Anexo. Clases Envoltorio o wrappers

Java es un lenguaje de programación orientado a objetos. Un programa Java debe contener objetos y las operaciones entre ellos. La única excepción a esto en un programa Java 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 envoltorio también llamadas clases contenedoras o wrappers.

Cada tipo primitivo tiene su correspondiente clase envoltorio:

Tipo primitivo	Clase Envoltorio
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 envoltorio son automáticas. No es necesario hacer un casting. Para realizarlas 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 envoltorio correspondientes a tipos primitivos numéricos tienen métodos similares.

`Integer(int valor)`
 	
`Integer(String valor)`
	Constructor a partir de un int
	`Integer n = new Integer(20);`
	
	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");`

API de la clase Integer:
http://docs.oracle.com/javase/7/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)`
`char toUpperCase(char ch)`
	Devuelve el char en minúsculas.
	Devuelve el char en mayúsculas.
	Son 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');`

API de la clase Character:
http://docs.oracle.com/javase/7/docs/api/java/lang/Character.html

Ejemplo de uso de la clase Character: programa que lee un texto por teclado y muestra cuántos dígitos y letras contiene.

```
import java.util.Scanner;

public class PruebaCharacter {

	public static void main(String[] args) {
	  Scanner teclado = new Scanner(System.in);
	  String texto;
	  int cuentaCifras = 0, cuentaLetras = 0;
	  System.out.println("Introduce texto ");
	  texto = teclado.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");
	}
}
```

Conversión entre números y cadenas

Las clases envoltorio permiten, entre otras cosas, realizar conversiones entre cadenas y números, obtener expresiones numéricas en diferentes bases, etcétera.
En el siguiente ejemplo, vemos cómo podemos realizar conversiones entre valores numéricos y cadenas y viceversa.

Algunos ejemplos son:

```
// --- operaciones con el tipo int ---
int i = 43;
// convierto de int a String
String sInt = Integer.toString(i);
// convierto de String a int
int i2 = Integer.parseInt(sInt);
// --- operaciones con el tipo double ---
double d = 24.2;
// convierto de double a String
```

```
String sDouble = Double.toString(d);
// convierto de String a double
double d2 = Double.parseDouble(sDouble);
```

Todas las clases envoltorio tienen los métodos **parseXxx** (siendo Xxx el tipo de datos al que se quiere parsear el string) y el método **toString** para convertir un valor del tipo que representan a cadena de caracteres.

Representación numérica en diferentes bases

Java, igual que C, permite expresar valores enteros en base 8 y en base 16. Para representar un entero en base 16, debemos anteponerle el prefijo 0x.

```
int i = 0x24ACF; // en decimal es 150223
System.out.println(i); // imprime 150223
```

Para expresar enteros en base 8, debemos anteponerles el prefijo 0.

```
int j = 0537; // en decimal es 351
System.out.println(j); // imprime 351
```

Utilizando la clase Integer podemos obtener la representación binaria, octal, hexadecimal y en cualquier base numérica de un entero dado.

Ejemplo: muestra un valor entero en diferentes bases numéricas.

```
import java.util.Scanner;
public class ConversionNumerica {

	public static void main(String[] args) {
		Scanner scanner=new Scanner(System.in);
		System.out.print("Introduzca un valor entero: ");
		int v = scanner.nextInt();
		System.out.println("Valor introducido: "+v);
		System.out.println("binario = "+Integer.toBinaryString(v));
		System.out.println("octal = "+Integer.toOctalString(v));
		System.out.println("hexadecimal = "+Integer.toHexString(v));
		System.out.print("Introduzca una base numerica: ");
		int b = scanner.nextInt();
		String sBaseN=Integer.toString(v,b);
		System.out.println(v + " en base("+b+") = " + sBaseN);
	}
}
```