

Examen parcial

<pre>if (condición) { sentencias; } if (condición) { sentencias; } else { sentencias; } if (condición) { sentencias; } else if (condición) { sentencias; } else { sentencias; }</pre>	<pre>for (inicialización; condición; actualización) { sentencias; } while (condición) { sentencias; } public static void nombre(parametros) { sentencias; } public static type nombre(parametros) { sentencias; return expresión; }</pre>
---	--

Clase String

`contains(str)`: true si la cadena contiene los caracteres de la otra
`endsWith(str)`, `startsWith(str)`: true si la cadena empieza o acaba con los caracteres de la otra
`equals(str)`: true si una cadena es igual a la otra
`equalsIgnoreCase(str)`: true si una cadena es igual a la otra, sin tener en cuenta mayúsculas
`indexOf(str)`: índice en la cadena de caracteres por el que empieza el string dado (-1 si no está)
`length()`: número de caracteres del string
`substring(i, j)`: caracteres que van desde el índice i (incluido) hasta j (excluido)
`substring(i)`: caracteres que van desde el índice i (incluido) hasta el final
`toLowerCase()`, `toUpperCase()`: string con todas sus letras en minúsculas o en mayúsculas
`charAt(i)`: carácter con índice i

Clase Scanner

`nextInt()`, `hasNextInt()`: lee un dato como int y comprueba que un dato se puede leer como int
`next()`, `hasNext()`: lee un dato como String y comprueba que es posible leerlo
`nextDouble()`, `hasNextDouble()`: lee un dato como double y comprueba que es posible leerlo
`nextLine()`, `hasNextLine()`: lee una línea como String y comprueba que es posible leerla

Clase Math

`Math.abs(valor)`: valor absoluto
`Math.min(v1, v2)`: valor menor de los 2 dados
`Math.max(v1, v2)`: valor mayor de los 2 dados
`Math.round(valor)`: número entero más cercano
`Math.pow(b, e)`: b elevado a e

Clase PrintStream

`println(texto);`
`print(texto);`

Clase File

`canRead()`: true si el fichero existe y se puede leer

Clase Random

`nextInt(max)`: entero aleatorio entre 0 y max - 1

Objetos

`Type objectName = new Type(parameters);`

Arrays

```
tipo[] nombre = new tipo[longitud];  
tipo[] nombre = {VAL1, VAL2, VAL3, ...};  
nombre[indice] = valor;  
nombre.length // Número de elementos del array
```

