

Introducción al desarrollo de RIA's con Adobe Flex 3.0 Dia 6

by S. Muñoz-Gutiérrez
stalinmunoz@yahoo.com,
informes@grupolinda.org

Grupo LINDA
Facultad de Ingeniería

UNAM México Octubre-Diciembre 2009



Expresiones regulares

- Se utilizan para reconocer secuencias de caracteres, búsquedas de patrones en cadenas y substituir segmentos dónde el patrón objetivo no es literal sino una familia de cadenas descritas por la sintaxis de nuestra expresión regular.
- Sigue norma ECMA-262
- AS3 soporta una sintaxis especial para definir expresiones regulares

Creando una expresión regular

- Utilizando diagonales "/"

```
var expresion1:RegExp = /abc/i;
```

- Utilizando el constructor de la clase

```
var expresion1:RegExp = new RegExp("abc","i");
```

Expresiones regulares
son instancias de la
clase RegExp

Expresión

Banderas indican como
se deben realizar las
comparaciones

Metacaracteres

- Las expresiones más simples son patrones literales
 - `/Hello/`
- Metacaracteres tienen un significado especial
 - `^ $ \ . * + ? () [] { } |`
- Si se desea que un metacaracter pierda su interpretación especial se debe anteceder con la diagonal invertida:
 - `\/\$/`

Interpretación de Metacaracteres

- **^ (caret)** comienzo de cadena o cuando se encuentra dentro de una clase de caracteres indica negación.
- **\$ (pesos)** fin de una cadena
- **\ (antidiagonal)** "escapa" o anula la interpretación de caracteres especiales.
- **.** (punto) cualquier caracter
- ***** (asterisco) 0 o más repeticiones del caracter previo

Interpretación de Metacaracteres

- **+** (**más**) 1 o más repeticiones del caracter previo
- **?** (**interrogación**) 0 o 1 repetición del caracter previo
- **()** (**paréntesis**) define grupos
- **[]** (**corchetes**) define clases de caracteres
- **|** (**"pipe"**) separa una o más opciones alternativas

Metasecuencias

- `{n}` n repeticiones
- `{n,}` n o más repeticiones
- `{n,m}` de n a m repeticiones
- `\d` dígito decimal
- `\D` cualquier caracter que no sea dígito decimal
- `\s` cualquier caracter en blanco (espacio, tabulador, nueva línea, retorno)
- `\S` cualquier caracter que no sea blanco

Clases

- Se definen con corchetes []
- Definen una lista de caracteres para una sola posición de la cadena
 - `/[aeiou]/` define una vocal
- Dentro de una clase los metacaracteres pierden su interpretación especial (exceptuando la antidiagonal).
- El guión se usa para indicar un rango
 - `/[0-9]/` indica cualquier dígito

Clases

- **^ (caret)** se utiliza para indicar el complemento de caracteres en una clase y se debe incluir al principio de la clase
 - **[^a-z]** indica cualquier caracter que no es letra inglesa
- Si se utiliza caret en cualquier otra posición pierde este significado y se interpreta de manera literal

Aplicación de cuantificadores

- `/[a-zñáéíóúü]+/` una o más letras españolas minúsculas
- `^*{2,10}/` de dos a 10 asteriscos consecutivos
- **"Greedy matching"** tratará de incluir tantos caracteres como sea posible, por ejemplo:
 - `/<p>.*</p>/` y la cadena
 - `"<p>hola</p><p>mundo</p>"`
 - Incluirá todos los caracteres de la cadena

Aplicación de cuantificadores

- Agregar ? (interrogación) tras cualquier cuantificador cambia su funcionamiento
- "Lazy matching" incluirá la cadena de longitud menor que corresponda con el patrón.
 - `/<p>.*?</p>/` y la cadena
 - `"<p>hola</p><p>mundo</p>"`
 - Identificara por separado las correspondencias:
 - `"<p>hola</p>"` y `"<p>mundo</p>"`

Aplicación de alternativas

- Usese para una lista de opciones
 - `/rojo|verde|azul|amarillo/`
 - Representa cualquiera de las cadenas rojo, verde, azul o amarillo
- Se pueden usar paréntesis para restringir el ambito de aplicación de las alternativas
 - `/color\s+(verde|blanco|rojo)/`
 - Limita las alternativas a la segunda palabra

Uso de grupos

- Se definen con paréntesis ()
- Cada correspondencia de un grupo es recuperable y puede ser utilizada en la misma expresión (**backreference**).
- Para recuperar un grupo se utiliza la antidiagonal y el índice del mismo comenzando en 1
 - `/(\d+), \1/` o bien `^(1,(\d+)/`
 - Identifica dos secuencia idénticas de dígitos separados por coma

Nombrando grupos

- Se pueden nombrar grupos utilizando la sintaxis `(?P<name> definición_de_grupo)`, ejemplo:

```
var tipoExp:RegExp =
```

```
  /(?P<nombre>\w+) extends (?P<tipo>\w+)/;
```


```
var cadena:String = "cat extends Animal";
```

```
var matches:Array = tipoExp.exec(cadena);
```

```
trace(matches.nombre); // cat
```

```
trace(matches.tipo); // Animal
```

Use exec para encontrar las correspondencias



Método exec

- Retorna un objeto con las siguientes propiedades
 - Un arreglo que tiene como primer elemento la cadena con la coincidencia y como elementos subsecuentes las cadenas de coincidencia de cada uno de los grupos de la expresión
 - Contiene un atributo `index` con el índice de la coincidencia
 - Contiene un atributo `input` con la cadena original

Banderas de control

- **g** busca por todas las correspondencias, no solo la primera coincidencia
- **i** ignora mayúsculas y minúsculas.
- **m** multiples líneas, si se establece \wedge y $\$$ se interpretan como comienzo y final de línea respectivamente
- **s** cuando se establece se permite que $.$ (punto) pueda corresponder con el caracter de nueva línea
- **x** se pueden incluir espacios en la expresión para hacerla más clara y no son interpretados

Ejercicio 1

- Implemente una pequeña interfaz con dos cajas de texto en las que el usuario pueda introducir en una una expresión regular y en la otra una cadena.
- Al oprimir un botón de "Verificar" se debe de evaluar la expresión regular y determinar si la expresión regular tiene alguna correspondencia en la cadena.

Ejercicio 2

- Desarrolle una clase que provee de un método estático que permita separar los comentarios de documentación de un método java.

```
/**
```

```
* Descripción del método
```

```
* @param param-name descripción
```

```
* @return descripción
```

```
*/
```

- Se debe retornar un objeto con la descripción del método, la descripción de lo que retorna y la lista de parámetros con su descripción.