryString(int i) String toOctalString(int i) String toHexString(int i)	Métodos estáticos que devuelven un String con la representación binaria, octal o hexadecimal del número. int numero = 12; String binario = Integer.toBinaryString(numero);
Integer valueOf(String s)	Método Estático. Devuelve un Integer a partir de un String. Integer m = Integer.valueOf("123");

<http://docs.oracle.com/javase/9/docs/api/java/lang/Integer.html>

Clase Character

Provee una serie de métodos para manipular los datos de tipo char. En la siguiente tabla aparecen algunos de estos métodos.

Character(char c)	Constructor a partir de un char char car = 'x'; Character a = new Character(car);
char charValue()	Devuelve el char equivalente Character n = new Character('q'); char c = n.charValue();
boolean isLowerCase(char ch) boolean isUpperCase(char ch) boolean isDigit(char ch) boolean isLetter(char ch)	Comprueba si es un carácter en minúsculas. Comprueba si es un carácter en mayúsculas. Comprueba si es un dígito (carácter del 0 al 9). Comprueba si es una letra. Todos son estáticos. if(Character.isUpperCase(c)){ }

`char toLowerCase(char ch)`

Devuelve el char en minúsculas.

`char toUpperCase(char ch)`

Devuelve el char en mayúsculas.

Métodos estáticos.

`char car = 'u';``System.out.println(Character.toUpperCase(car));``Character valueOf(char c)`

Método Estático. Devuelve un Character a partir de un char.

`Character m = Character.valueOf('a');`<http://docs.oracle.com/javase/9/docs/api/java/lang/Character.html>

Ejemplo de uso de la clase Character: Programa que lee un texto y muestra cuántas letras y cifras tiene.

```
import java.util.Scanner;
public class JavaApplication325 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String texto;
        int cuentaCifras = 0, cuentaLetras = 0;
        System.out.println("Introduce texto ");
        texto = sc.nextLine();
        for (int i = 0; i < texto.length(); i++) {
            if (Character.isDigit(texto.charAt(i))) {
                cuentaCifras++;
            } else if (Character.isLetter(texto.charAt(i))) {
                cuentaLetras++;
            }
        }
        System.out.println("El texto contiene " + cuentaCifras + " dígitos");
        System.out.println("El texto contiene " + cuentaLetras + " letras");
    }
}
```