

DESARROLLO DE APLICACIONES WEB

Anexo I - Unidad 12

Expresiones regulares

1r DAW

IES La Mola de Novelda

Departament d'informàtica

Índice

1.- Comprobar expresiones regulares.....3

2.- Ejemplos.....4

Anexo I / Unidad 12: Expresiones regulares

1.- COMPROBAR EXPRESIONES REGULARES

Para comprobar expresiones regulares en Java se pueden usar varias opciones:

- Usar el método ***matches()*** de la clase ***String***. El parámetro del método es la expresión regular que se aplica al propio string, y el método devuelve ***true*** o ***false***, según si el ***String*** cumple o el patrón de la expresión:

```
String cadena = "abc";
if(cadena.matches("[a-z]*"))
    System.out.println("Si cumple el patrón");
else
    System.out.println("No cumple el patrón");
```

Si cumple el patrón

- Usar el método ***matches()*** de la clase ***Pattern***. Casi igual que con ***String***. El método recibe la expresión regular que se aplica y un ***String*** que se comprueba. El método es estático, por lo que se puede llamar con la propia clase:

```
if (Pattern.matches("[a-z]*", "abc"))
    System.out.println("SI cumple patron");
else
    System.out.println("NO cumple patron");
```

SI cumple patron

- Usar la clase ***Pattern*** para "compilar" una expresión regular.
 - Si se va a usar varias veces una misma expresión regular, es mejor "compilar" una sola vez la expresión, y usarla luego varias veces contra distintos textos.
 - Para compilar y usar después el compilado:
 - Compilamos la expresión regular con ***Pattern.compile(regex)***. Devuelve un objeto de la clase ***Pattern***, que representa la expresión regular ya "compilada".
 - En el patrón compilado, aplicados ***matcher(texto)*** con el texto que tenemos que validar, y así se crea un objeto ***Matcher***, que es como un "tester". El resultado del "tester" se ve aplicando su método ***matches()*** que devuelve ***true*** si coincide el patrón en el texto indicado.

Veamos el siguiente ejemplo:

```
String cadena;
String expresionRegular = "[a-z]*";
Pattern compilado = Pattern.compile(expresionRegular);

do{
    System.out.println("Escribe una cadena: ");
    cadena = new Scanner(System.in).nextLine();
    Matcher mat = compilado.matcher(cadena);
    if(mat.matches())
        System.out.println("Si cumple el patrón");
    else
        System.out.println("No cumple el patrón");
}while(!cadena.equals("FIN"));
```

Escribe una cadena:
Hola
No cumple el patrón
Escribe una cadena:
hola
Si cumple el patrón
Escribe una cadena:
FIN
No cumple el patrón

Para futuros ejercicios se puede usar un simple método de elaboración propia:

```
public static boolean validarRegEx(String expresionregular, String valor) {
    return Pattern.matches(expresionregular, valor);
    // return valor.matches(expresionregular); // otra forma
}
```

2.- EJEMPLOS

```
// ----- Comprobar si un String contiene "abc"
String cadenabuena = "abc";
String cadenamala = "ab0";
String patron = "abc.*";
System.out.println(cadenabuena.matches(patron)); // TRUE
System.out.println(cadenamala.matches(patron)); // FALSE

// otra forma
System.out.println(Pattern.matches(patron, "abc")); // TRUE
System.out.println(Pattern.matches(patron, "abX")); // FALSE

// ----- Comprobar si un String contiene "abc" o "Abc"
System.out.println("abc".matches("[aA]bc.*")); // TRUE
System.out.println("aBc".matches("[aA]bc.*")); // FALSE

// ----- Comprobar si un String comienza por "abc" o "Abc"
System.out.println("abc".matches("^aA]bc.*")); // TRUE
System.out.println("aBc".matches("^aA]bc.*")); // FALSE

// ----- Comprobar si un String tiene un mínimo de 5 letras mayúsculas o minúsculas y un máximo de 10
System.out.println("ffasd".matches("[a-zA-Z]{5,10}$")); // TRUE
System.out.println("dfa44".matches("[a-zA-Z]{5,10}$")); // FALSE

// ----- Comprobar si un String solo contiene los caracteres 0 ó 1, y debe contener algo
System.out.println("01101001".matches("(0|1)+$")); // TRUE
System.out.println("10020102".matches("(0|1)+$")); // FALSE

// ----- Comprobar si un String es una matricula española (4 numeros y tres letras mayúsculas)
System.out.println("3423GCB".matches("[0-9]{4}[A-Z]{3}$")); // TRUE
System.out.println("342XGCB".matches("[0-9]{4}[A-Z]{3}$")); // FALSE

// ----- Comprobar si un String contiene o bien "abc" o bien "Abc"
System.out.println("zzabccz".matches("^[aA]bc.*$")); // TRUE
System.out.println("zzacczz".matches("^[aA]bc.*$")); // TRUE
```