Strings e Arrays

Strings

Un **String** representa datos en formato textual. Case todas as linguaxes teñen un tipo ou clase específica para almacenar texto. En Java, os strings son **inmutables**: cada operación que modifica o texto crea un novo obxecto.

En Java, os Strings son un **tipo de datos complexo** que se pode manipular mediante métodos (obxectos). Podes consultar no API de Java os métodos dispoñibles

```
// Almacena en saludo a referencia do obxecto "Ola".
String saludo = "Ola";

// crea un novo obxecto
String nuevoSaludo = saludo + " mundo";

// Ola mundo
System.out.println(nuevoSaludo);

// OLA MUNDO (chamaos ao método do obxecto almacenado en nuevoSaludo)
System.out.println(nuevoSaludo.toUppercase());

 // HOLA  (chamamos ao método toUppercase()  do obxecto "hola")
System.out.println("hola".toUppercase);
```

Cada transformación feita sobre un String (concatenación sobre todo) implica a creación dun novo obxecto String, o que non resulta eficiente.
Para mellorar a eficiencia de estas operacións existen as clases ***StringBuffer*** e ***StringBuilder***. As dúas clases fan mais o menos o mesmo, pero:

- ***StringBuilder*** é máis eficiente pero non funciona ben en aplicacións multifío na que varios fios fagan uso do mesmo obxecto StringBuilder ao mesmo tempo (non e sincronizada)
- ***StringBuffer*** é mais lento, pero funciona ben en aplicacións multifío (é sincronizada)

Arrays

Un array e un conxunto de elementos do mesmo tipo que se poden referenciar cunha única variable e acceder mediante un índice. O número de elementos do Array adoita ser fixo e definido no momento da creación do array (instanciación do Array).
Para o acceso a cada un dos elementos de xeito individual se usa un índice comezando dende o número 0 (polo tanto o último índice dun array de 10 elementos é o 9).

```
// Creación dun Array de 5 numeros int
int[] numeros=new int[5];
numeros[0]=10
numeros[6]=20

// O atributo length dos Array indica o número de elementos que ten o array
// Os elementos non asignados valen 0 en Java, pero iso e algo dependente da linguaxe
for(int i = 0; i < numeros.length; i++) {
   System.out.println(numeros[i]);
}
```

Tamén podemos instanciar o Array facendo unha inicialización:

```
String[] diasSemana={ "luns","martes","mércores","xoves","vernes","sábado","domingo" };
```

Si definimos un Array onde cada elemento é a súa vez outro Array teremos un Array de dúas dimensións, e poderemos utilizar dous índices para acceder aos elementos (podemos pensar en iso como si foran “filas” e “columnas”).

```
int[][] matriz = { {1,2}, {3,4} };
System.out.println(matriz[1][0]); // 3
```

O número de elementos dos Array se ten que decidir no momento en que se instancia xa sexa usando “new” ou cunha inicialización.

JAVA dispón da clase Arrays que proporciona un montón de métodos estáticos útiles para o traballo con obxectos Array.

As Expresións Regulares

Unha expresión regular é unha cadea que define un patrón para buscar, validar ou extraer datos dun String. Resulta moi útil para validar correos, teléfonos, formatos, ou extraer información.

```
import java.util.regex.*;

String texto = "O meu correo é xavier@example.com";
String patron = "\\w+@\\w+\\.\\w+";

Pattern p = Pattern.compile(patron);
Matcher m = p.matcher(texto);

if(m.find()) {
   System.out.println("Email atopado: " + m.group());
}
```

Elementos básicos:

Símbolo	Significado
.	Calquera carácter
*	Cero ou máis repeticións
+	Unha ou máis repeticións
?	Cero ou unha repetición
\d	Díxito
\w	Letra, número ou guion baixo
[]	Conxunto de caracteres
^ / $	Inicio / fin da cadea

(e necesario poñer unha \ antes da \ para que non se interprete como un caracter de escape como \n ou \t, si non como a letra \ real)
"\\w+@\\w+\\.\\w+", por exemplo indica:

- **\\w ou sexa, \w** : Letra, número ou guión baixo que ...
- **+** : deben aparecer como mínimo 1 ou máis, e logo ...
- **@** : ten que estar o caracter @, e logo...
- **\\w ou sexa, \w** : unha letra, numero ou guión baixo que ...
- **+** : debe aparecer como mínimo 1, e logo ...
- **\\. ou sexa \.** : un punto, e logo ... (ollo o \, se fora . Simplemente indicaría un caracter calquera)
- **\\w ou sexa, \w** : Letra, número ou guión baixo